Overview of the MVAPICH Project: Latest Status and Future Roadmap

MVAPICH2 User Group (MUG) Meeting

by

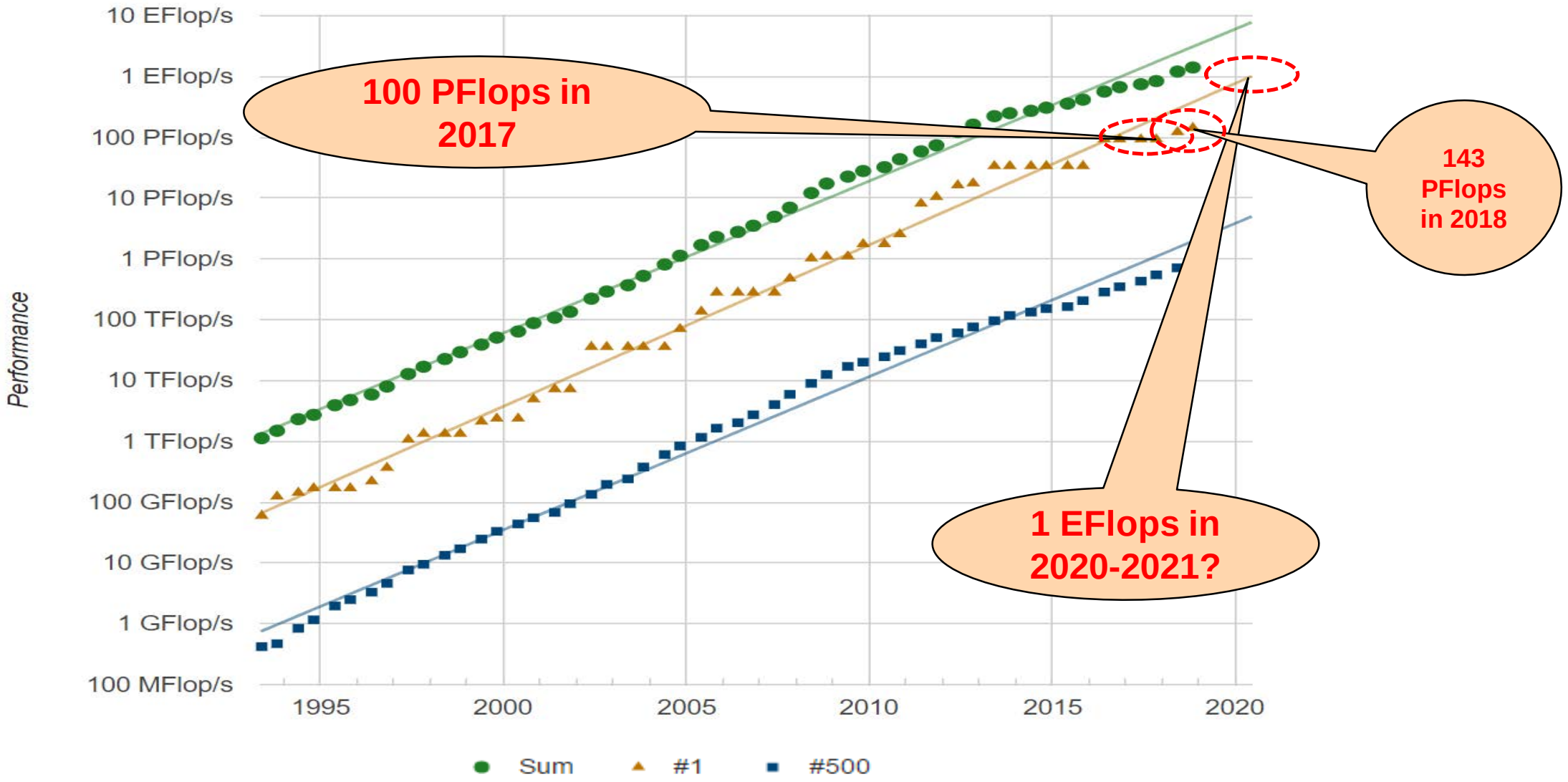
Dhabaleswar K. (DK) Panda

The Ohio State University

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

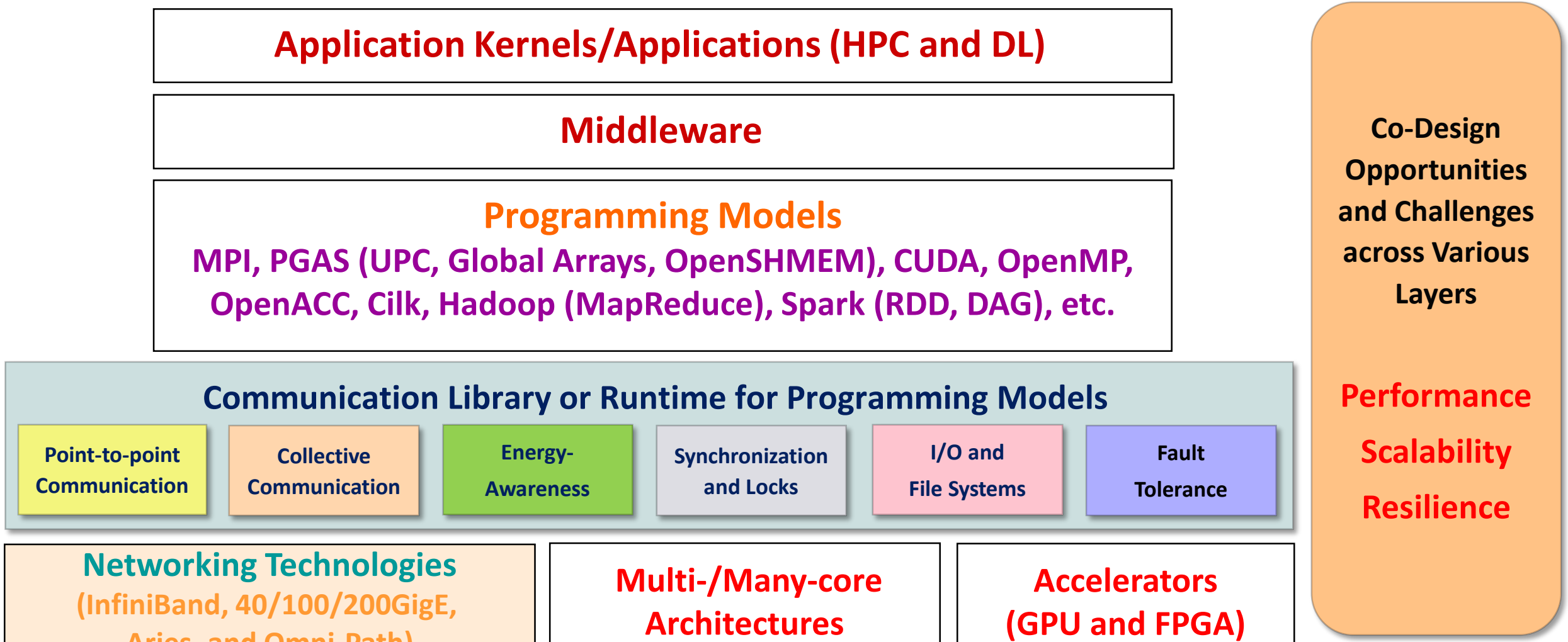
http://www.cse.ohio-state.edu/~panda

High-End Computing (HEC): PetaFlop to ExaFlop



Expected to have an ExaFlop system in 2020-2021!

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



Designing (MPI+X) at Exascale

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

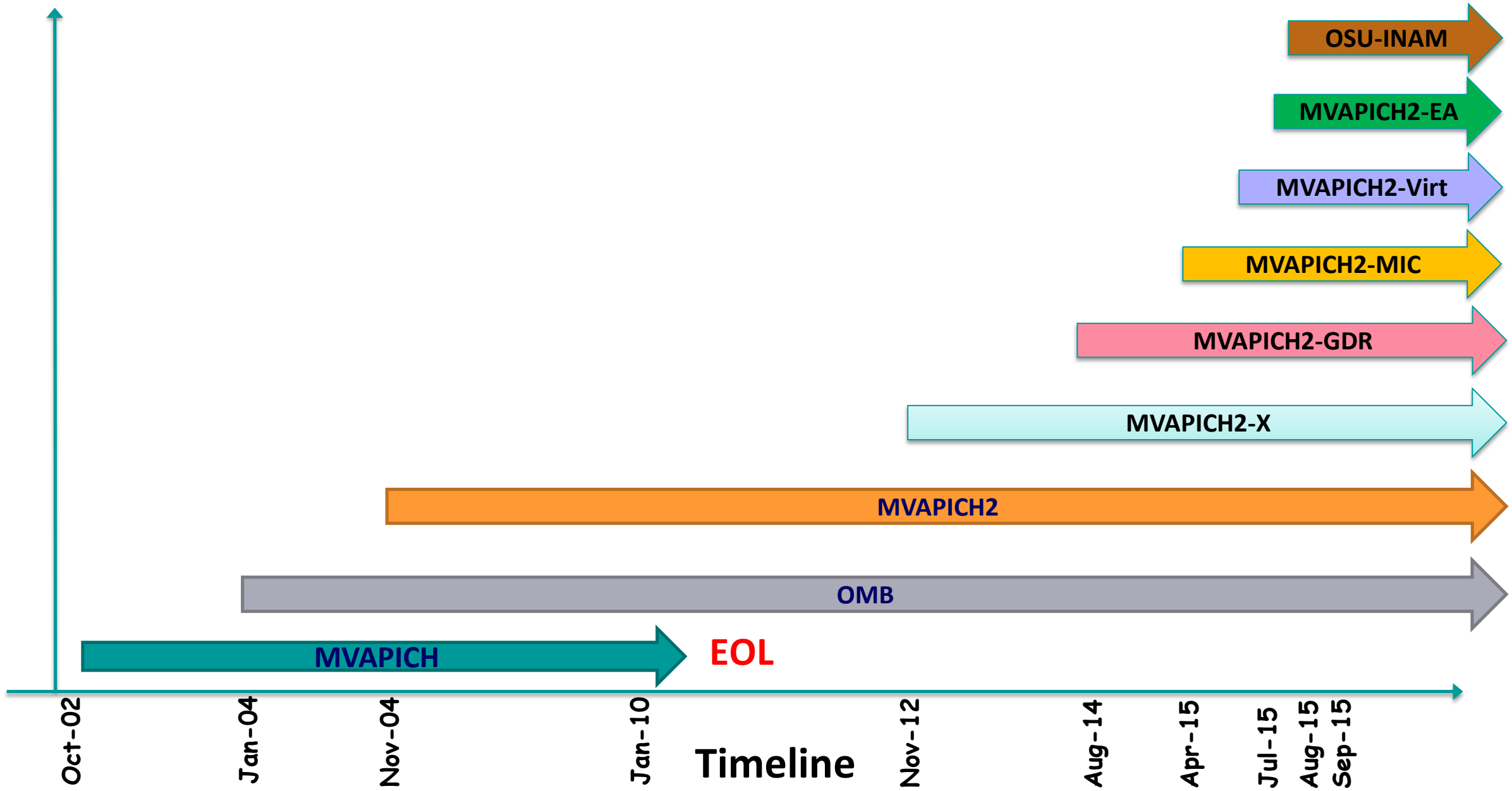
Overview of the MVAPICH2 Project

- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.1), Started in 2001, First version available in 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
  - **Used by more than 3,025 organizations in 89 countries**
  - **More than 564,000 (> 0.5 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Nov '18 ranking)
    - 3rd, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China
    - 5th, 448, 448 cores (Frontera) at TACC
    - 8th, 391,680 cores (ABCI) in Japan
    - 15th, 570,020 cores (Neurion) in South Korea and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)
  - **http://mvapich.cse.ohio-state.edu**

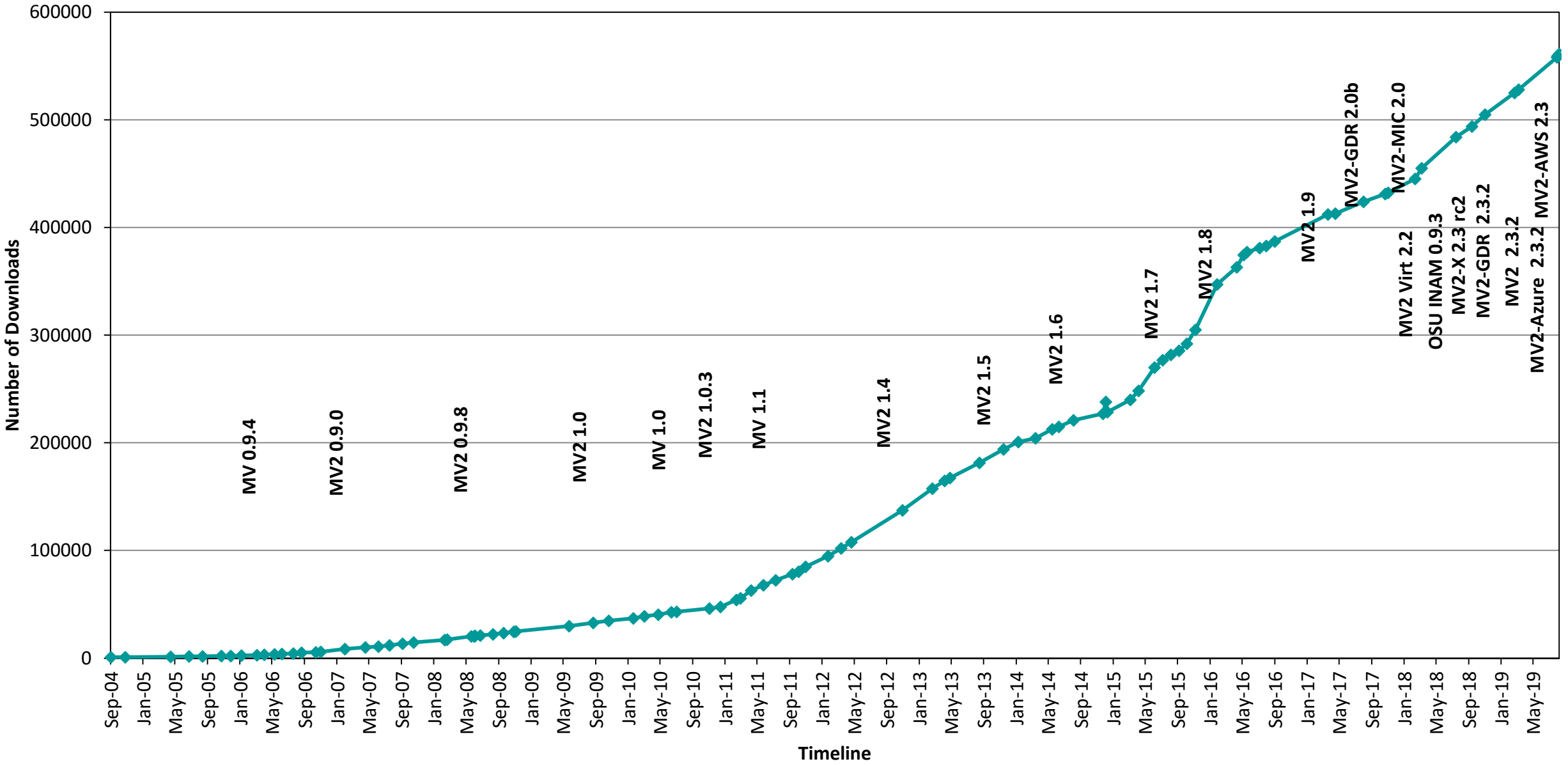
Partner in the TACC Frontera System

- Empowering Top500 systems for over a decade

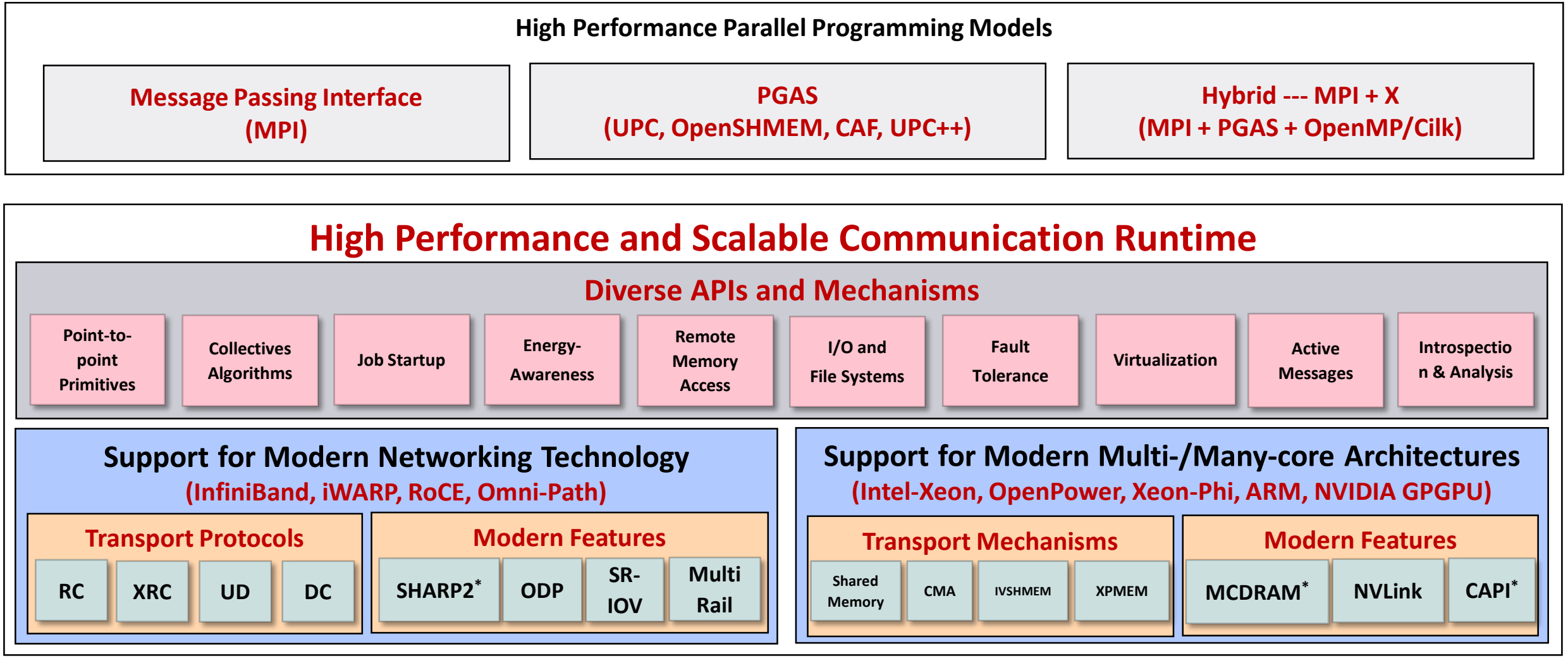
MVAPICH Project Timeline



MVAPICH2 Release Timeline and Downloads



Architecture of MVAPICH2 Software Family (for HPC and DL)



* Upcoming

Strong Procedure for Design, Development and Release

- Research is done for exploring new designs
- Designs are first presented to conference/journal publications
- Best performing designs are incorporated into the codebase
- 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

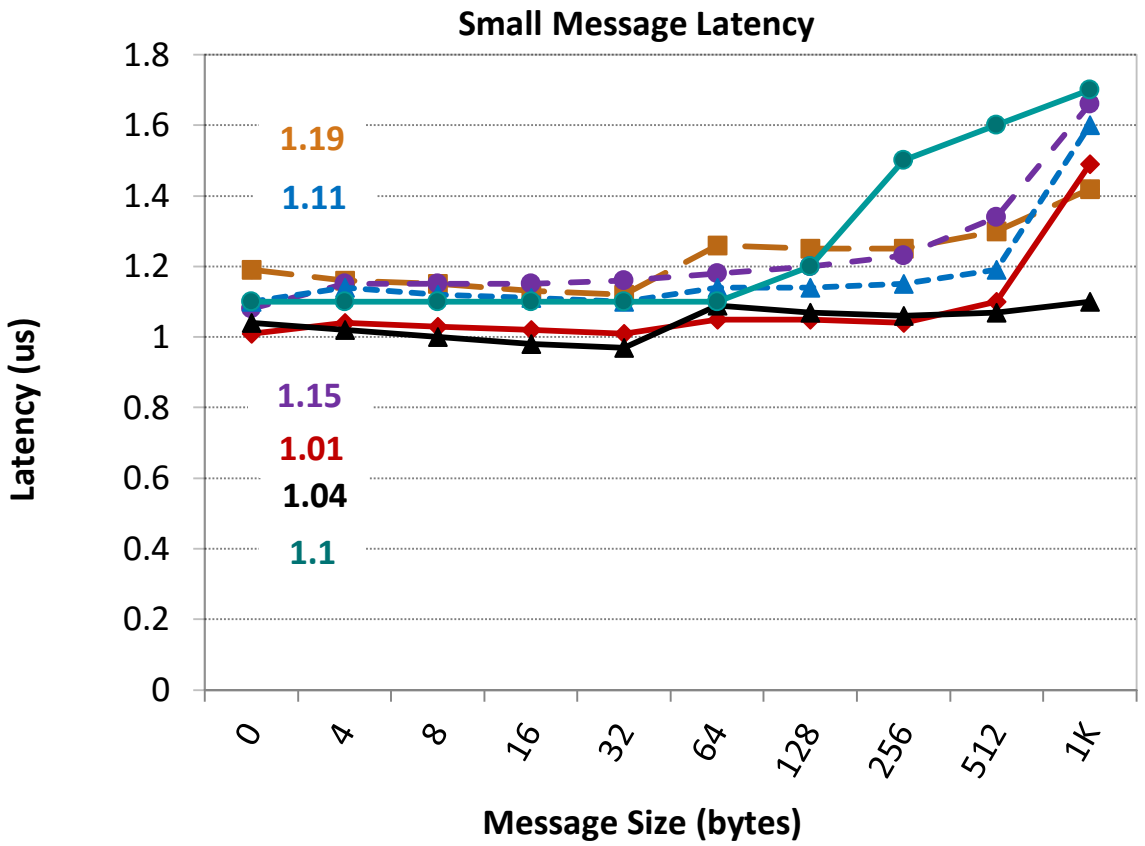
MVAPICH2 2.3.2

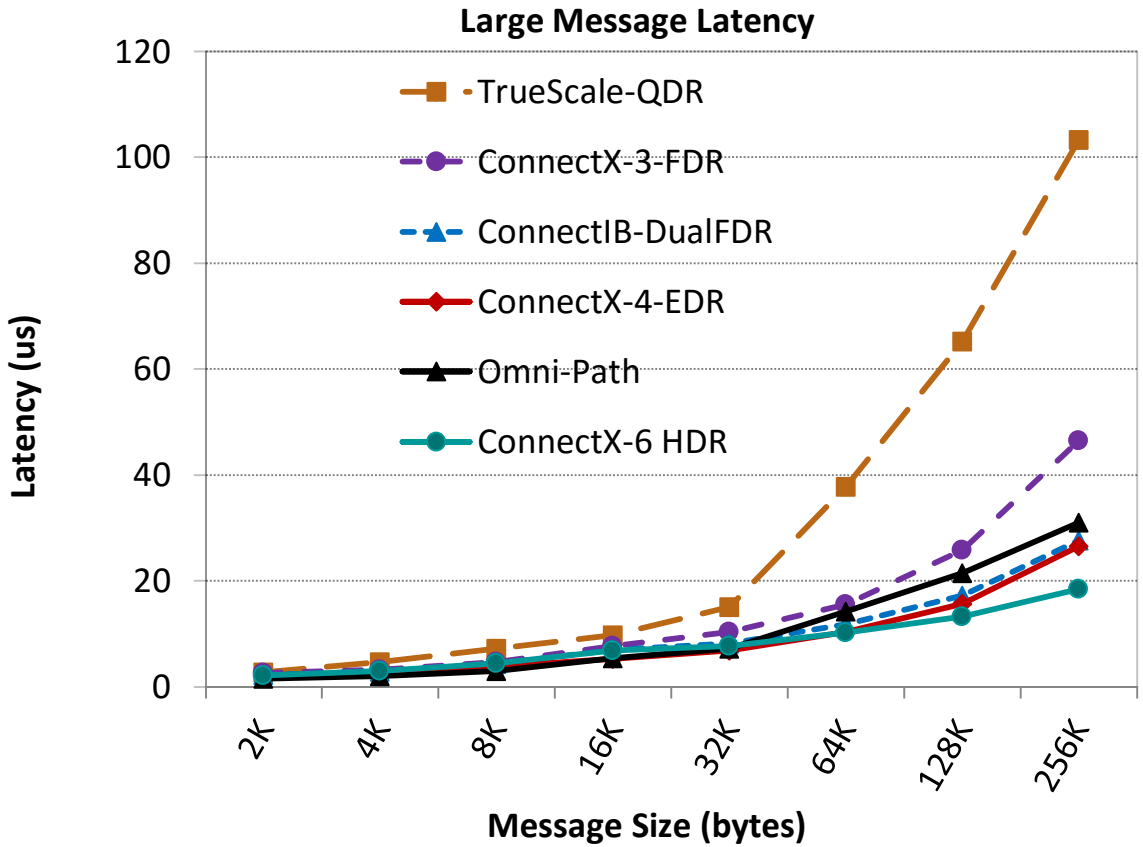
- **Released on 08/09/2019**
- Major Features and Enhancements
  - **Improved performance for inter-node communication**
  - **Improved performance for Gather, Reduce, and Allreduce with cyclic hostfile**
  - **- Thanks to X-ScaleSolutions for the patch**
  - **Improved performance for intra-node point-to-point communication**
  - **Add support for Mellanox HDR adapters**
  - **Add support for Cascade lake systems**
  - **Add support for Microsoft Azure platform**
    - **Enhanced point-to-point and collective tuning for Microsoft Azure**
  - **Add support for new NUMA-aware hybrid binding policy**
  - **Add support for AMD EPYC Rome architecture**
  - **Improved multi-rail selection logic**
  - **Enhanced heterogeneity detection logic**
  - **Enhanced point-to-point and collective tuning for AMD EPYC Rome, Frontera@TACC, Mayer@Sandia, Pitzer@OSC, Summit@ORNL, Lassen@LLNL, and Sierra@LLNL systems**
  - **Add multiple PVARs and CVARs for point-to-point and collective operations**

Highlights of MVAPICH2 2.3.2-GA Release

- Support for highly-efficient inter-node and intra-node communication
- Scalable Start-up
- Optimized Collectives using SHArP and Multi-Leaders
- Support for OpenPOWER and ARM architectures
- Performance Engineering with MPI-T
- Application Scalability and Best Practices

One-way Latency: MPI over IB with MVAPICH2





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

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

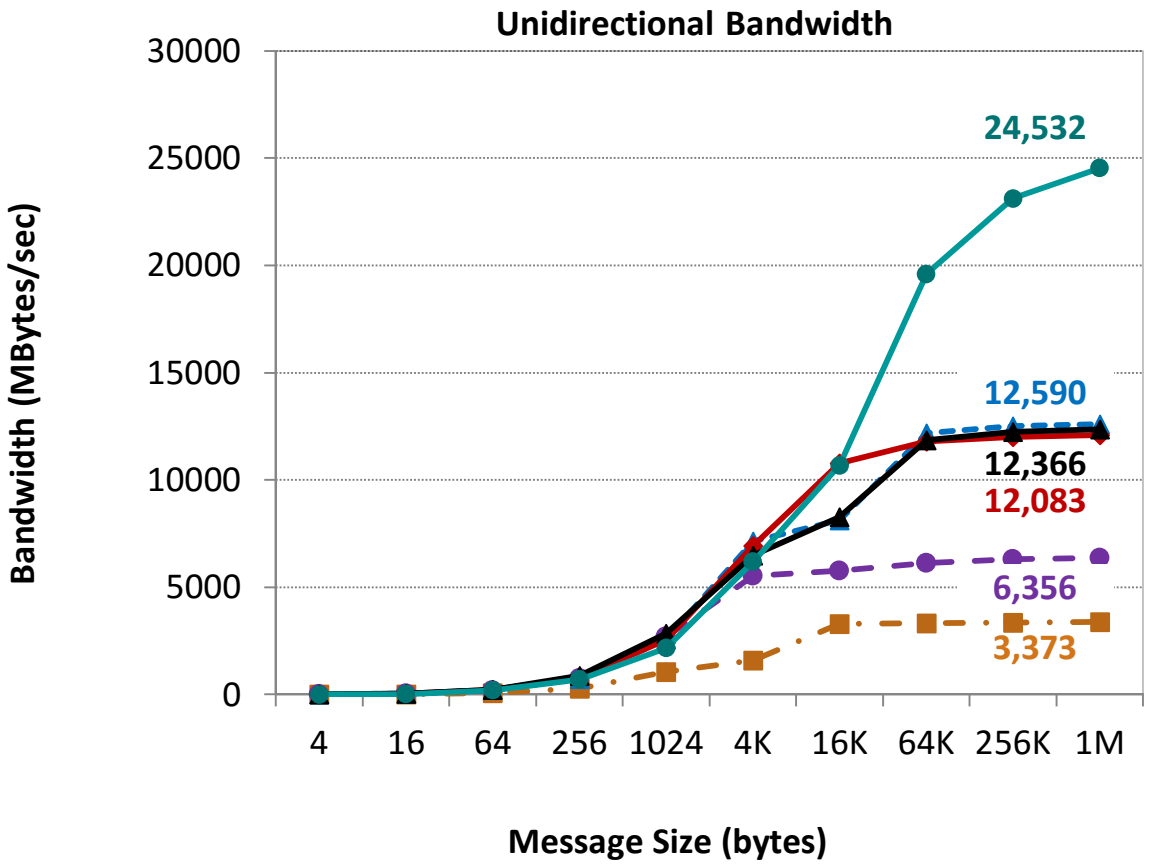
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

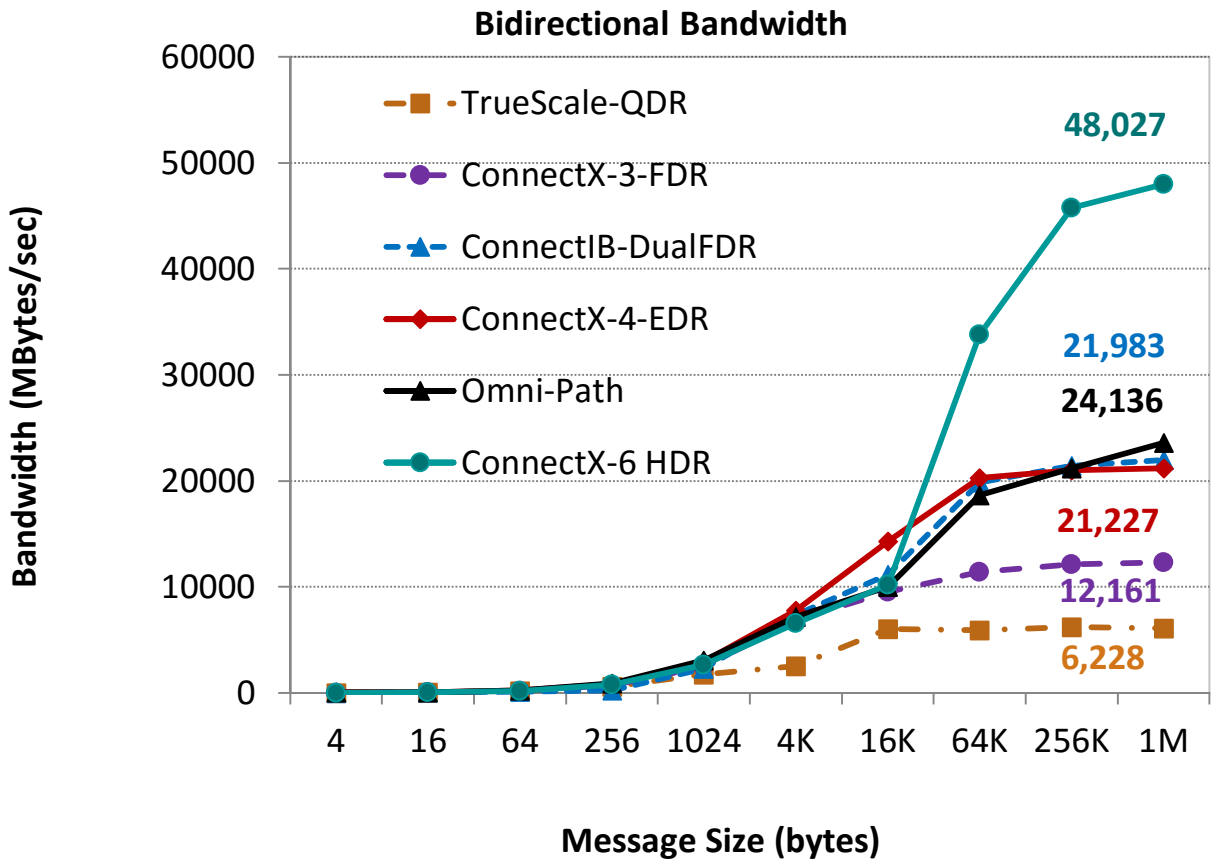
ConnectX-4-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch

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

ConnectX-6-HDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch

Bandwidth: MPI over IB with MVAPICH2





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

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

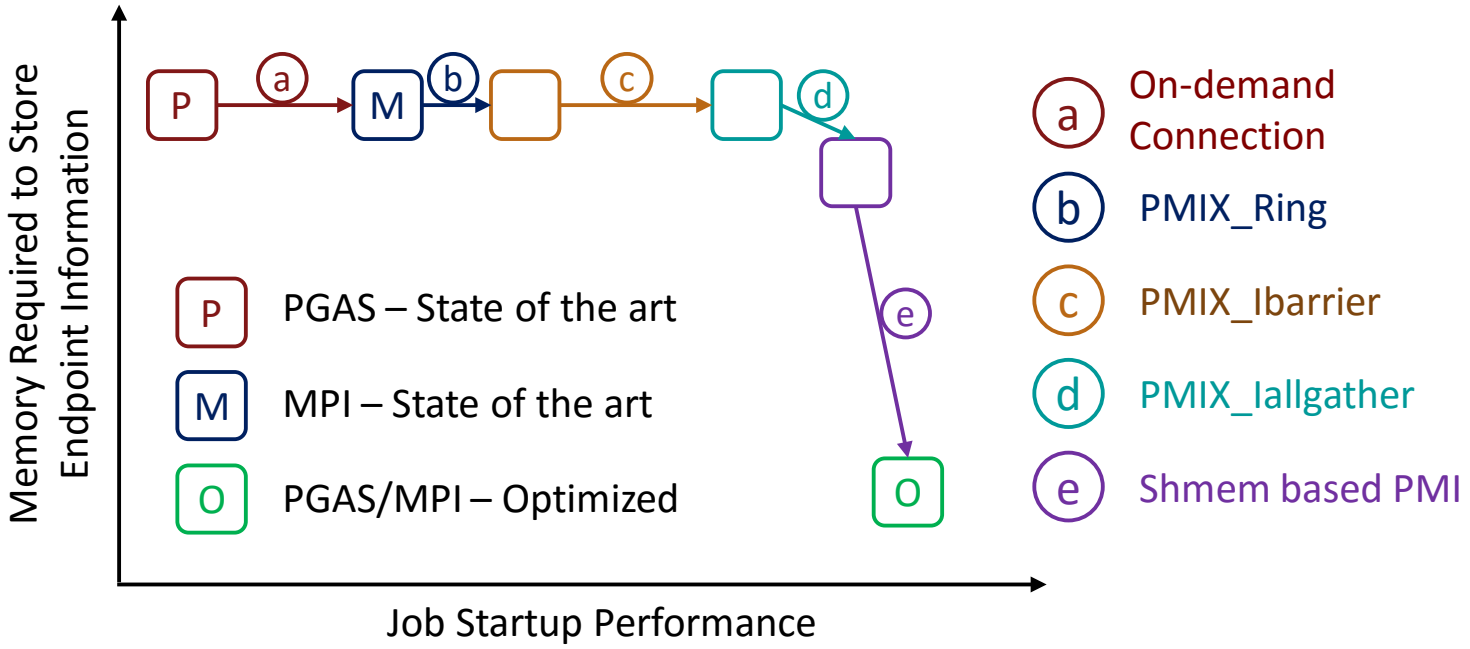
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

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

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

ConnectX-6-HDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch

Towards High Performance and Scalable Startup at Exascale



- Near-constant MPI and OpenSHMEM initialization time at any process count
- 10x and 30x improvement in startup time of MPI and OpenSHMEM respectively at 16,384 processes
- Memory consumption reduced for remote endpoint information by O(processes per node)
- 1GB Memory saved per node with 1M processes and 16 processes per node

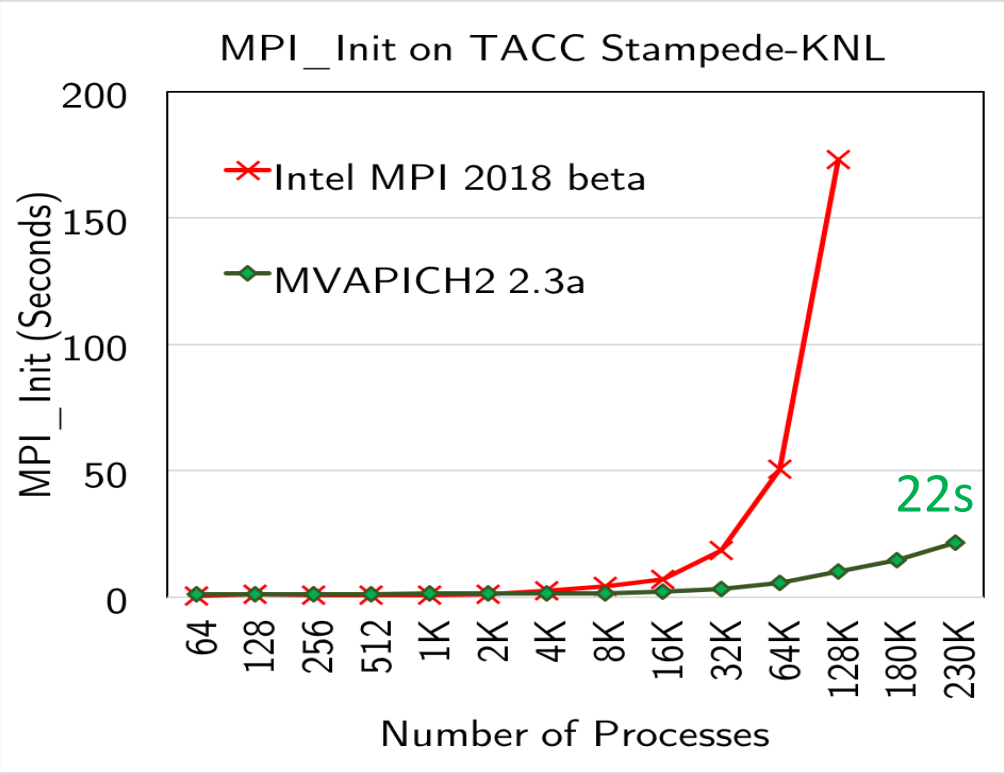
(a) **On-demand Connection Management for OpenSHMEM and OpenSHMEM+MPI.** S. Chakraborty, H. Subramoni, J. Perkins, A. A. Awan, and D K Panda, 20th International Workshop on High-level Parallel Programming Models and Supportive Environments (HIPS '15)

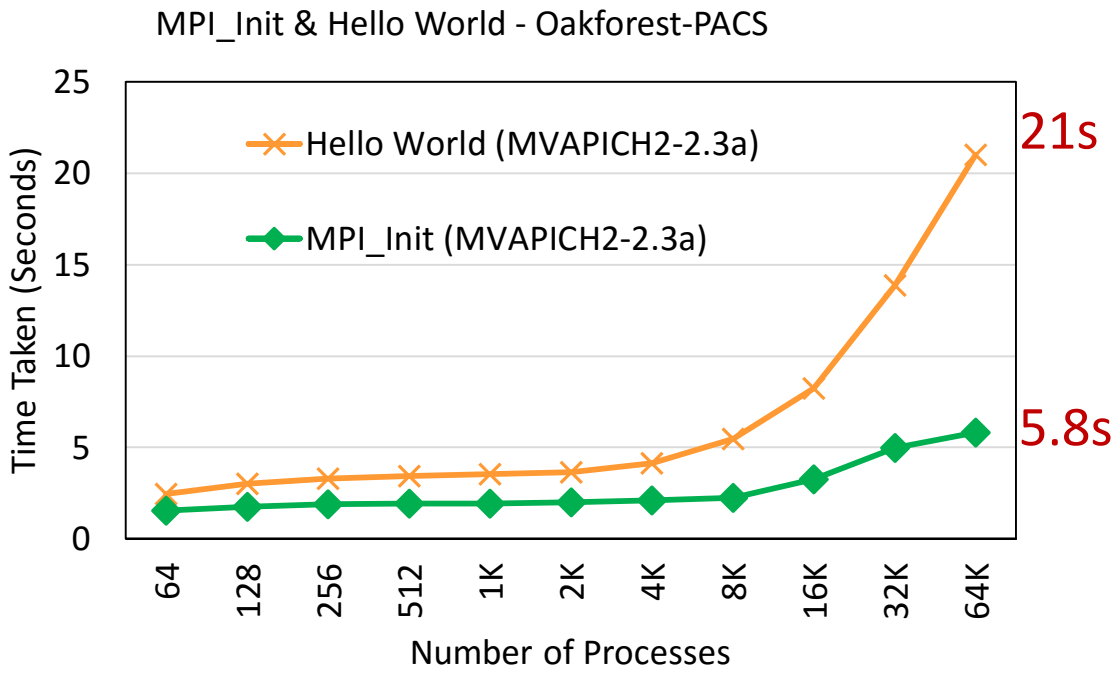
(b) **PMI Extensions for Scalable MPI Startup.** S. Chakraborty, H. Subramoni, A. Moody, J. Perkins, M. Arnold, and D K Panda, Proceedings of the 21st European MPI Users' Group Meeting (EuroMPI/Asia '14)

(c) (d) **Non-blocking PMI Extensions for Fast MPI Startup.** S. Chakraborty, H. Subramoni, A. Moody, A. Venkatesh, J. Perkins, and D K Panda, 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid '15)

(e) **SHMEMPMI – Shared Memory based PMI for Improved Performance and Scalability.** S. Chakraborty, H. Subramoni, J. Perkins, and D K Panda, 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid '16)

Startup Performance on KNL + Omni-Path

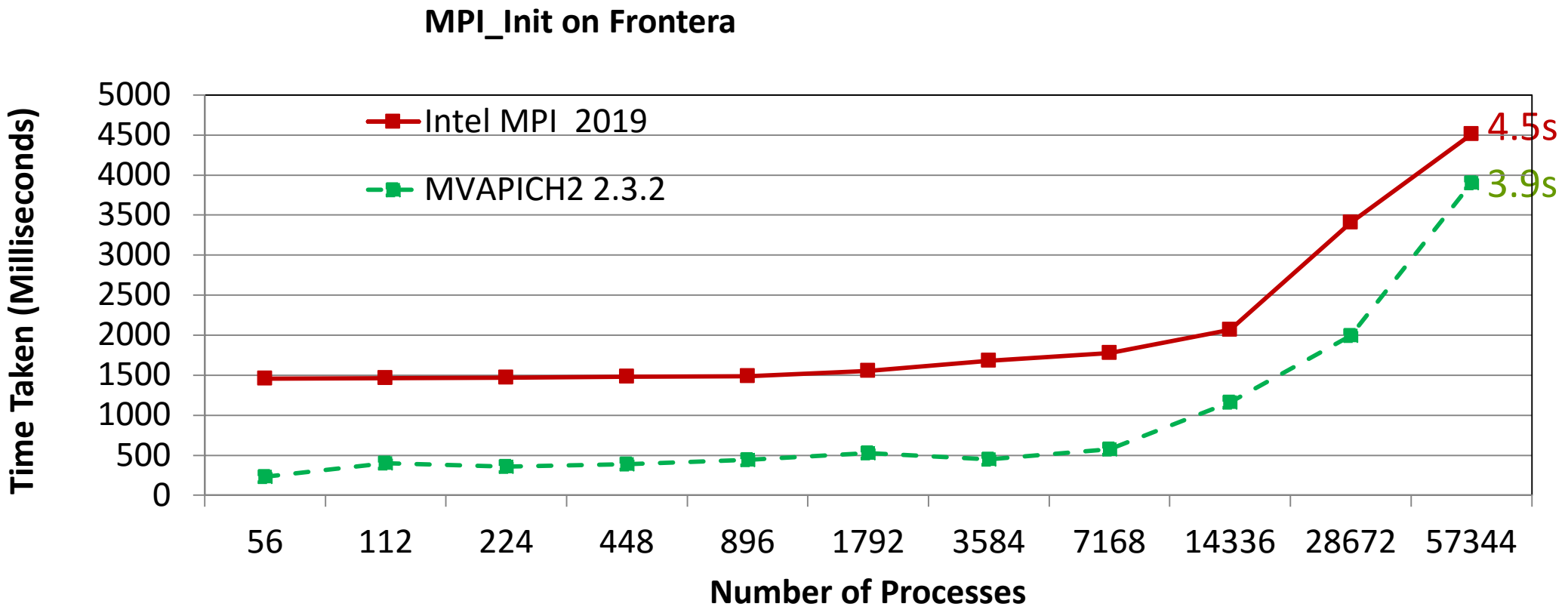




- MPI_Init takes 22 seconds on 229,376 processes on 3,584 KNL nodes (Stampede2 – Full scale)
- 8.8 times faster than Intel MPI at 128K processes (Courtesy: TACC)
- At 64K processes, MPI_Init and Hello World takes 5.8s and 21s respectively (Oakforest-PACS)
- All numbers reported with 64 processes per node

New designs available since MVAPICH2-2.3a and as patch for SLURM-15.08.8 and SLURM-16.05.1

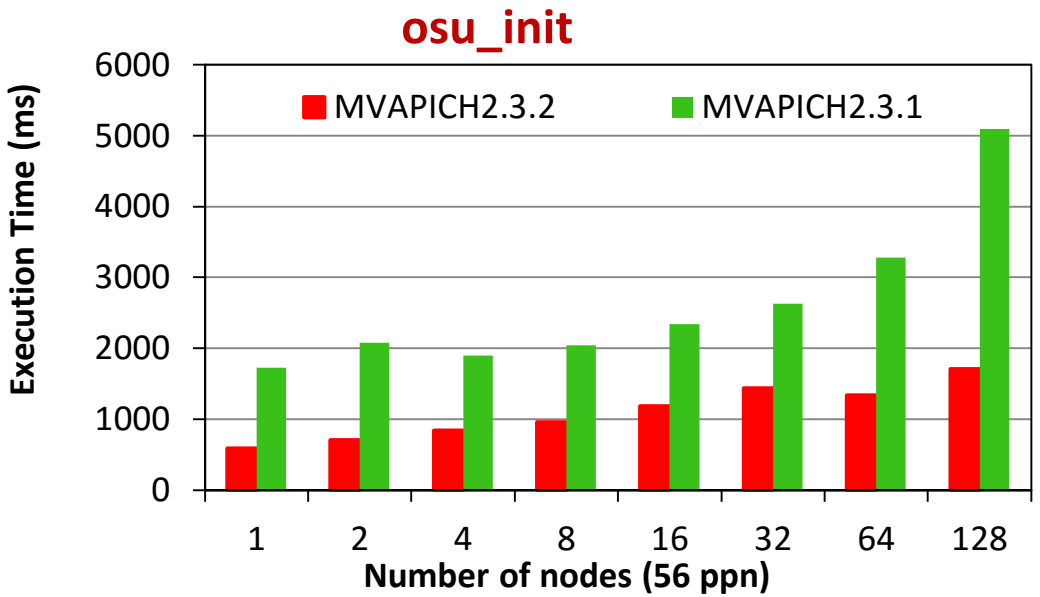
Startup Performance on TACC Frontera

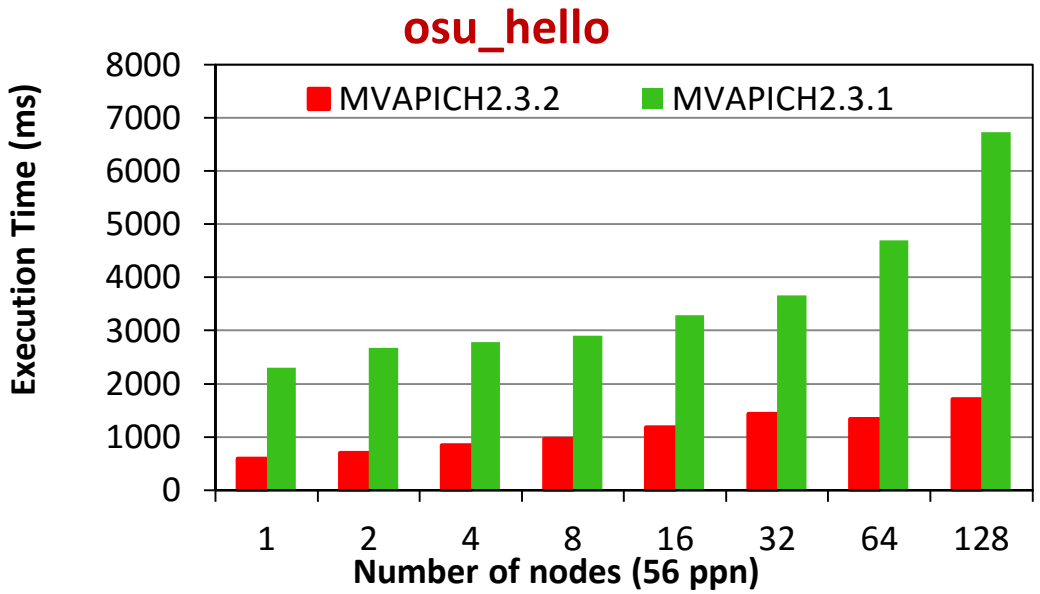


- MPI_Init takes 3.9 seconds on 57,344 processes on 1,024 nodes
- All numbers reported with 56 processes per node

New designs available in MVAPICH2-2.3.2

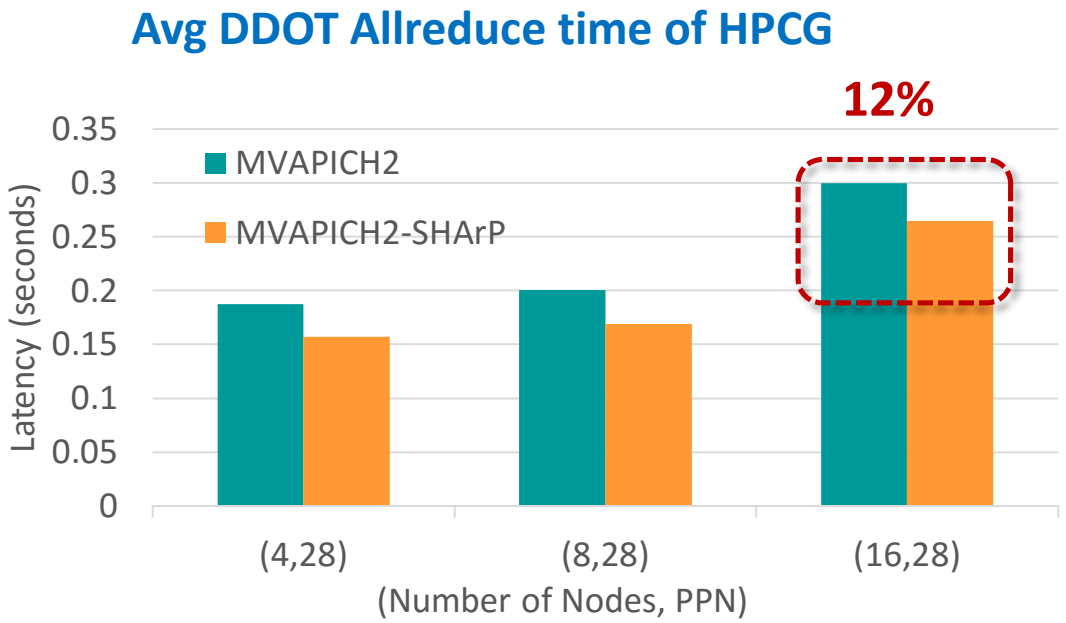
Startup Performance on TACC Frontera MVAPICH2 2.3.1 vs 2.3.2





- MVAPICH2 2.3.2 significantly improves performance on top of MVAPICH2 2.3.1
- All numbers reported with 56 processes per node

Benefits of SHARP Allreduce at Application Level

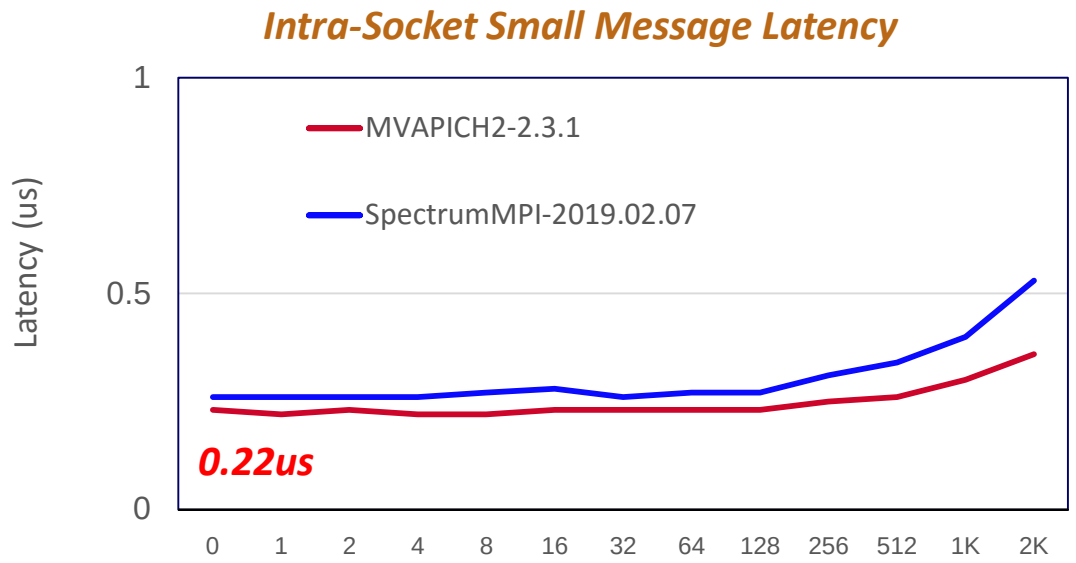


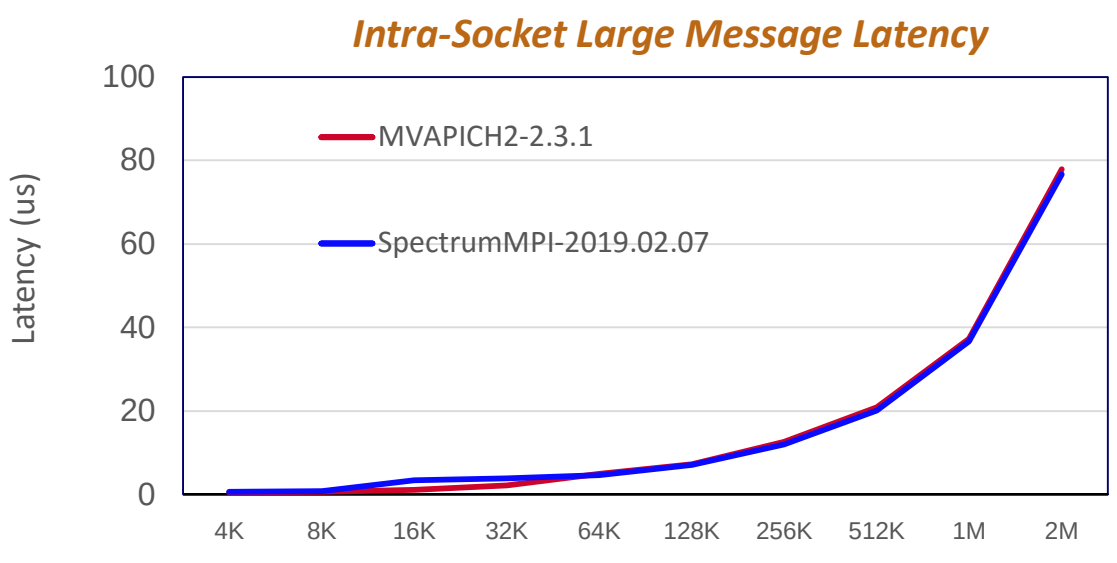
SHARP support available since MVAPICH2 2.3a

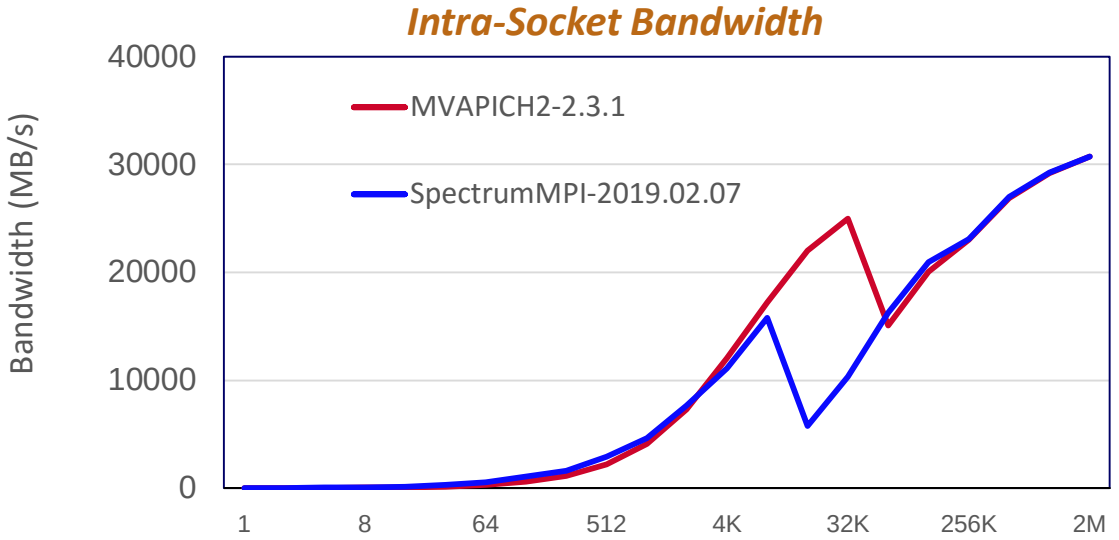
Parameter	Description	Default
MV2_ENABLE_SHARP=1	Enables SHARP-based collectives	Disabled
--enable-sharp	Configure flag to enable SHARP	Disabled

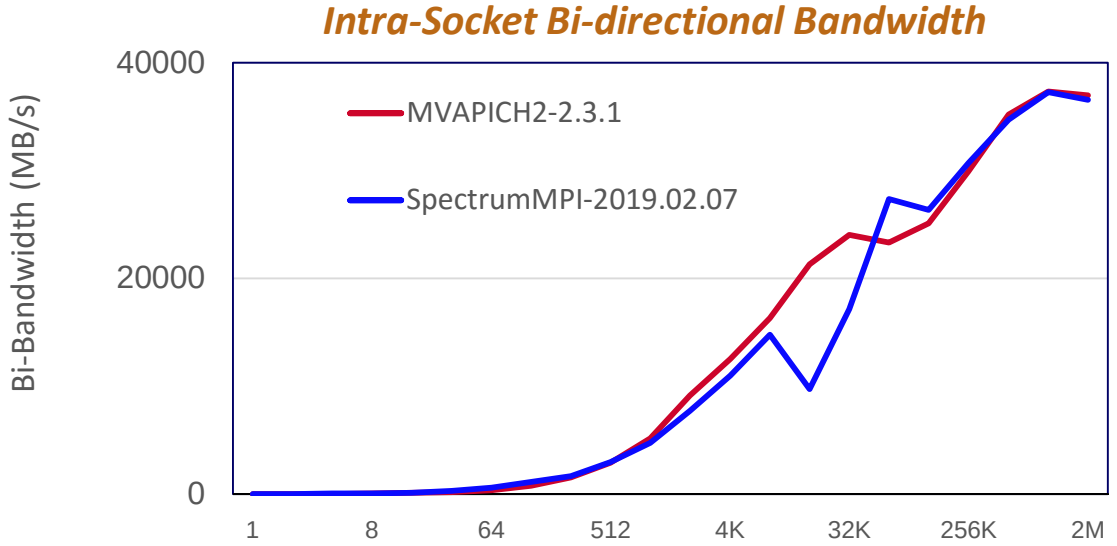
- **Refer to Running Collectives with Hardware based SHARP support section of MVAPICH2 user guide for more information**
- **http://mvapich.cse.ohio-state.edu/static/media/mvapich/mvapich2-2.3-userguide.html#x1-990006.26**

Intra-node Point-to-Point Performance on OpenPower



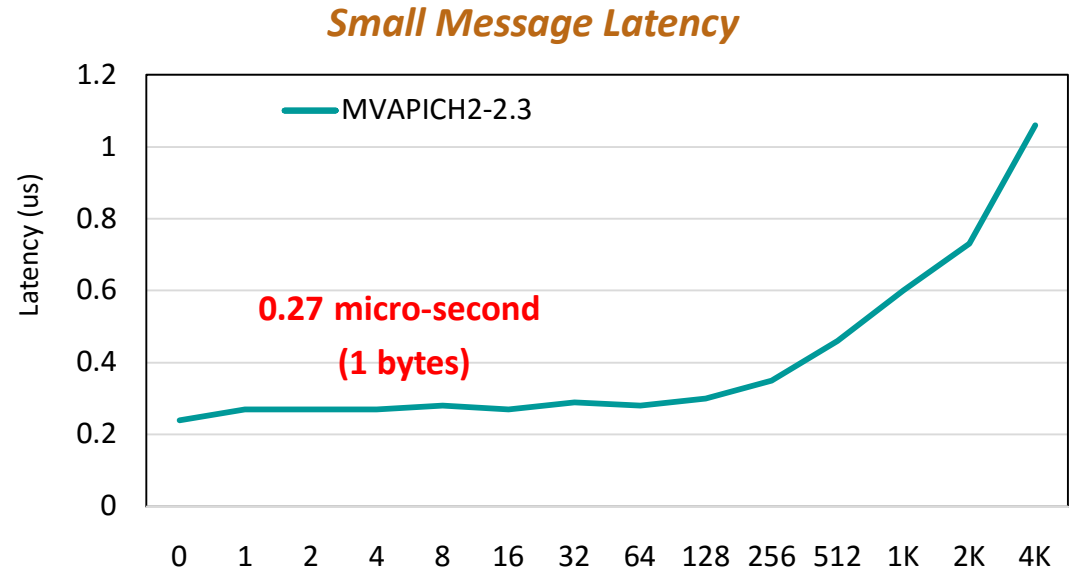


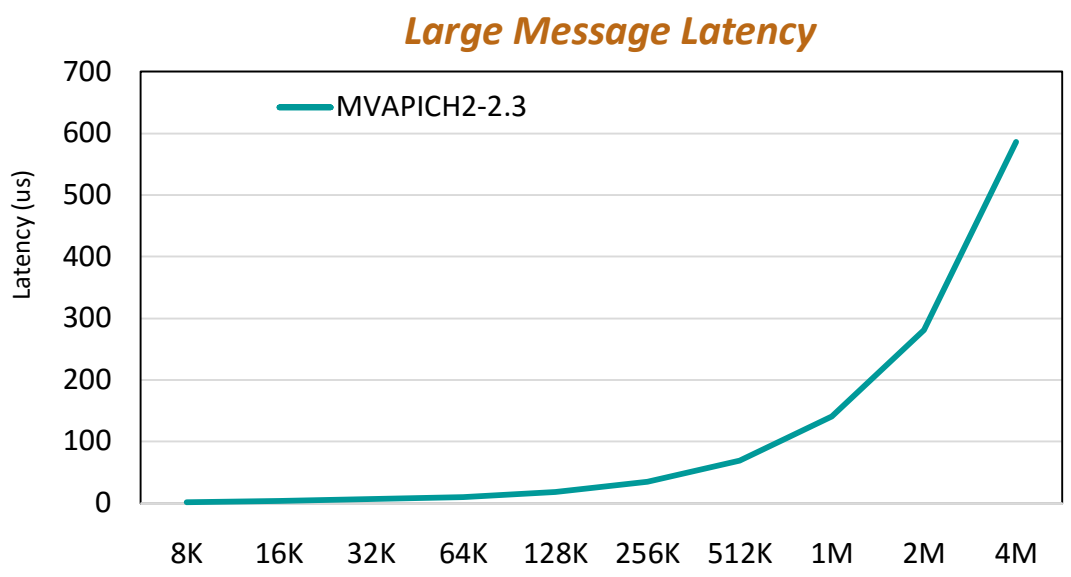


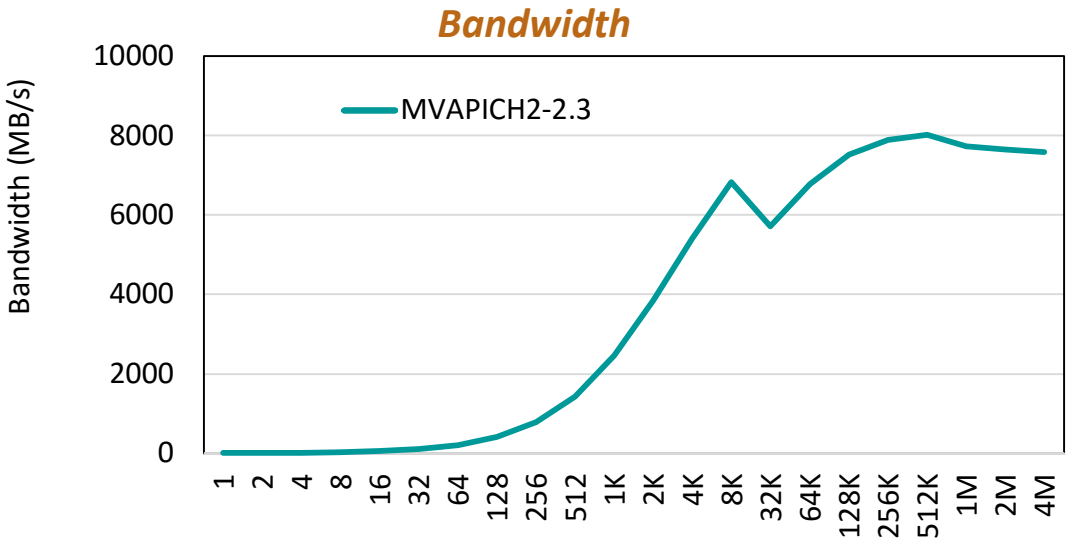


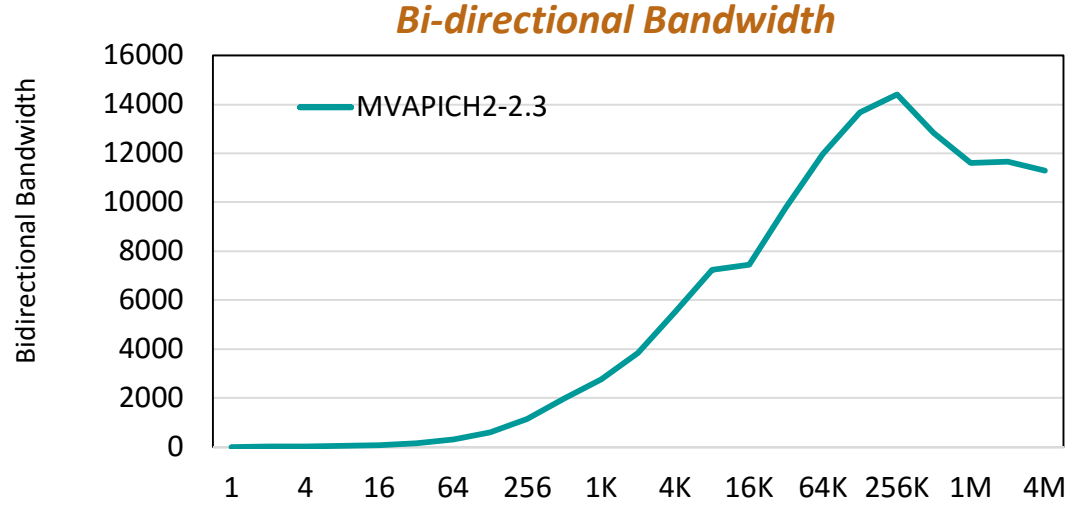
Platform: Two nodes of OpenPOWER (POWER9-ppc64le) CPU using Mellanox EDR (MT4121) HCA

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





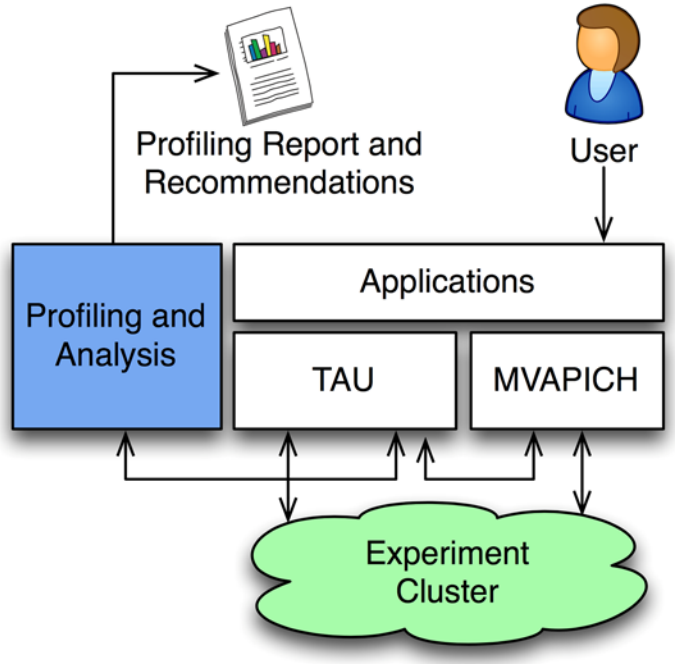




Platform: ARM Cortex A72 (aarch64) processor with 64 cores dual-socket CPU. Each socket contains 32 cores.

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 today and poster presentations



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

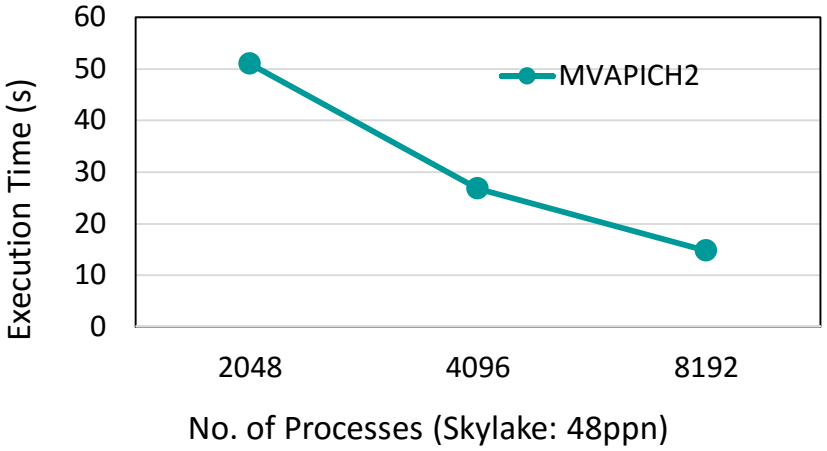
VBUF usage with CVAR based tuning as displayed by ParaProf

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

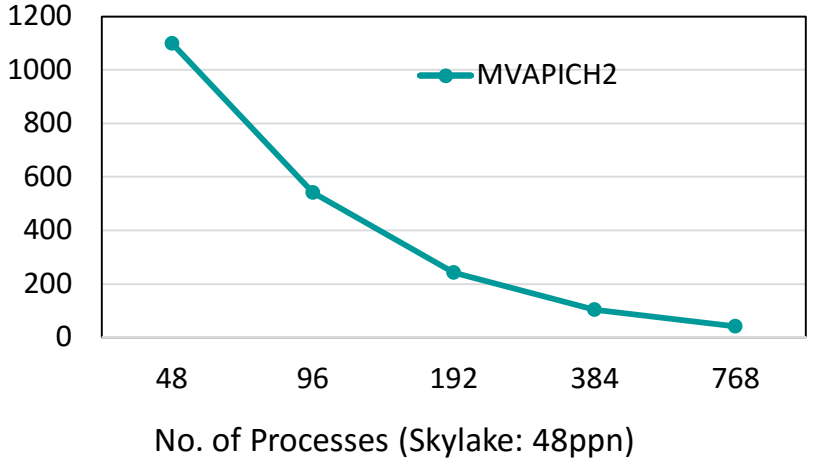
Name	MaxValue	MinValue	MeanValue	Std. Dev.	NumSamp...	Total
mv2_total_vbuf_memory (Total amount of memory in bytes used for VBUFs)	1,815,056	1,815,056	1,815,056	0	1	1,815,056
mv2_ud_vbuf_allocated (Number of UD VBUFs allocated)	0	0	0	0	0	0
mv2_ud_vbuf_available (Number of UD VBUFs available)	0	0	0	0	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

Application Scalability on Skylake and KNL

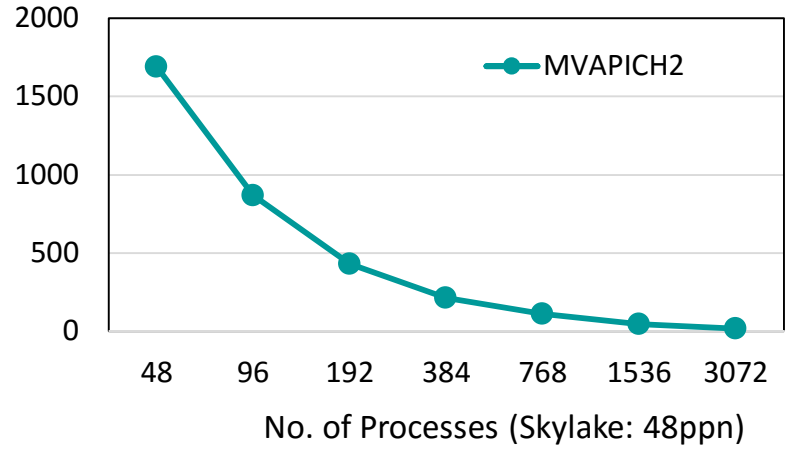
MiniFE (1300x1300x1300 ~ 910 GB)

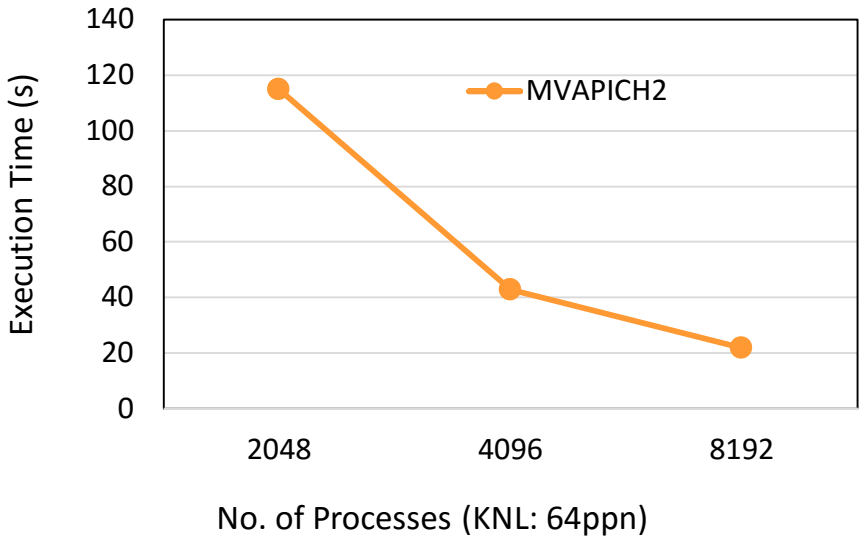


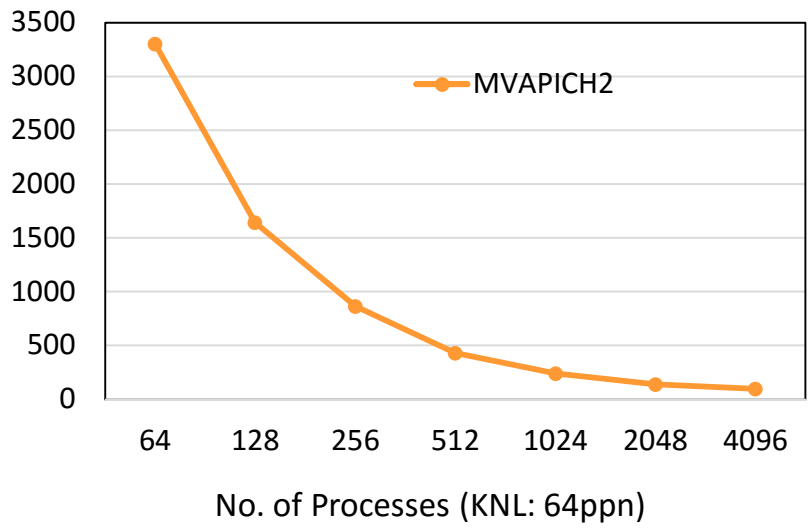
NEURON (YuEtAl2012)

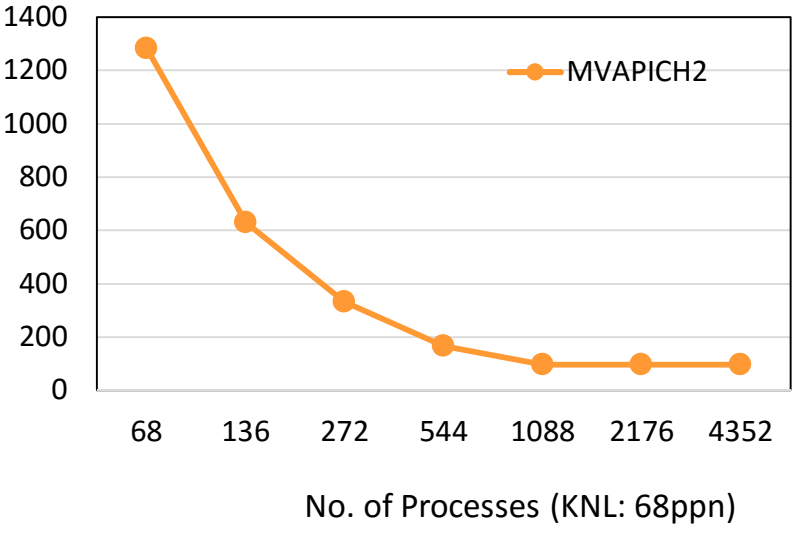


Cloverleaf (bm64) MPI+OpenMP, NUM_OMP_THREADS = 2









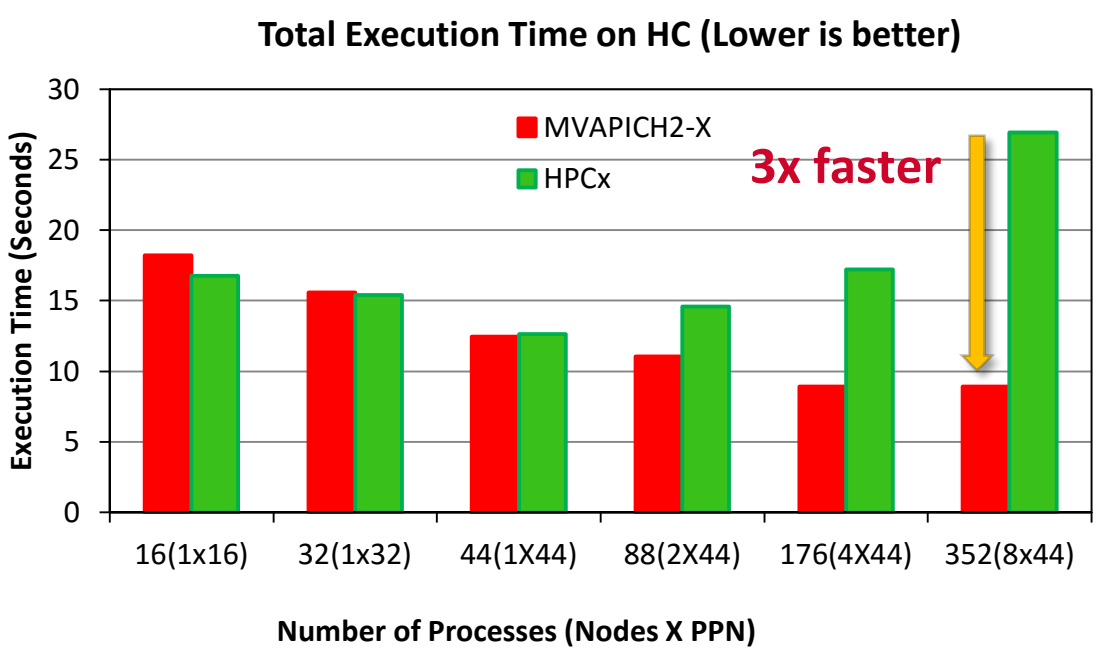
Courtesy: Mahidhar Tatineni @SDSC, Dong Ju (DJ) Choi@SDSC, and Samuel Khuvis@OSC ---- Testbed: TACC Stampede2 using MVAPICH2-2.3b

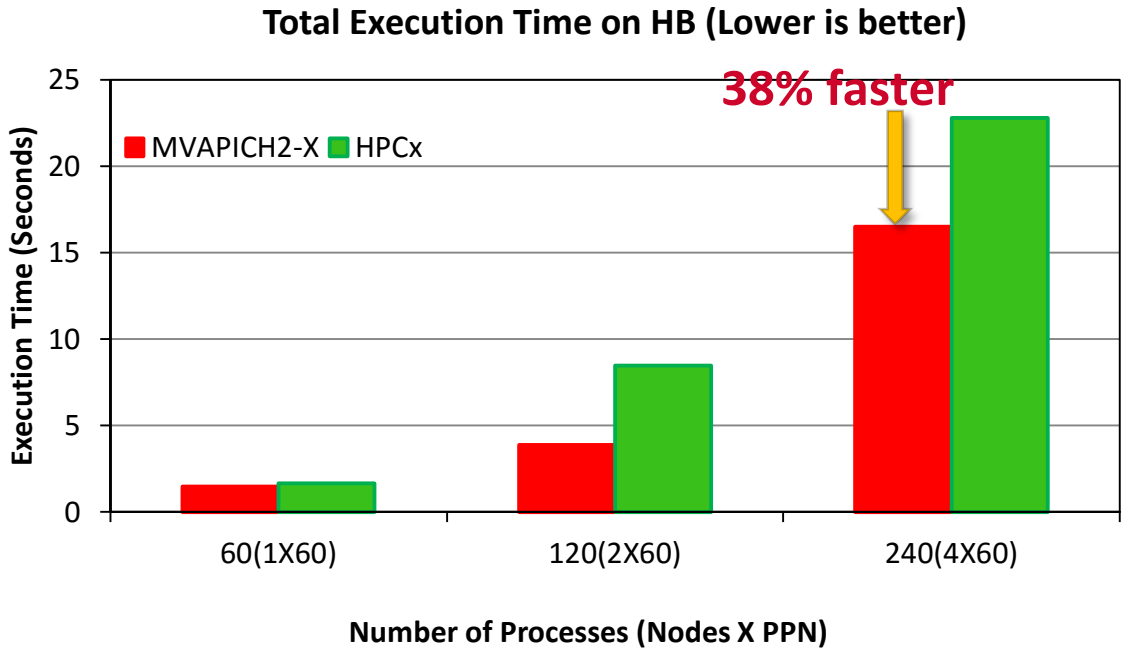
Runtime parameters: *MV2_SMPI_LENGTH_QUEUE=524288 PSM2_MQ_RNDV_SHM_THRESH=128K PSM2_MQ_RNDV_HFI_THRESH=128K*

MVAPICH2-Azure 2.3.2

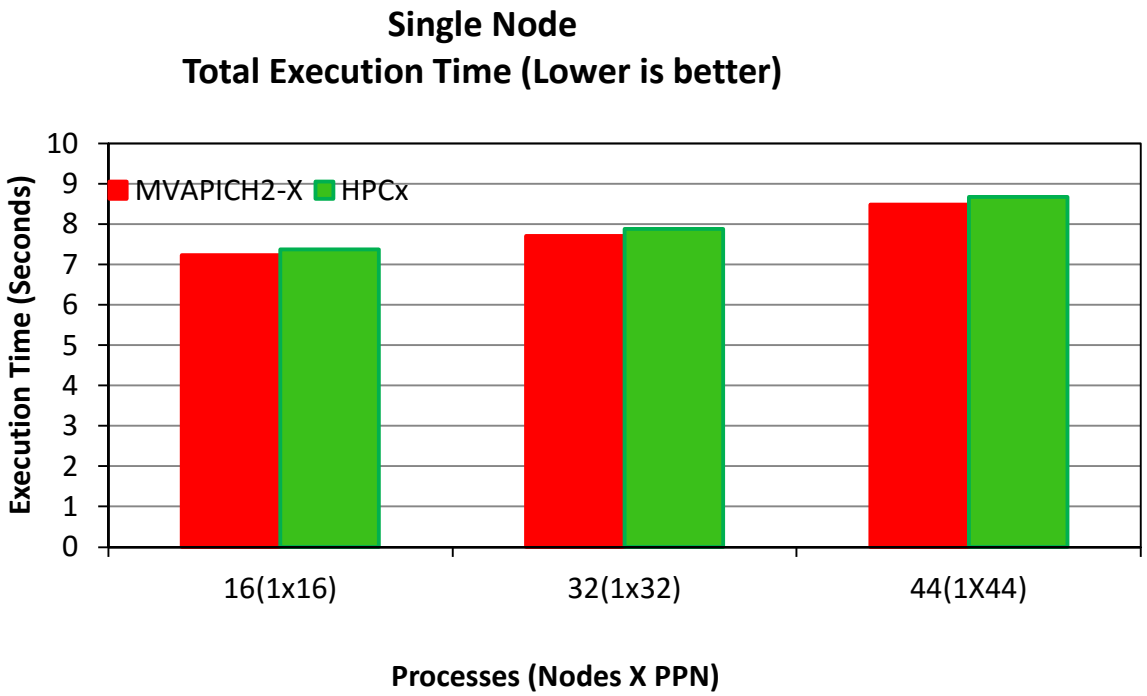
- **Released on 08/16/2019**
- Major Features and Enhancements
  - **Based on MVAPICH2-2.3.2**
  - **Enhanced tuning for point-to-point and collective operations**
  - **Targeted for Azure HB & HC virtual machine instances**
  - **Flexibility for 'one-click' deployment**
  - **Tested with Azure HB &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

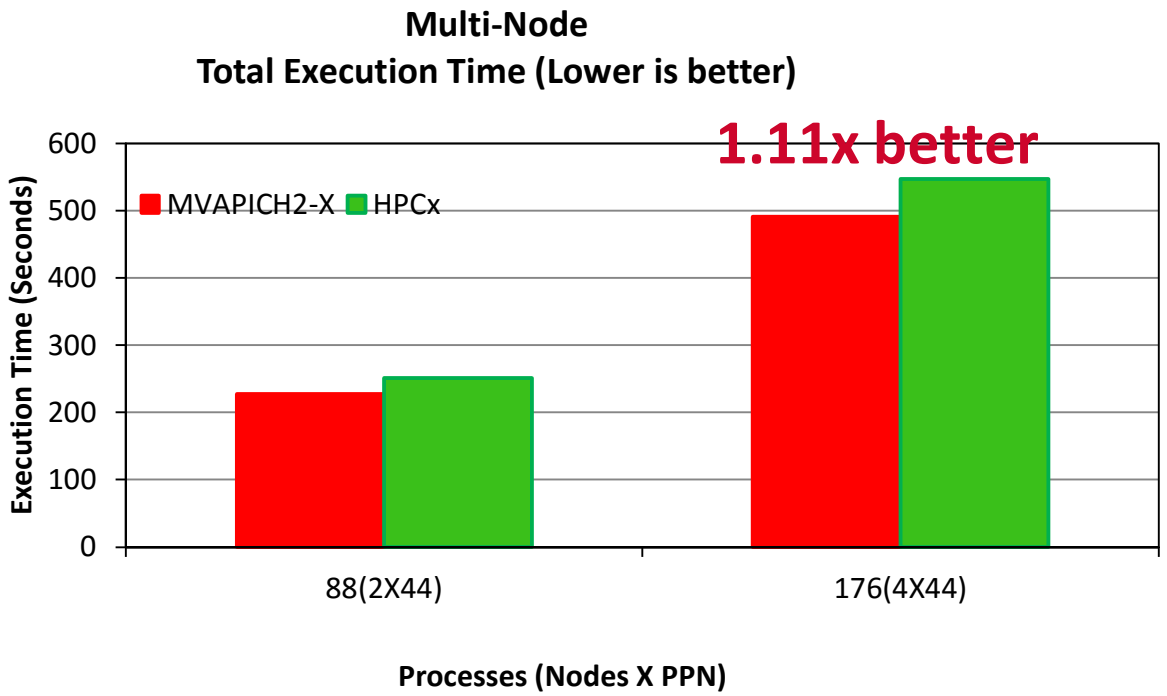
Performance of Radix





Performance of FDS (HC)





Part of input parameter: MESH IJK=5,5,5, XB=-1.0,0.0,-1.0,0.0,0.0,1.0, MULT_ID='mesh array'

MVAPICH2 Upcoming Features

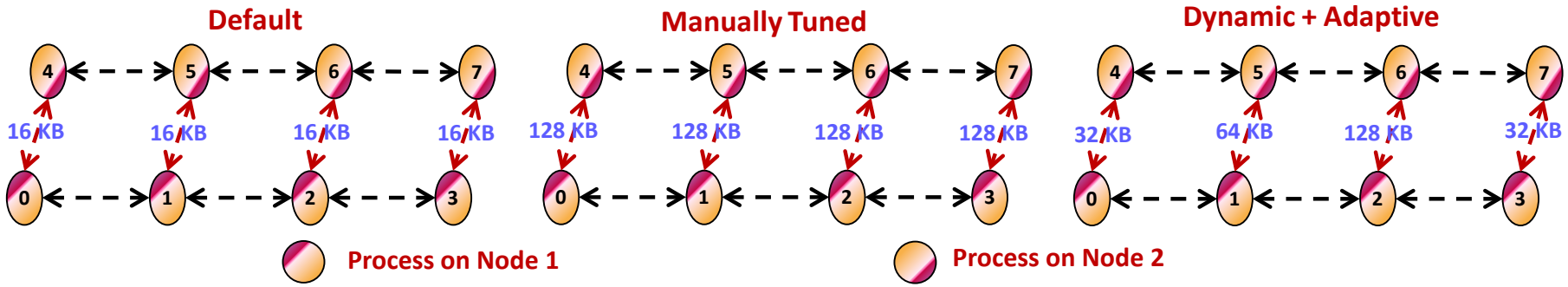
- Integration of SHARP2 and associated Collective Optimizations
- Communication optimizations on upcoming architectures
  - Intel Cooper Lake
  - AMD Rome
  - ARM
- Dynamic and Adaptive Communication Protocols

Dynamic and Adaptive MPI Point-to-point Communication Protocols

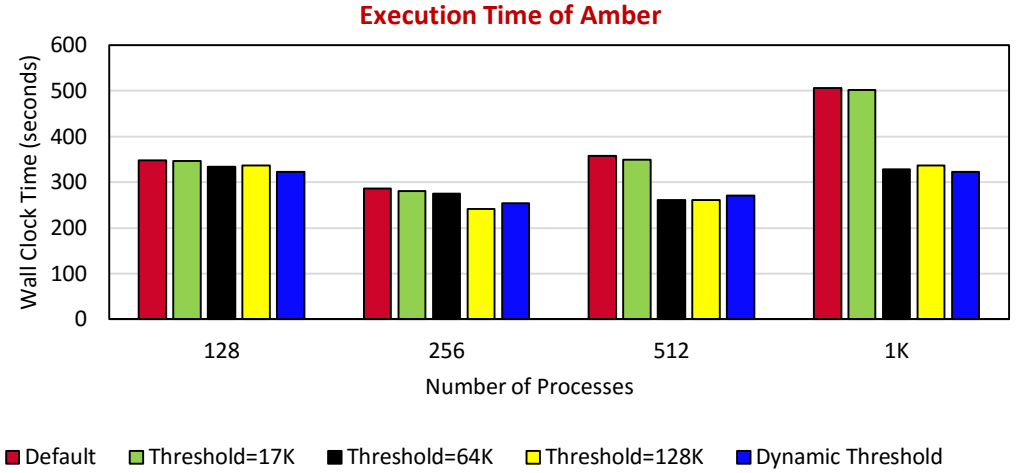
Desired Eager Threshold

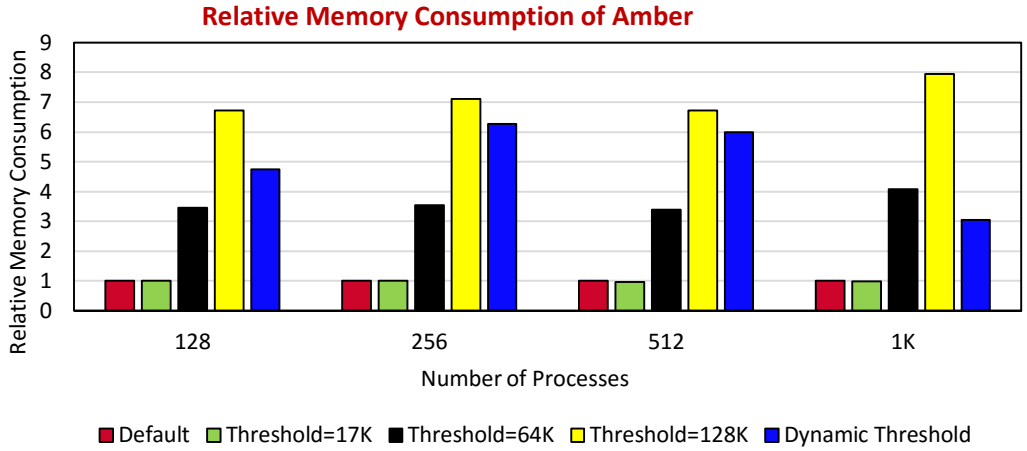
Process Pair	Eager Threshold (KB)
0 – 4	32
1 – 5	64
2 – 6	128
3 – 7	32

Eager Threshold for Example Communication Pattern with Different Designs



Default	Poor overlap; Low memory requirement	Low Performance; High Productivity
Manually Tuned	Good overlap; High memory requirement	High Performance; Low Productivity
Dynamic + Adaptive	Good overlap; Optimal memory requirement	High Performance; High Productivity



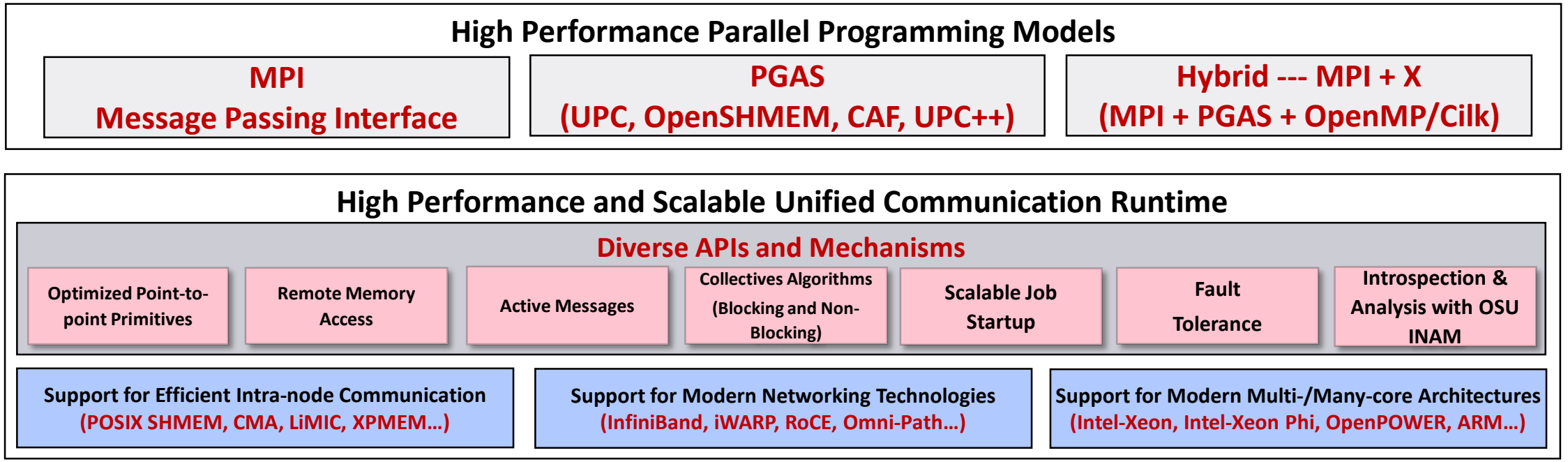


H. Subramoni, S. Chakraborty, D. K. Panda, Designing Dynamic & Adaptive MPI Point-to-Point Communication Protocols for Efficient Overlap of Computation & Communication, ISC'17 - Best Paper

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-X for Hybrid MPI + PGAS Applications



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

MVAPICH2-X 2.3rc2

- **Released on 03/01/2019**
- Major Features and Enhancements
  - **MPI Features**
  - **Based on MVAPICH2 2.3.1**
    - **OFA-IB-CH3, OFA-IB-RoCE, PSM-CH3, and PSM2-CH3 interfaces**
  - **MPI (Advanced) Features**
    - Improved performance of large message communication
    - Support for advanced co-operative (COOP) rendezvous protocols in SMP channel
      - OFA-IB-CH3 and OFA-IB-RoCE interfaces
    - Support for RGET, RPUT, and COOP protocols for CMA and XPMEM
      - OFA-IB-CH3 and OFA-IB-RoCE interfaces
    - Support for load balanced and dynamic rendezvous protocol selection
      - OFA-IB-CH3 and OFA-IB-RoCE interfaces
    - Support for XPMEM-based MPI collective operations (Broadcast, Gather, Scatter, Allgather)
      - OFA-IB-CH3, OFA-IB-RoCE, PSM-CH3, and PSM2-CH3 interfaces
    - Extend support for XPMEM-based MPI collective operations (Reduce and All-Reduce for PSM-CH3 and PSM2-CH3 interfaces
    - Improved connection establishment for DC transport
      - OFA-IB-CH3 interface
    - Add improved Alltoallv algorithm for small messages
    - OFA-IB-CH3, OFA-IB-RoCE, PSM-CH3, and PSM2-CH3 interfaces
  - **OpenSHMEM Features**
    - Support for XPMEM-based collective operations (Broadcast, Collect, Reduce_all, Reduce, Scatter, Gather)
  - **UPC Features**
    - Support for XPMEM-based collective operations (Broadcast, Collect, Scatter, Gather)
  - **UPC++ Features**
    - Support for XPMEM-based collective operations (Broadcast, Collect, Scatter, Gather)
  - **Unified Runtime Features**
    - Based on MVAPICH2 2.3.1 (OFA-IB-CH3 interface). All the runtime features enabled by default in OFA-IB-CH3 and OFA-IB-RoCE interface of MVAPICH2 2.3.1 are available in MVAPICH2-X 2.3rc2

MVAPICH2-X Feature Table

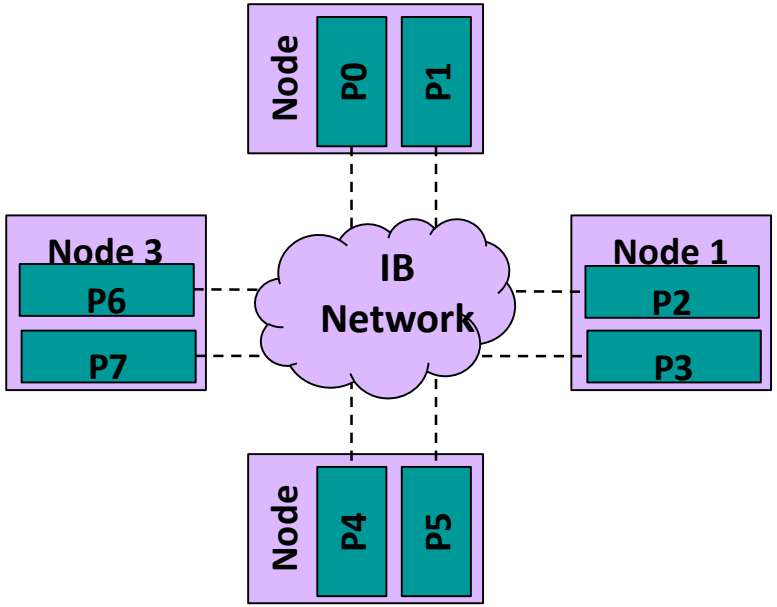
Features for InfiniBand (OFA-IB-CH3) and RoCE (OFA-RoCE-CH3)	Basic	Basic-XPMEM	Intermediate	Advanced
Architecture Specific Point-to-point and Collective Optimizations for x86, OpenPOWER, and ARM	✓	✓	✓	✓
Optimized Support for PGAS models (UPC, UPC++, OpenSHMEM, CAF) and Hybrid MPI+PGAS models	✓	✓	✓	✓
CMA-Aware Collectives	✓	✓	✓	✓
Optimized Asynchronous Progress*	✓	✓	✓	✓
InfiniBand Hardware Multicast-based MPI_Bcast*+	✓	✓	✓	✓
OSU InfiniBand Network Analysis and Monitoring (INAM)*+				✓
XPMEM-based Point-to-Point and Collectives		✓	✓	✓
Direct Connected (DC) Transport Protocol*+			✓	✓
User mode Memory Registration (UMR)*+				✓
On Demand Paging (ODP)*+				✓
Core-direct based Collective Offload*+				✓
SHARP-based Collective Offload*+				✓

- *** indicates disabled by default at runtime. Must use appropriate environment variable in MVAPICH2-X user guide to enable it.**
- **+ indicates features only tested with InfiniBand network**

Overview of Some of the MVAPICH2-X Features

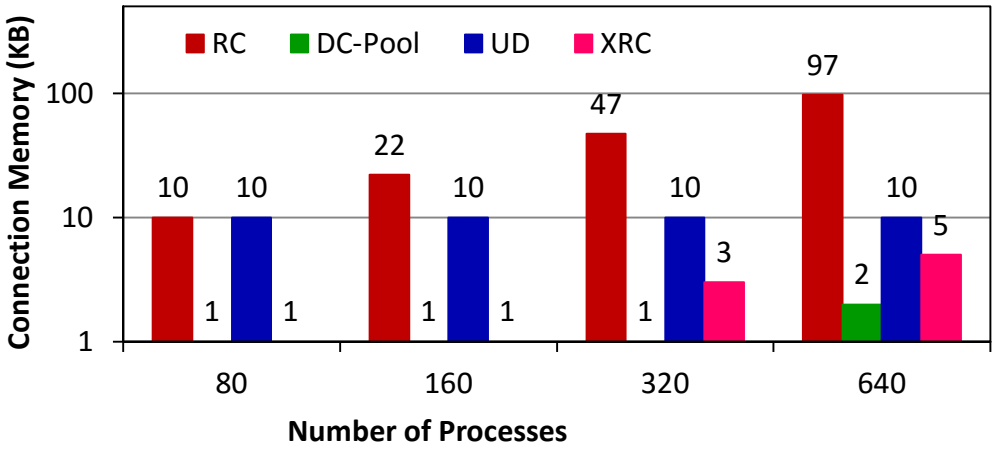
- Direct Connect (DC) Transport
- Co-operative Rendezvous Protocol
- Advanced All-reduce with SHARP
- CMA-based Collectives
- Asynchronous Progress
- XPMEM-based Reduction Collectives
- XPMEM-based Non-reduction Collectives
- Optimized Collective Communication and Advanced Transport Protocols
- PGAS and Hybrid MPI+PGAS Support

Minimizing Memory Footprint by Direct Connect (DC) Transport

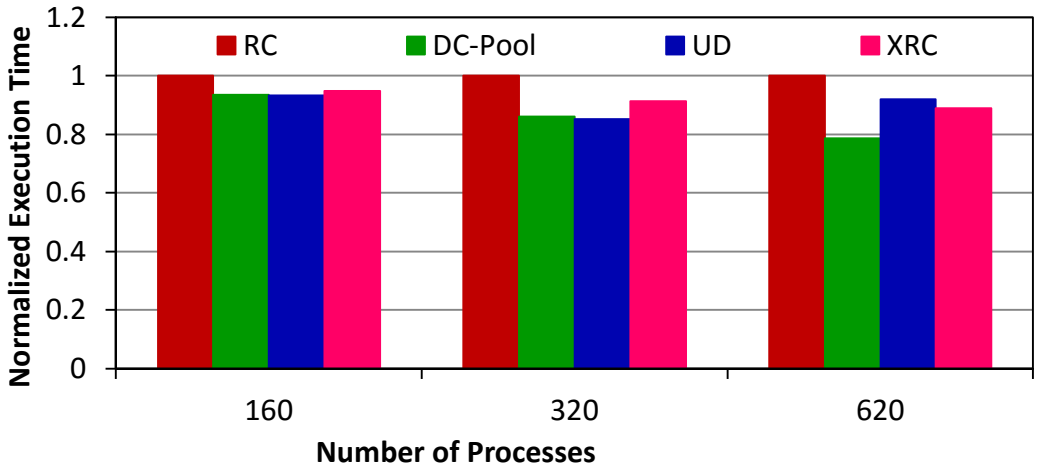


- Constant connection cost (*One QP for any peer*)
- Full Feature Set (RDMA, Atomics etc)
- Separate objects for send (DC Initiator) and receive (DC Target)
  - DC Target identified by "DCT Number"
  - Messages routed with (DCT Number, LID)
  - Requires same "DC Key" to enable communication
- Available since MVAPICH2-X 2.2a

Memory Footprint for Alltoall



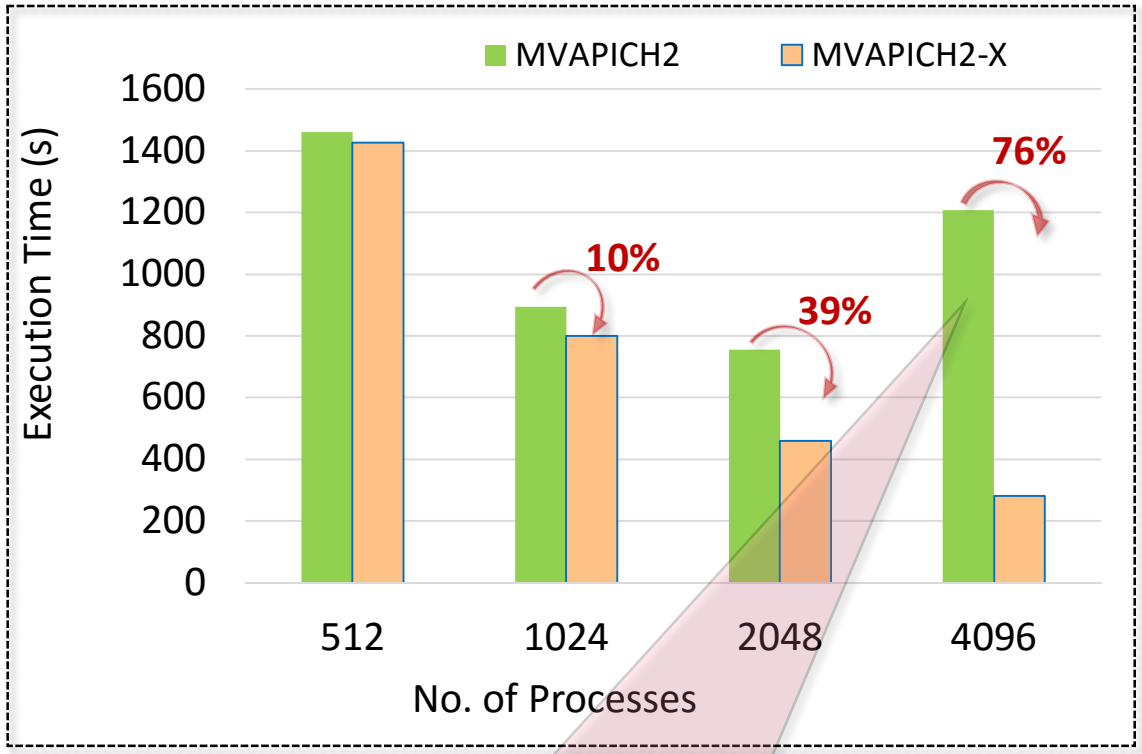
NAMD - Apoa1: Large data set



H. Subramoni, K. Hamidouche, A. Venkatesh, S. Chakraborty and D. K. Panda, Designing MPI Library with Dynamic Connected Transport (DCT) of InfiniBand : Early Experiences. IEEE International Supercomputing Conference (ISC '14)

Impact of DC Transport Protocol on Neuron

Neuron with YuEtAl2012

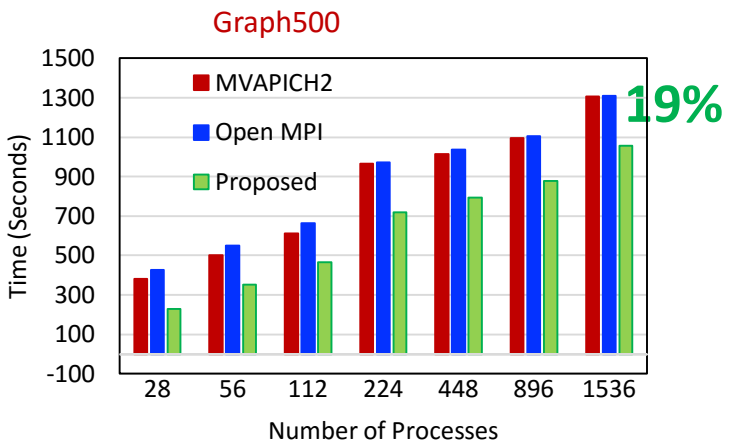


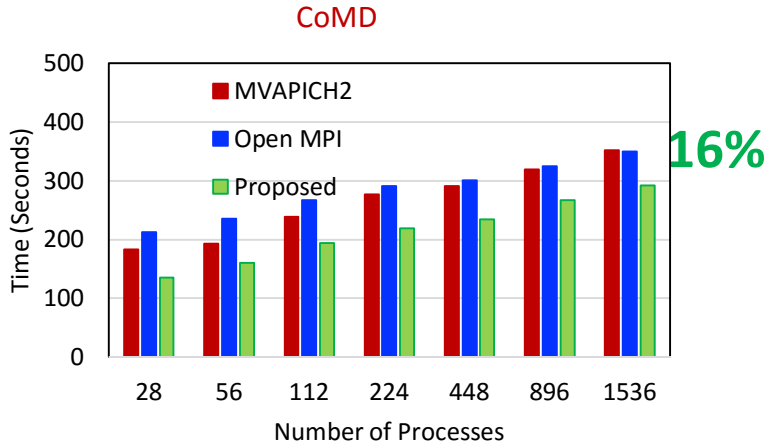
Overhead of RC protocol for connection establishment and communication

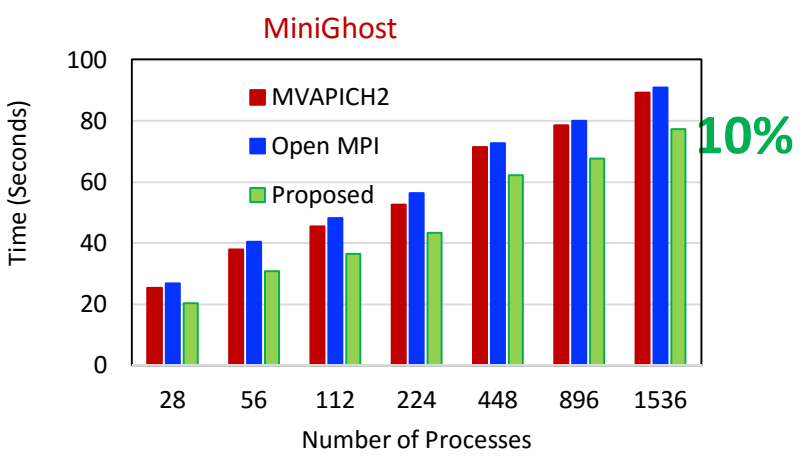
- Up to 76% benefits over MVAPICH2 for Neuron using Direct Connected transport protocol at scale
  - VERSION 7.6.2 master (f5a1284) 2018-08-15
- Numbers taken on bbpv2.epfl.ch
  - Knights Landing nodes with 64 ppn
  - ./x86_64/special -mpi -c stop_time=2000 -c is_split=1 parinit.hoc
  - Used "runtime"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

Available from MVAPICH2-X 2.3rc2 onwards

Cooperative Rendezvous Protocols







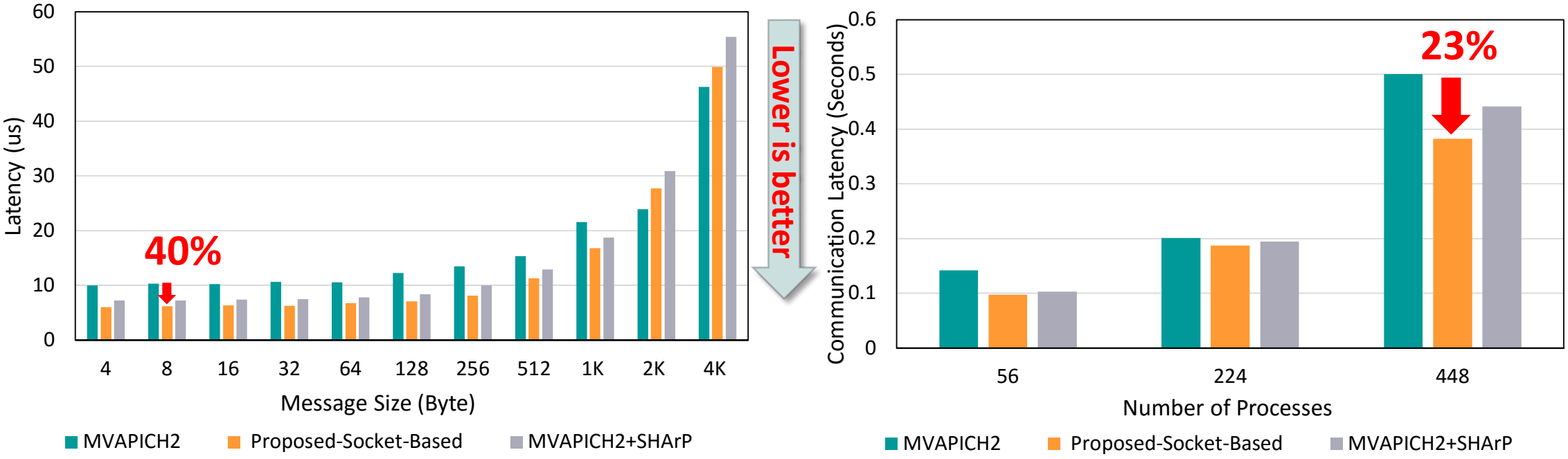
- Use both sender and receiver CPUs to progress communication concurrently
- Dynamically select rendezvous protocol based on communication primitives and sender/receiver availability (load balancing)
- Up to 2x improvement in large message latency and bandwidth
- Up to 19% improvement for Graph500 at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

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

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

Available in MVAPICH2-X 2.3rc2

Advanced Allreduce Collective Designs Using SHArP and Multi-Leaders



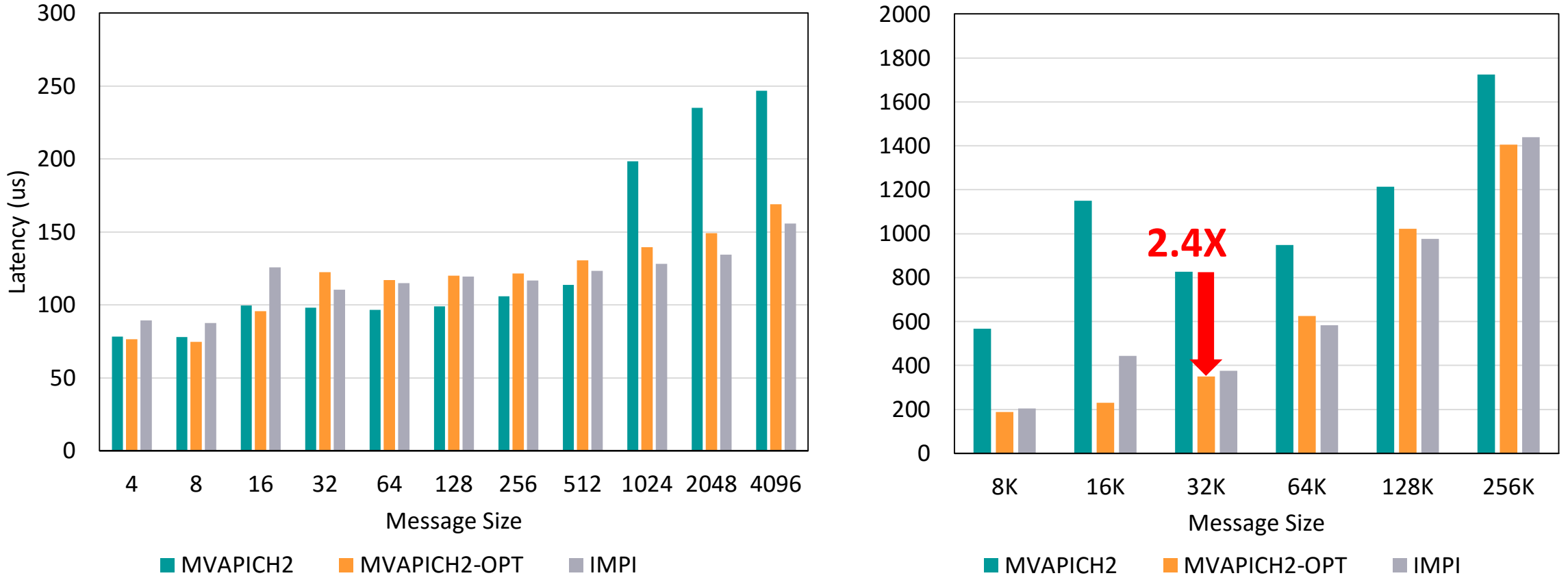
OSU Micro Benchmark (16 Nodes, 28 PPN)

HPCG (28 PPN)

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

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

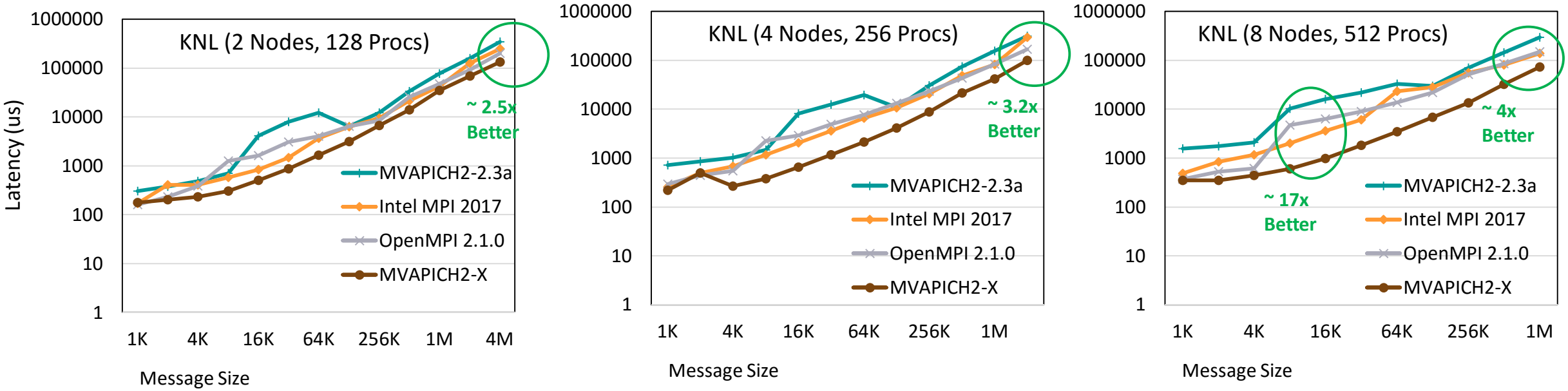
Performance of MPI_Allreduce On Stampede2 (10,240 Processes)



OSU Micro Benchmark 64 PPN

- MPI_Allreduce latency with 32K bytes reduced by 2.4X

Optimized CMA-based Collectives for Large Messages



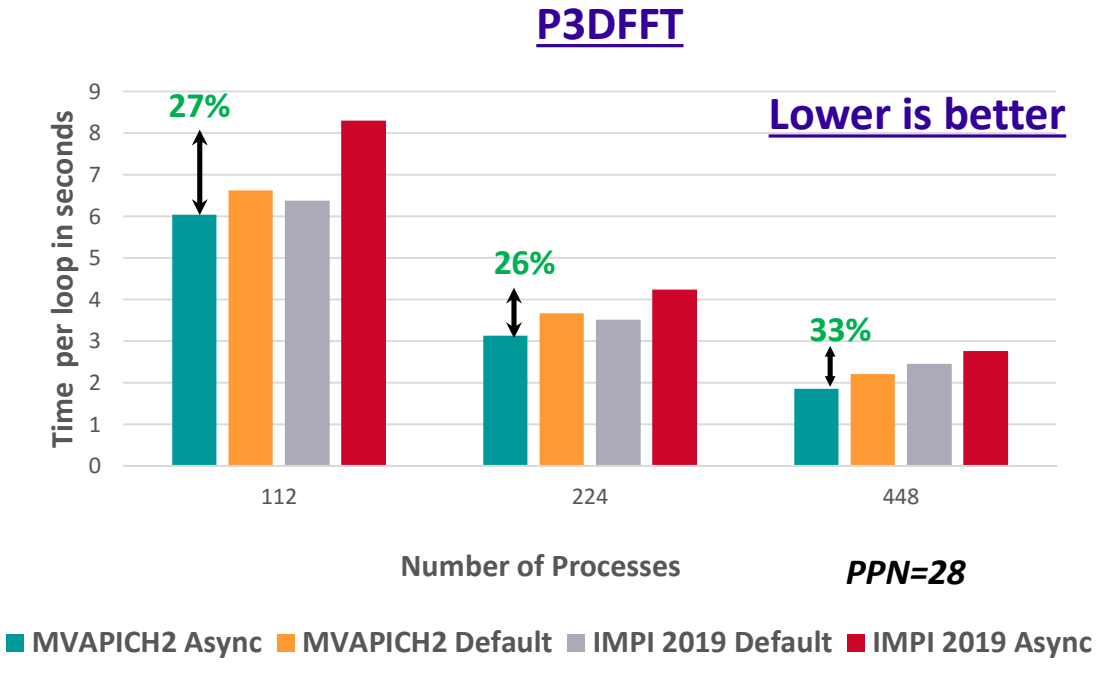
Performance of MPI_Gather on KNL nodes (64PPN)

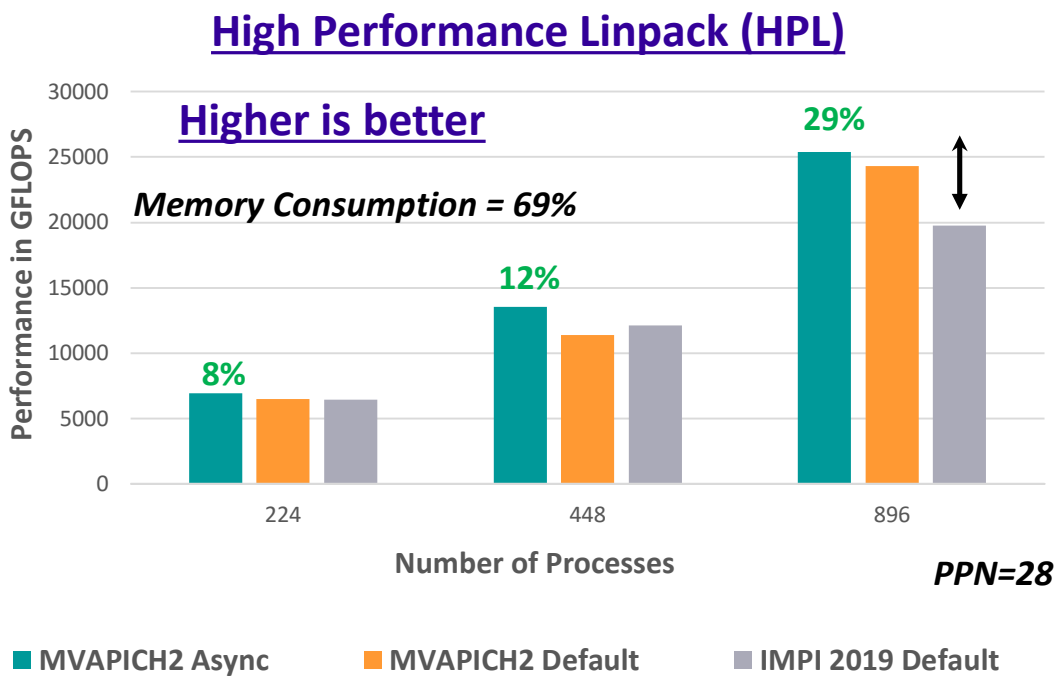
- Significant improvement over existing implementation for Scatter/Gather with 1MB messages (up to 4x on KNL, 2x on Broadwell, 14x on OpenPOWER)
- New two-level algorithms for better scalability
- Improved performance for other collectives (Bcast, Allgather, and Alltoall)

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

Available since MVAPICH2-X 2.3b

Benefits of the New Asynchronous Progress Design: Broadwell + InfiniBand





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

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

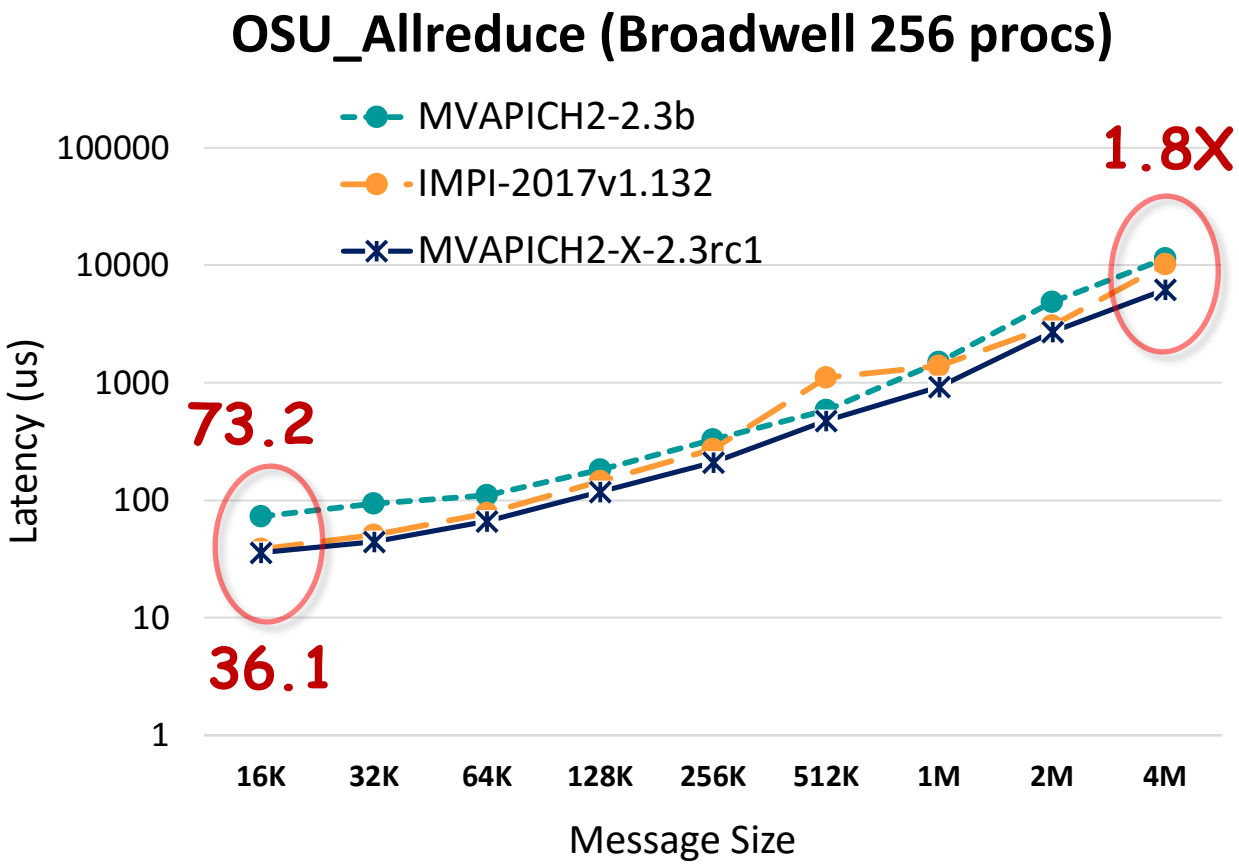
A. Ruhela, H. Subramoni, S. Chakraborty, M. Bayatpour, P. Kousha, and D.K. Panda, "Efficient design for MPI Asynchronous Progress without Dedicated Resources", Parallel Computing 2019

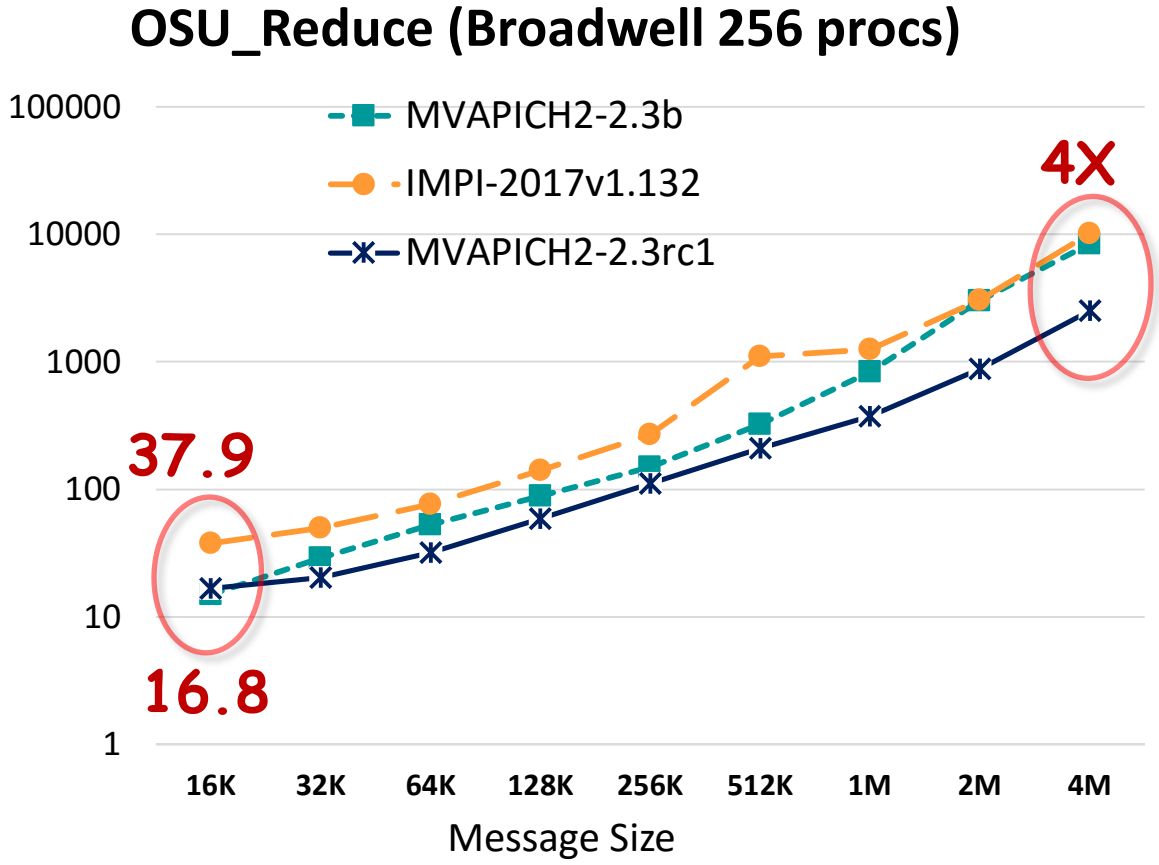
Available since MVAPICH2-X 2.3rc1

Overview of Some of the MVAPICH2-X Features

- Direct Connect (DC) Transport
- Co-operative Rendezvous Protocol
- Advanced All-reduce with SHARP
- CMA-based Collectives
- Asynchronous Progress
- XPMEM-based Reduction Collectives
- XPMEM-based Non-reduction Collectives
- Optimized Collective Communication and Advanced Transport Protocols
- PGAS and Hybrid MPI+PGAS Support

Shared Address Space (XPMEM)-based Collectives Design





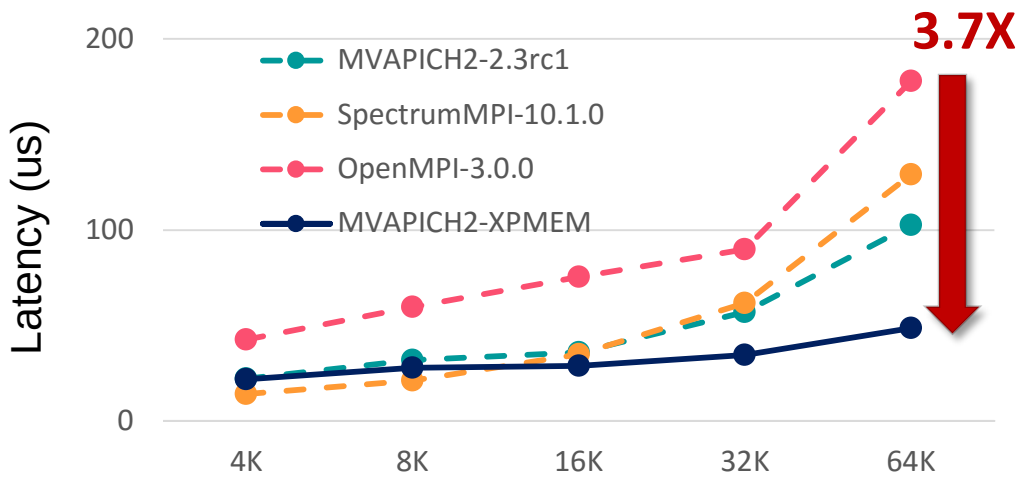
- "*Shared Address Space*"-based true *zero-copy* Reduction collective designs in MVAPICH2
- Offloaded computation/communication to peers ranks in reduction collective operation
- Up to **4X** improvement for 4MB Reduce and up to **1.8X** improvement for 4M AllReduce

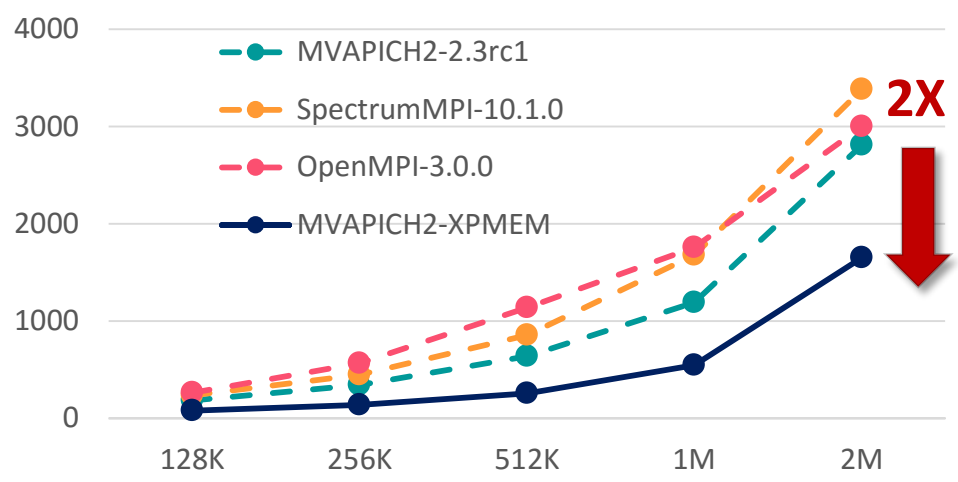
J. Hashmi, S. Chakraborty, M. Bayatpour, H. Subramoni, and D. Panda, Designing Efficient Shared Address Space Reduction Collectives for Multi-/Many-cores, International Parallel & Distributed Processing Symposium (IPDPS '18), May 2018.

Available since MVAPICH2-X 2.3rc1

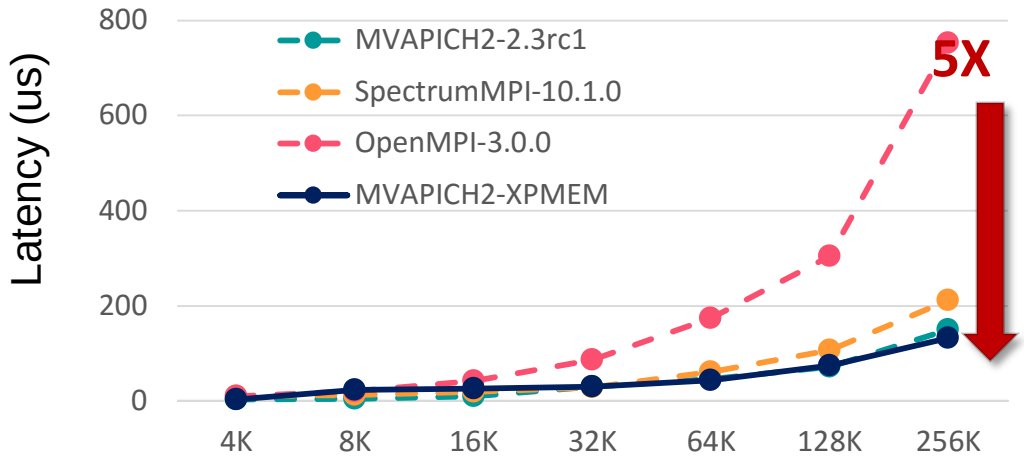
Reduction Collectives on IBM OpenPOWER

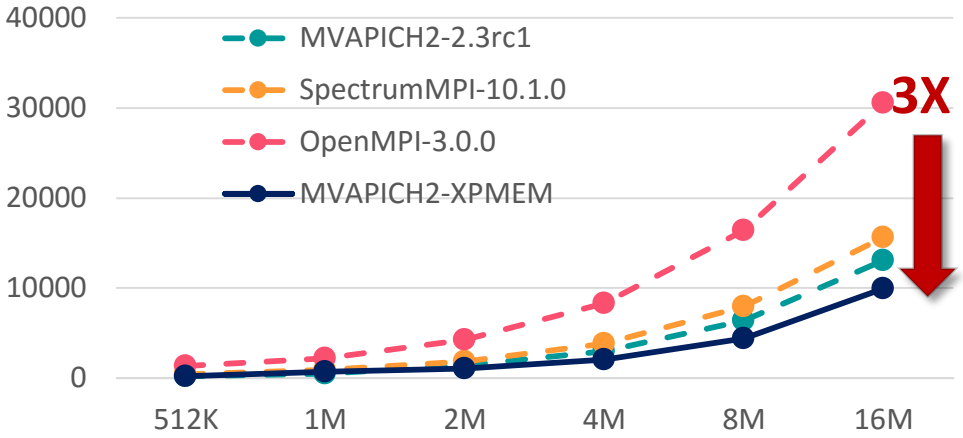
MPI_Allreduce





MPI_Reduce

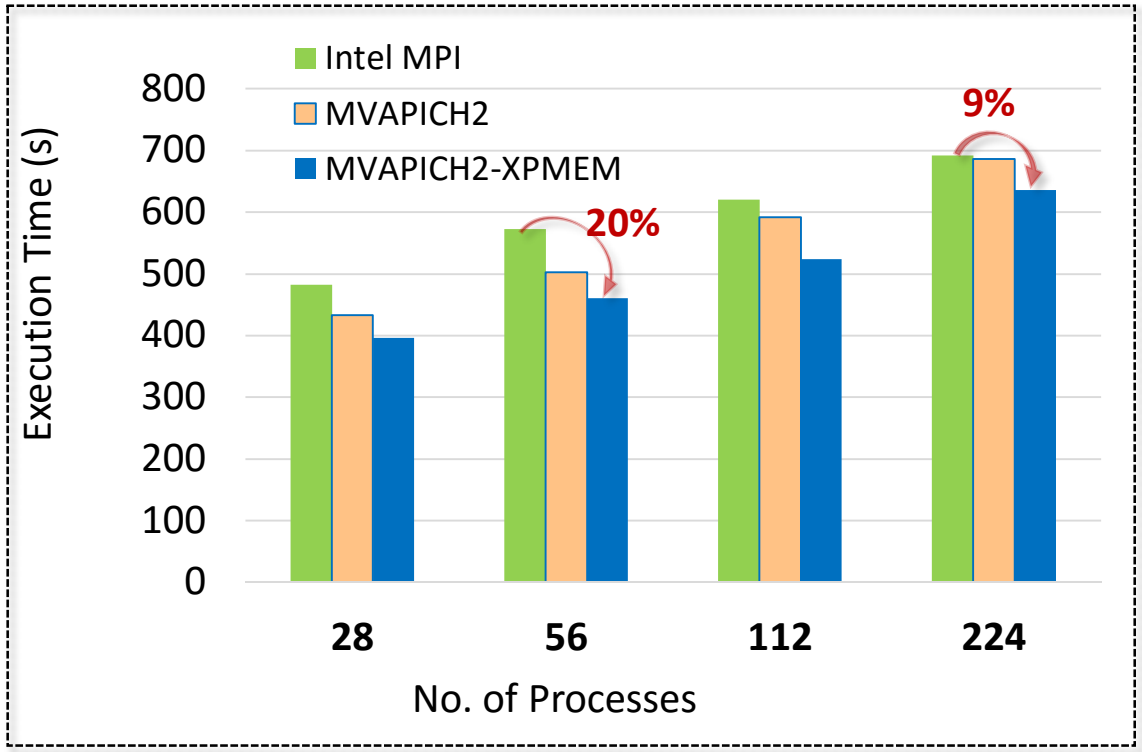




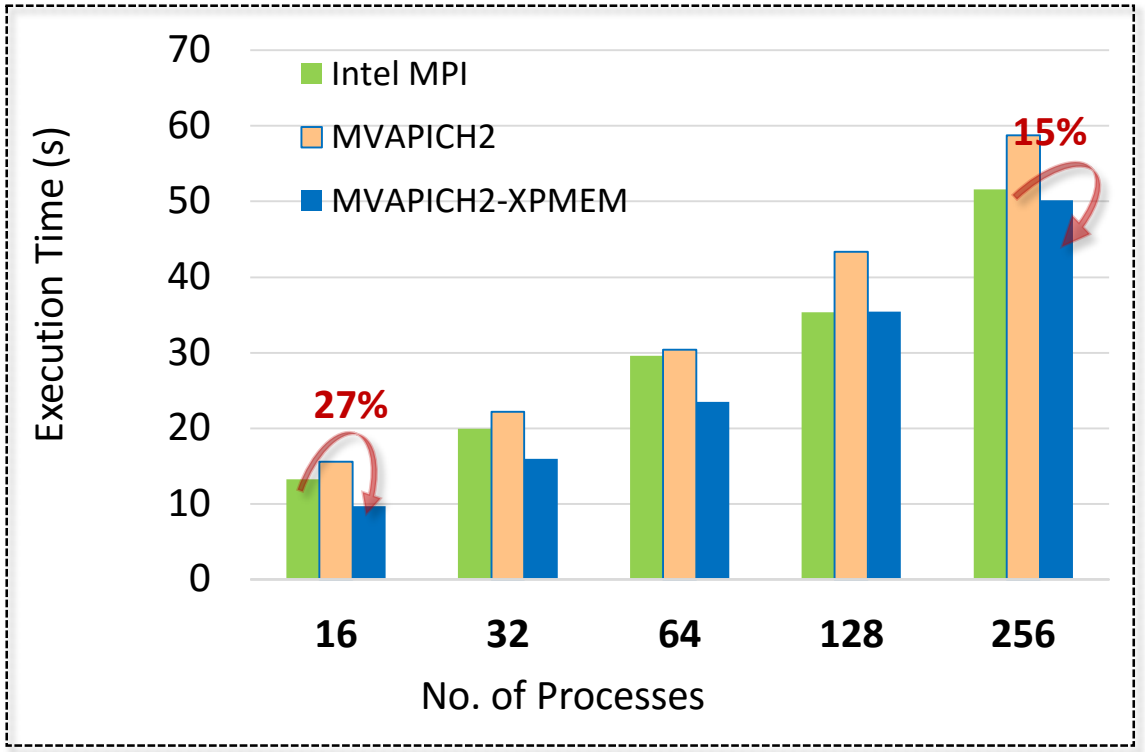
- Two POWER8 dual-socket nodes each with 20 ppn
- Up to **2X** improvement for Allreduce and **3X** improvement for Reduce at 4MB message
- Used osu_reduce and osu_allreduce from OSU Microbenchmarks v5.5

Application Level Benefits of XPMEM-based Designs

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

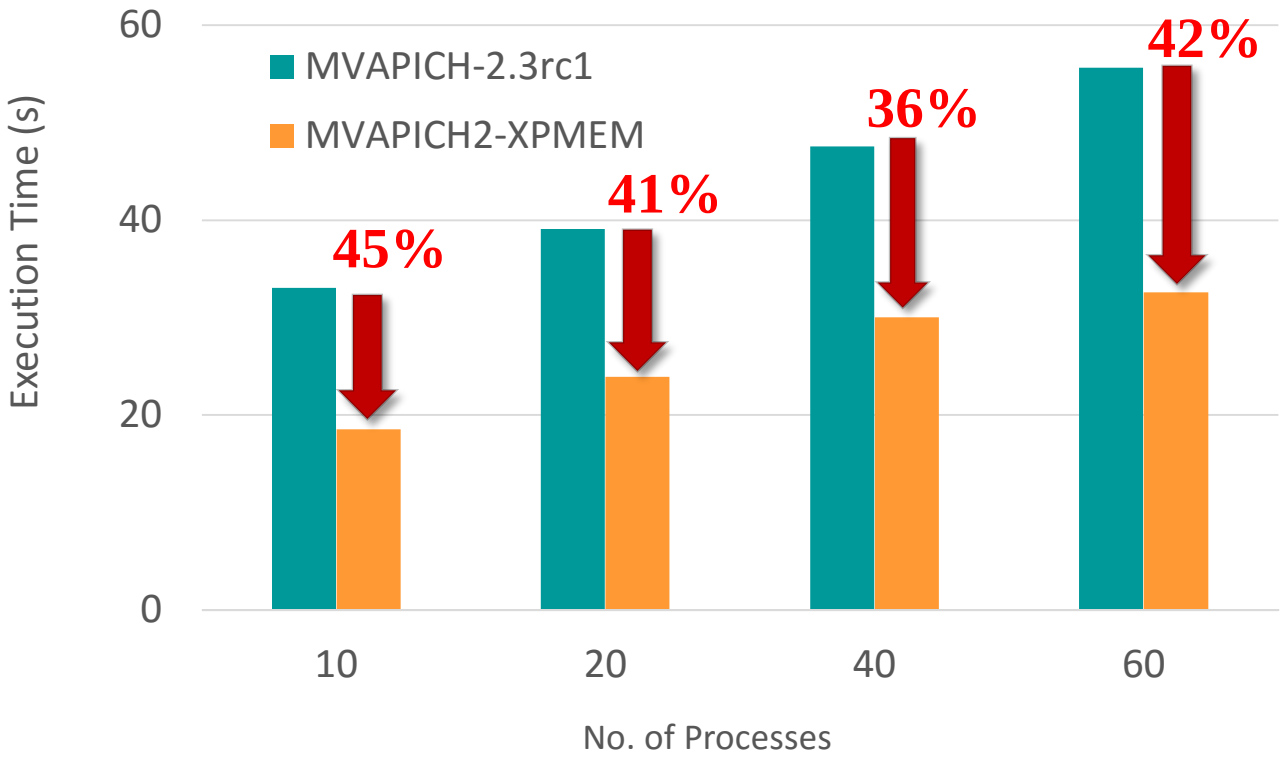


MiniAMR (dual-socket, ppn=16)



- Intel XeonCPU E5-2687W v3 @ 3.10GHz (10-core, 2-socket)
- Up to **20%** benefits over IMPI for CNTK DNN training using AllReduce
- Up to **27%** benefits over IMPI and up to **15%** improvement over MVAPICH2 for MiniAMR application kernel

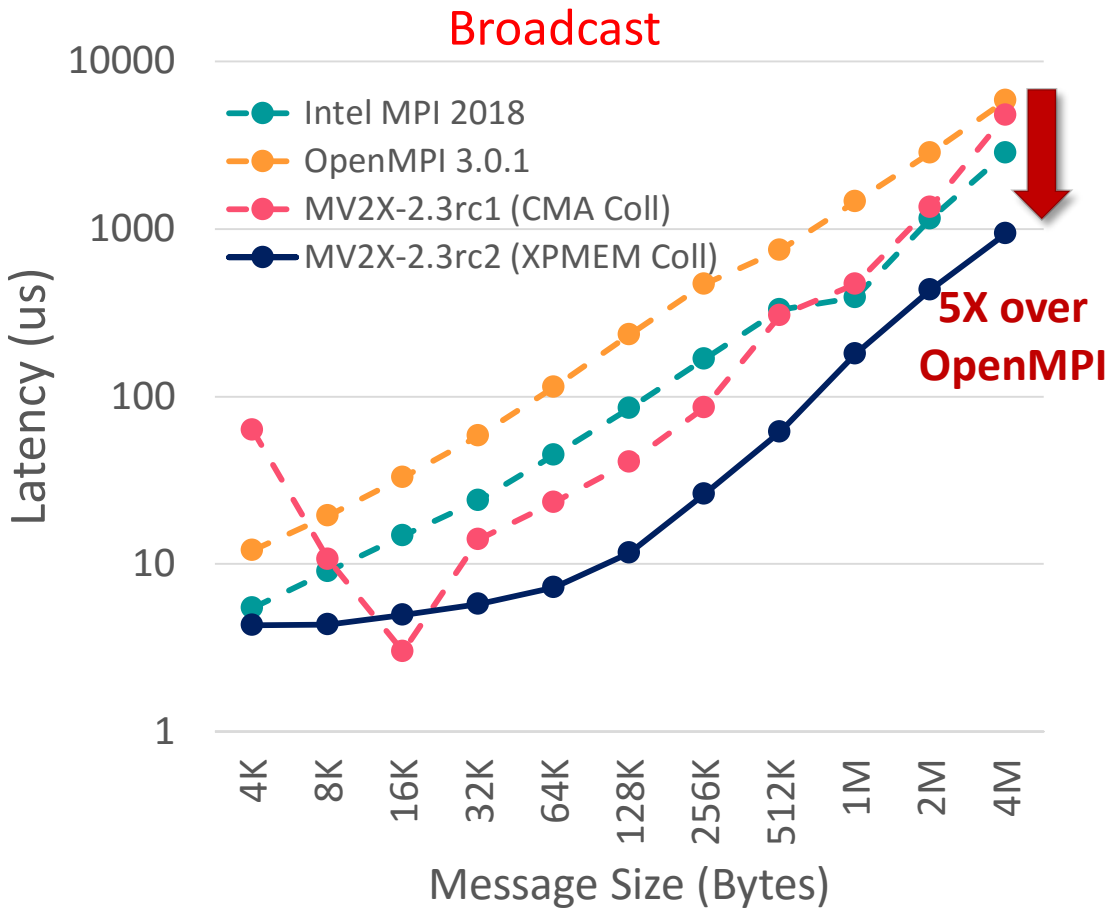
Impact of XPMEM-based Designs on MiniAMR

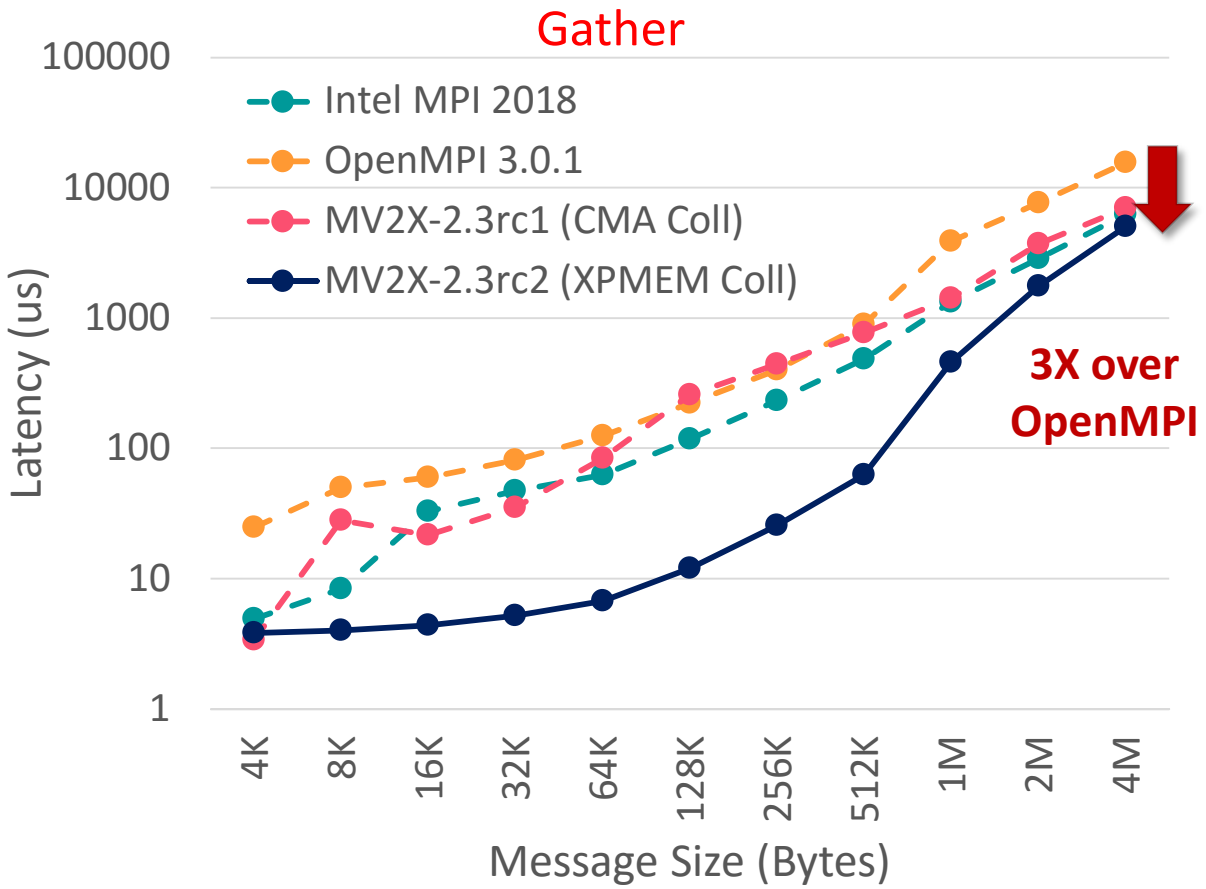


OpenPOWER (weak-scaling, 3 nodes, ppn=20)

- Two POWER8 dual-socket nodes each with 20 ppn
- MiniAMR application execution time comparing MVAPICH2-2.3rc1 and optimized All-Reduce design
  - MiniAMR application for weak-scaling workload on up to three POWER8 nodes.
  - Up to 45% improvement over MVAPICH2-2.3rc1 in mesh-refinement time

Performance of Non-Reduction Collectives with XPMEM

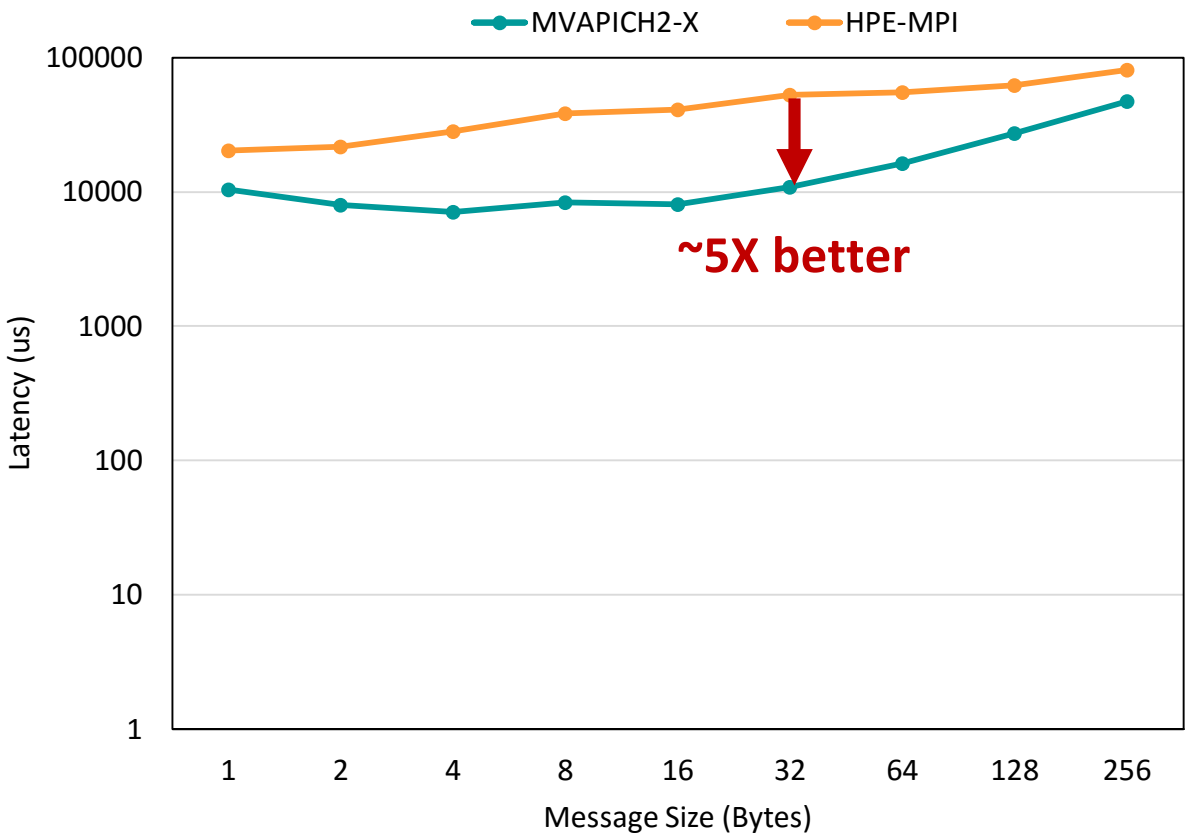




- **28 MPI Processes** on single dual-socket Broadwell E5-2680v4, 2x14 core processor
- Used osu_bcast from OSU Microbenchmarks v5.5

Impact of Optimized Small Message MPI_Alltoallv Algorithm

- **Optimized designs in MVAPICH2-X offer significantly improved performance for small message MPI_Alltoallv**

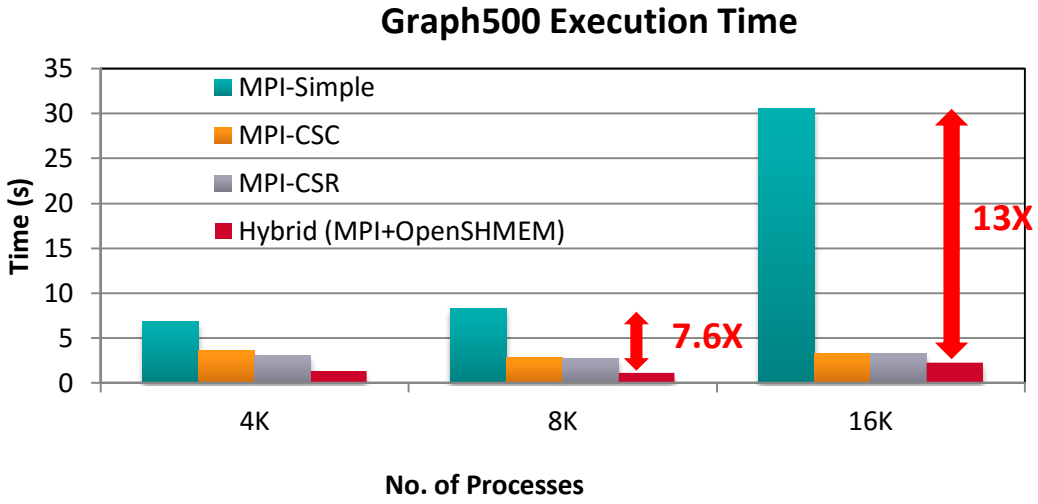


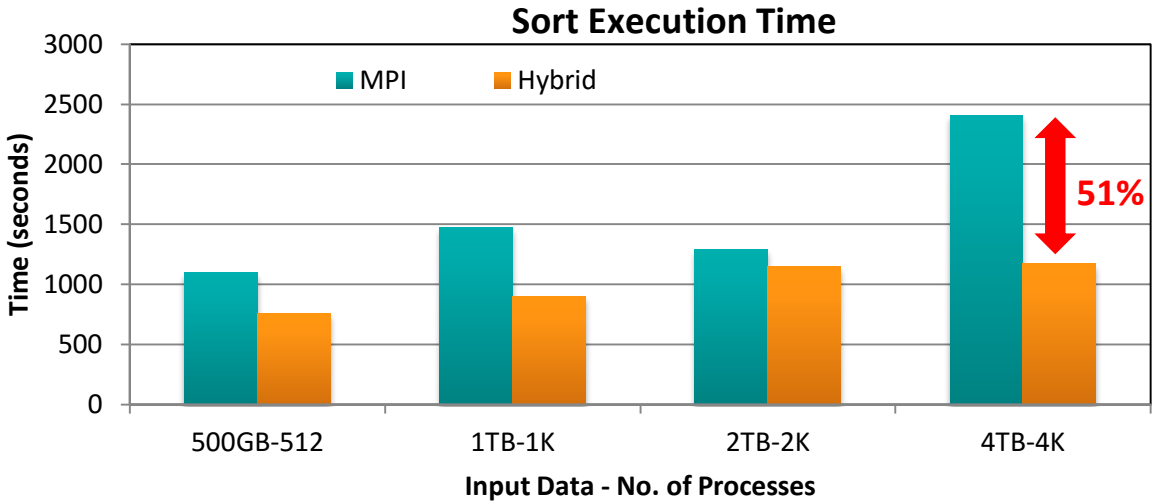
Courtesy: Pramod Shivaji Kumbhar@EPFL

- Up to 5X benefits over HPE-MPI using optimized using optimized Alltoallv algorithm and Direct Connected transport protocol
- Numbers taken on bbpv2.epfl.ch
  - 96 KNL nodes with 64 ppn (6,144 processes)
  - osu_alltoallv from OSU Micro Benchmarks
- Environment variables used
  - MV2_USE_DC=1
  - MV2_NUM_DC_TGT=64
  - MV2_SMALL_MSG_DC_POOL=96
  - MV2_LARGE_MSG_DC_POOL=96
  - MV2_USE_RDMA_CM=0

Available from MVAPICH2-X 2.3rc2 onwards

Application Level Performance with Graph500 and Sort





- Performance of Hybrid (MPI+ OpenSHMEM) Graph500 Design
  - 8,192 processes
    - **2.4X** improvement over MPI-CSR
    - **7.6X** improvement over MPI-Simple
  - 16,384 processes
    - **1.5X** improvement over MPI-CSR
    - **13X** improvement over MPI-Simple

- Performance of Hybrid (MPI+OpenSHMEM) Sort Application
  - 4,096 processes, 4 TB Input Size
    - MPI – 2408 sec; 0.16 TB/min
    - Hybrid – 1172 sec; 0.36 TB/min
    - **51%** improvement over MPI-design

J. Jose, K. Kandalla, S. Potluri, J. Zhang and D. K. Panda, Optimizing Collective Communication in OpenSHMEM, Int'l Conference on Partitioned Global Address Space Programming Models (PGAS '13), October 2013.

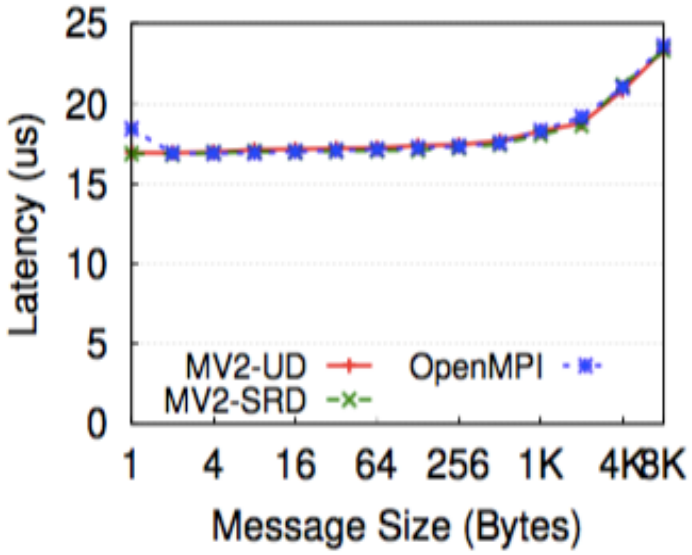
J. Jose, S. Potluri, K. Tomko and D. K. Panda, Designing Scalable Graph500 Benchmark with Hybrid MPI+OpenSHMEM Programming Models, International Supercomputing Conference (ISC'13), June 2013

J. Jose, K. Kandalla, M. Luo and D. K. Panda, Supporting Hybrid MPI and OpenSHMEM over InfiniBand: Design and Performance Evaluation, Int'l Conference on Parallel Processing (ICPP '12), September 2012

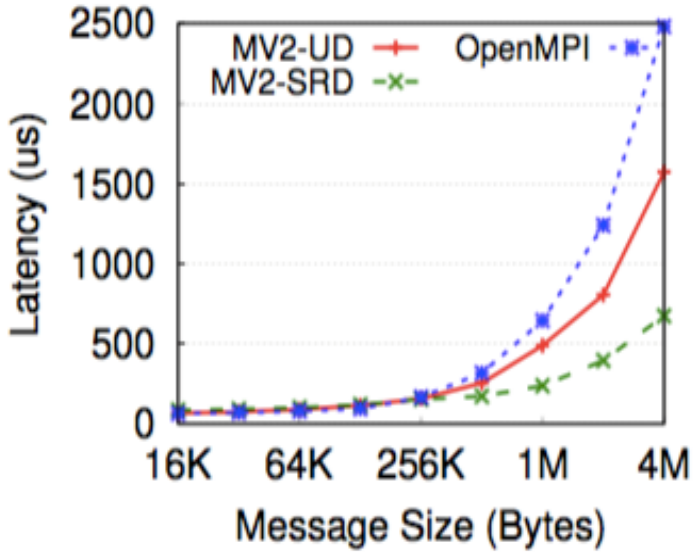
MVAPICH2-X-AWS 2.3

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

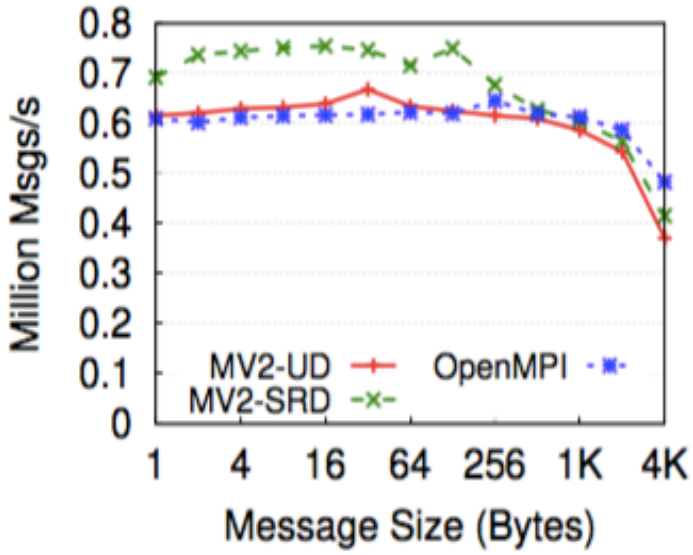
Point-to-Point Performance



(a) Small Message Latency



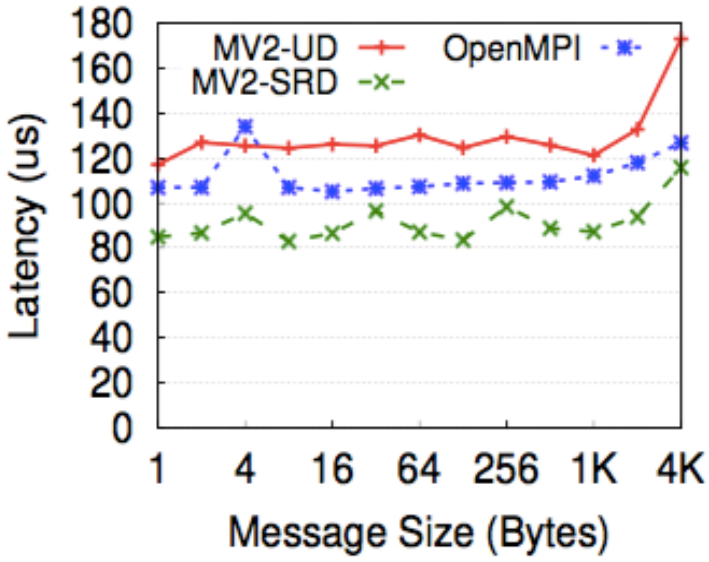
(b) Large Message Latency



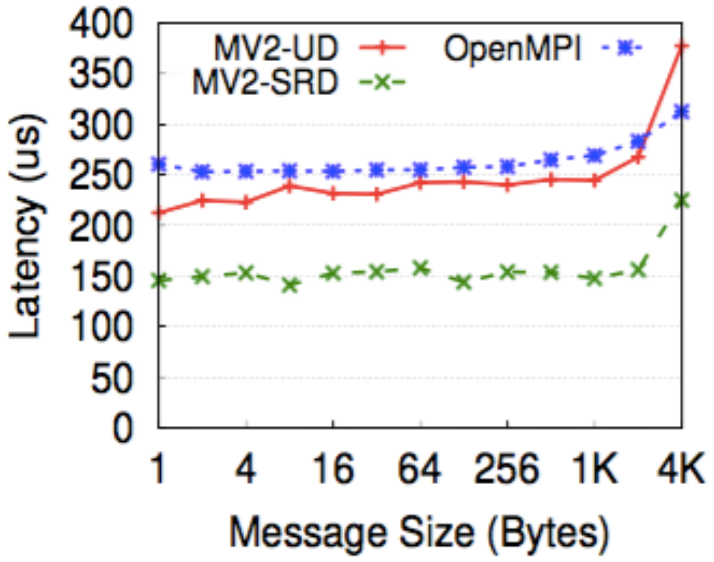
(c) Unidirectional Message Rate

- Both UD and SRD shows similar latency for small messages
- SRD shows higher message rate due to lack of software reliability overhead
- SRD is faster for large messages due to larger MTU size

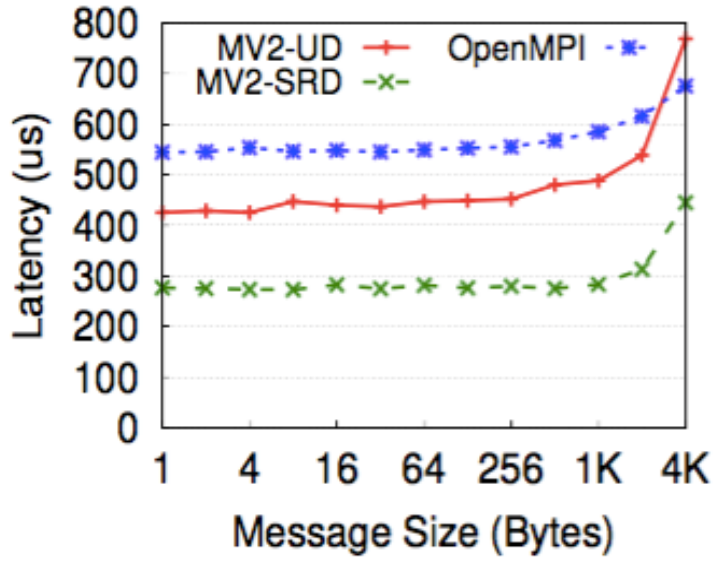
Collective Performance: MPI Gatherv



(a) 2 Nodes, 72 Processes



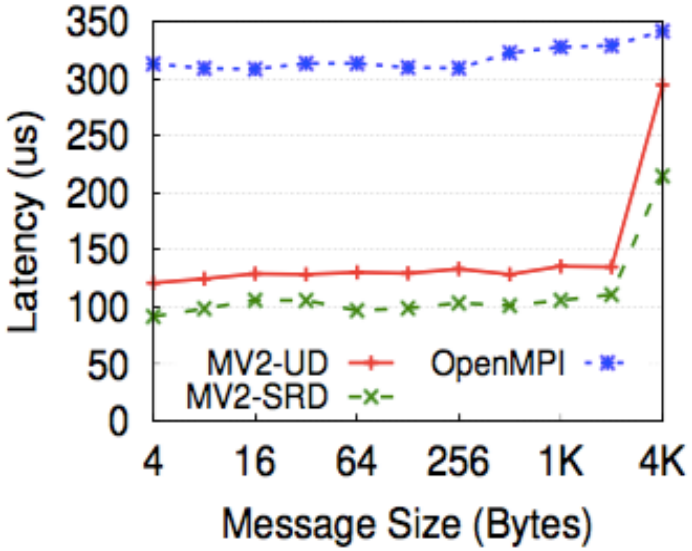
(b) 4 Nodes, 144 Processes



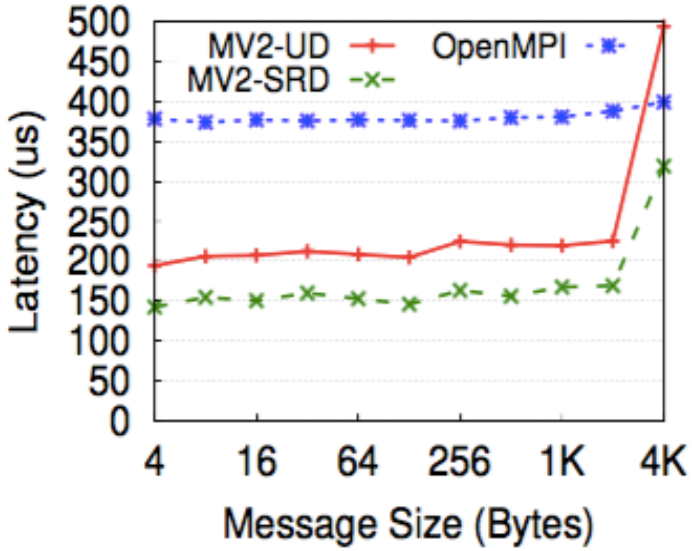
(c) 8 Nodes, 288 Processes

- Up to 33% improvement with SRD compared to UD
- Root does not need to send explicit acks to non-root processes
- Non-roots can exit as soon as the message is sent (no need to wait for acks)

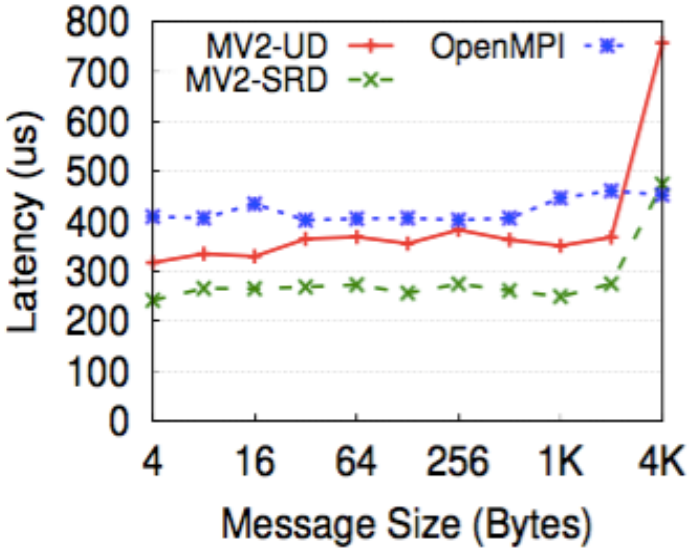
Collective Performance: MPI Allreduce



(a) 2 Nodes, 72 Processes



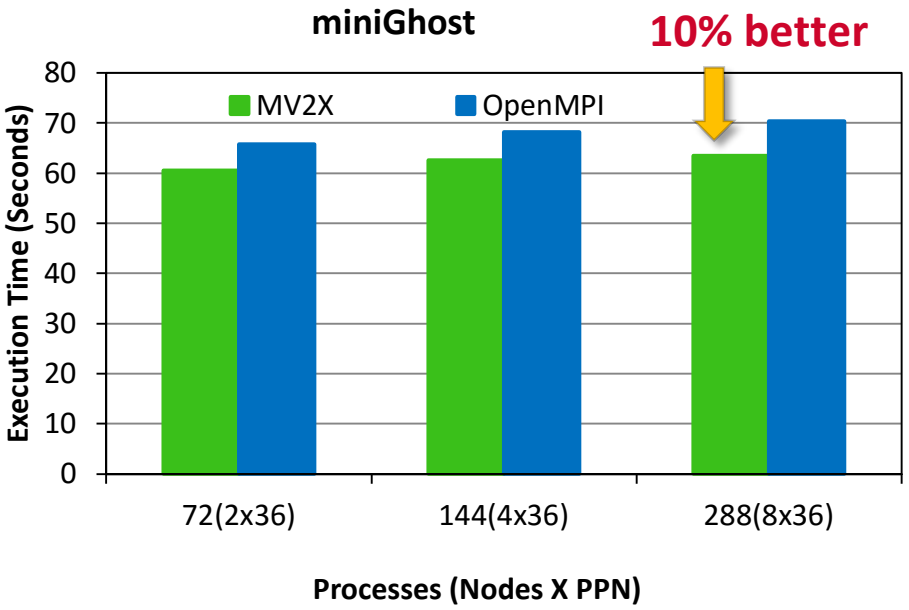
(b) 4 Nodes, 144 Processes

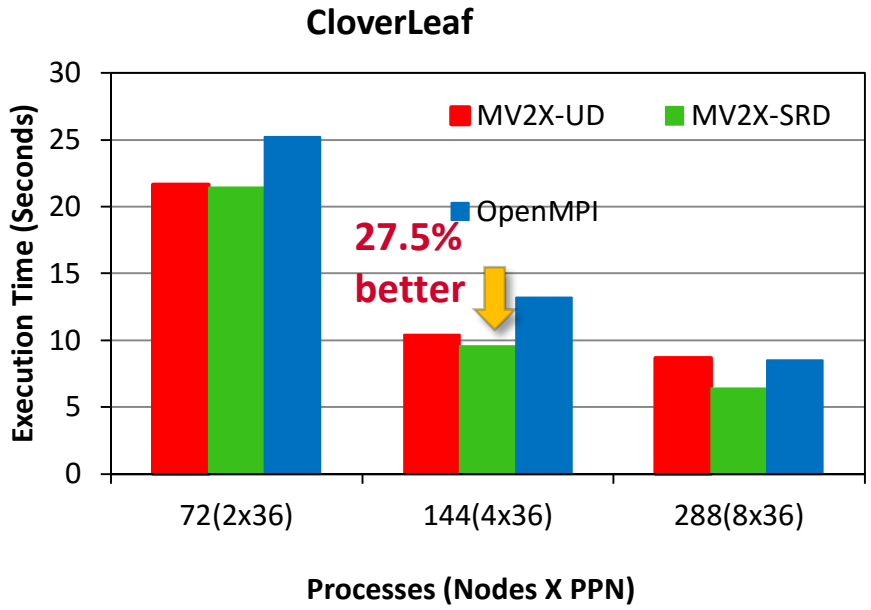


(c) 8 Nodes, 288 Processes

- Up to 18% improvement with SRD compared to UD
- Bidirectional communication pattern allows piggybacking of acks
- Modest improvement compared to asymmetric communication patterns

Application Performance





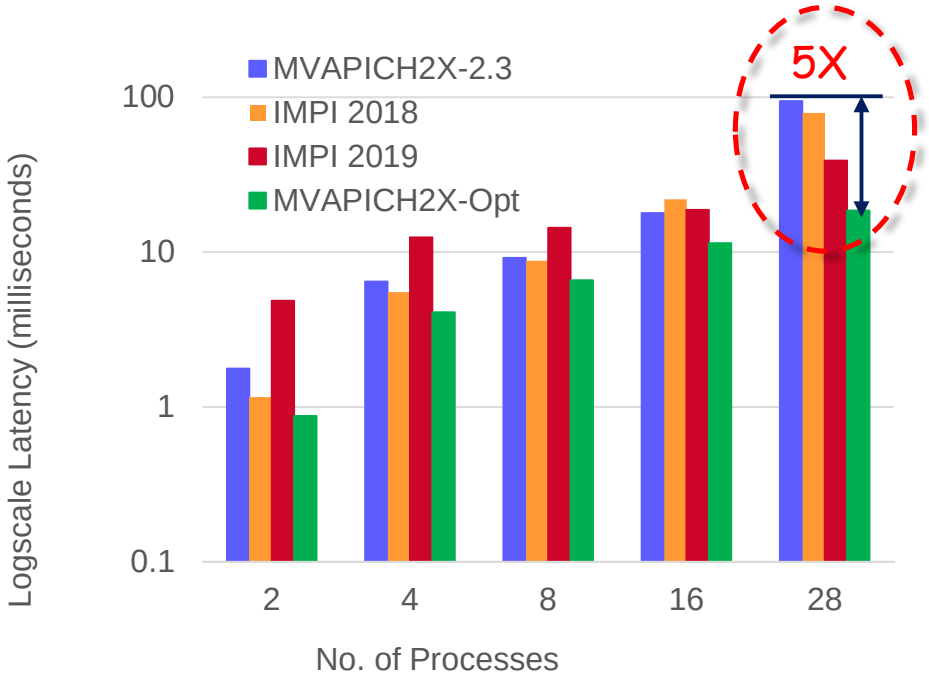
- Up to 10% performance improvement for MiniGhost on 8 nodes
- Up to 27% better performance with CloverLeaf on 8 nodes

S. Chakraborty, S. Xu, H. Subramoni and D. K. Panda, Designing Scalable and High-Performance MPI Libraries on Amazon Elastic Adapter, Hot Interconnect, 2019

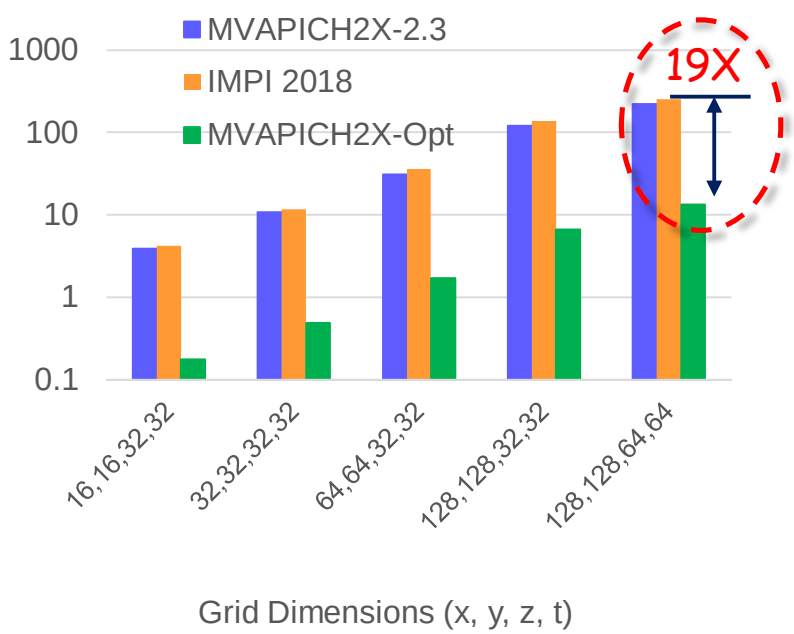
MVAPICH2-X Upcoming Features

- XPMEM-based MPI Derived Datatype Designs
- Exploiting Hardware Tag Matching

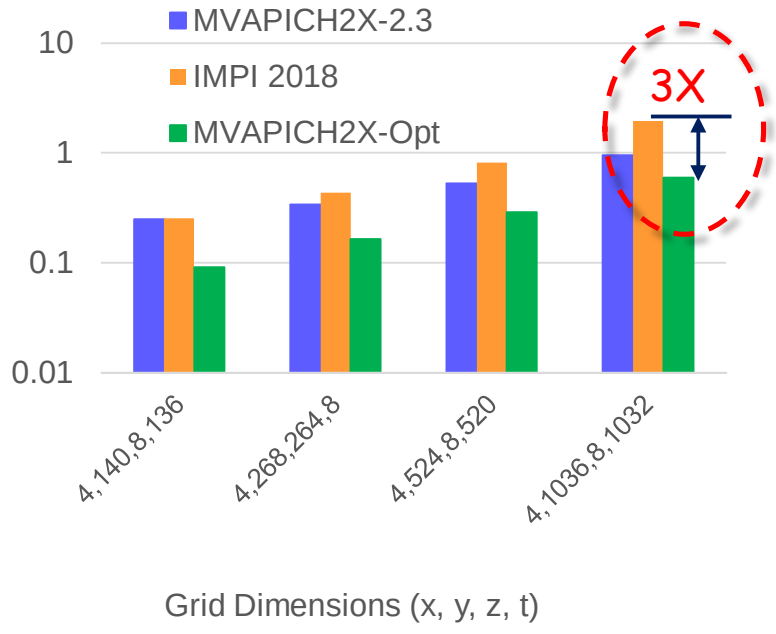
Efficient Zero-copy MPI Datatypes for Emerging Architectures



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



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



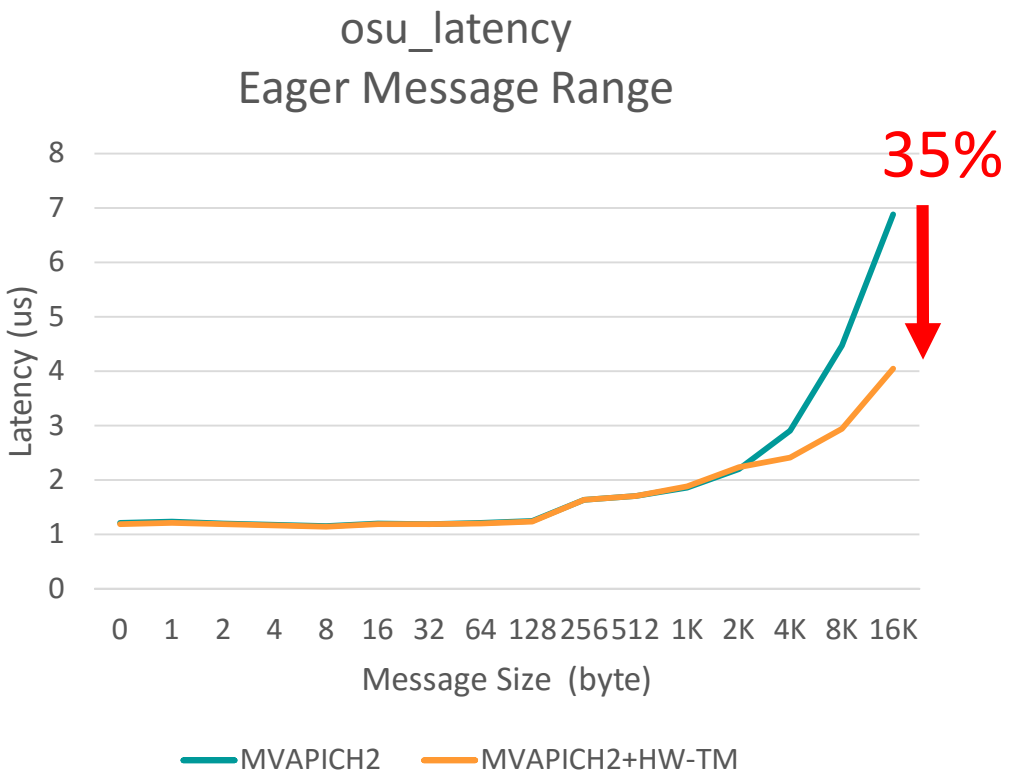
NAS-MG Datatype Kernel on **OpenPOWER** (20-core)

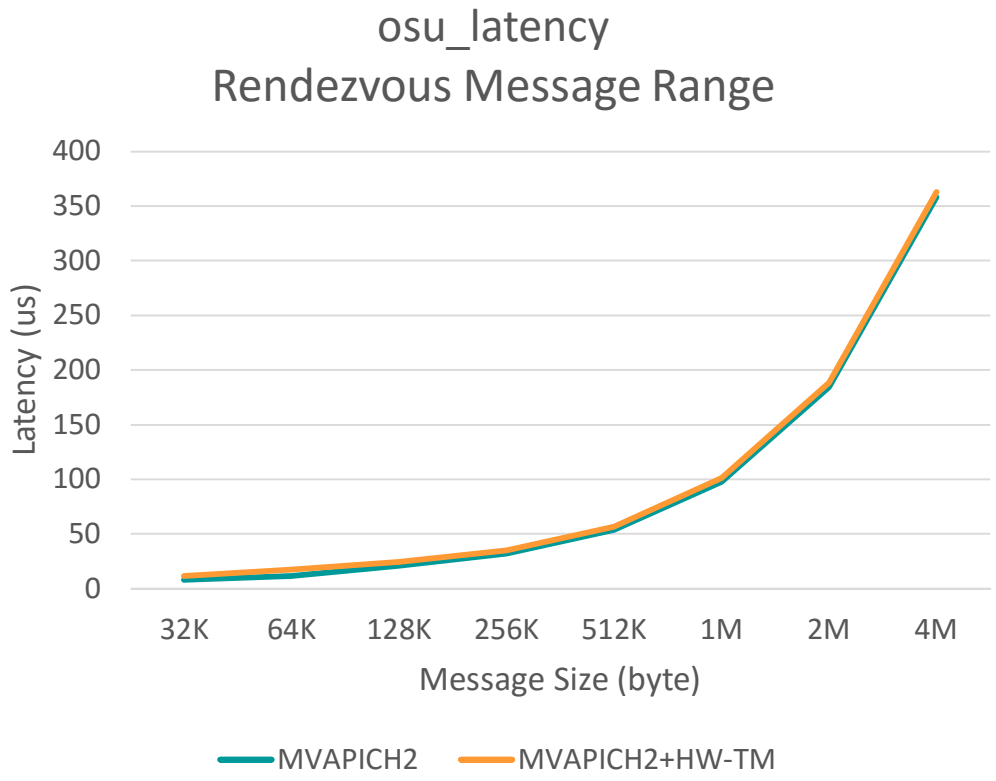
- New designs for efficient zero-copy based MPI derived datatype processing
- Efficient schemes mitigate datatype translation, packing, and exchange overheads
- Demonstrated benefits over prevalent MPI libraries for various application kernels
- To be available in the upcoming MVAPICH2-X release

Hardware Tag Matching Support

- Offloads the processing of point-to-point MPI messages from the host processor to HCA
- Enables zero copy of MPI message transfers
  - Messages are written directly to the user's buffer without extra buffering and copies
- Provides rendezvous progress offload to HCA
  - Increases the overlap of communication and computation

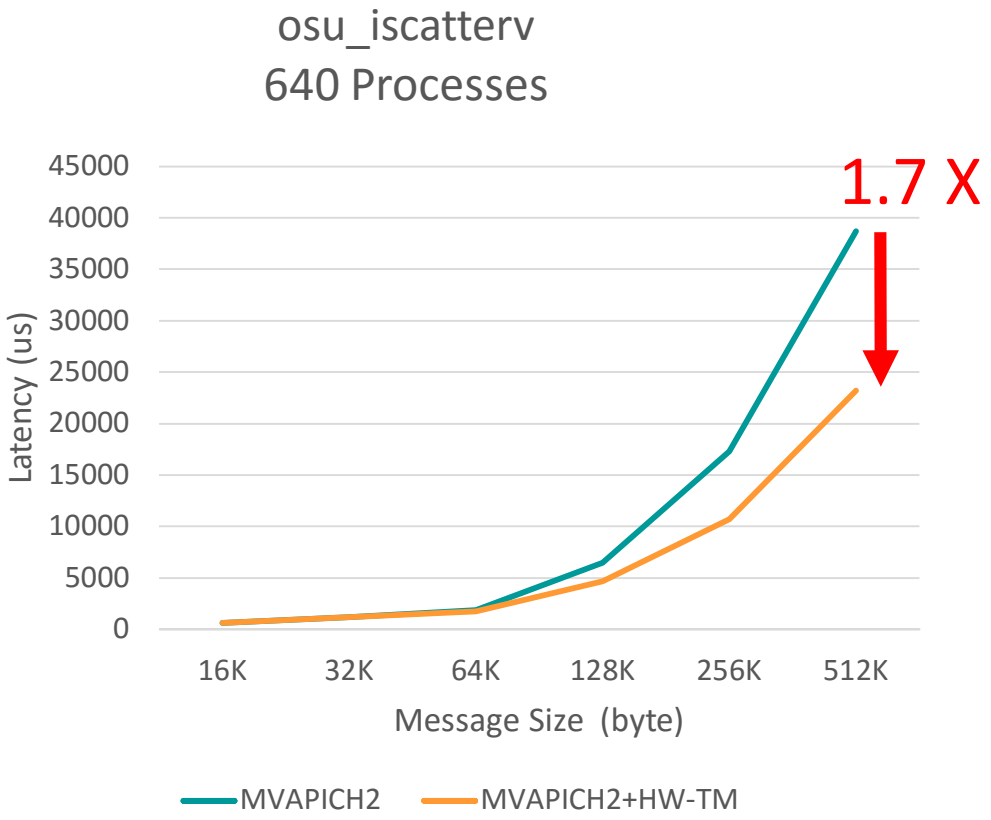
Impact of Zero Copy MPI Message Passing using HW Tag Matching

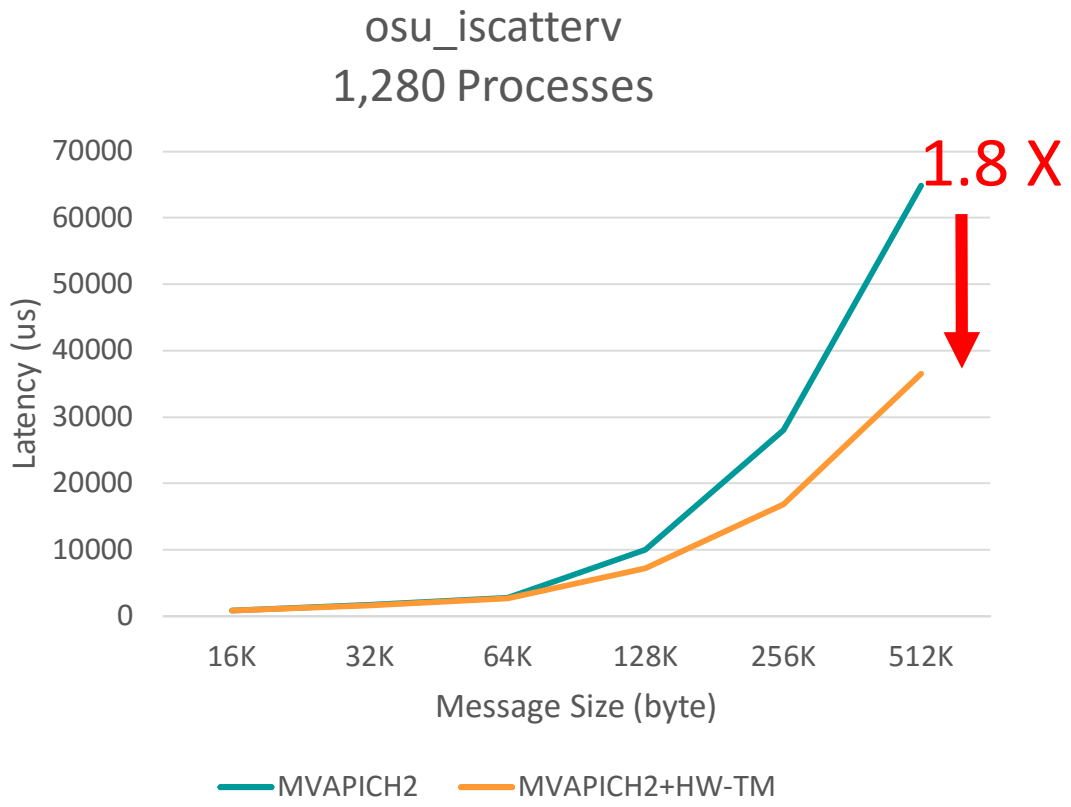




Removal of intermediate buffering/copies can lead up to 35% performance improvement in latency of medium messages

Impact of Rendezvous Offload using HW Tag Matching





The increased overlap can lead to 1.8X performance improvement in total latency of osu_iscatterv

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

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

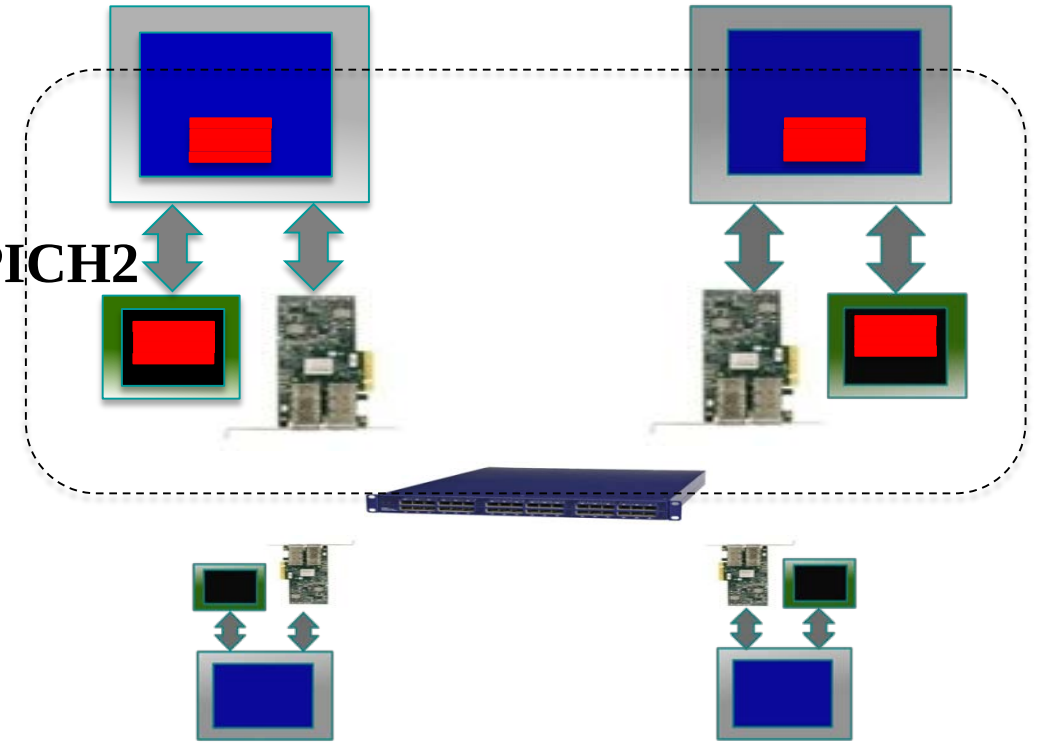
- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

At Sender:

```
MPI_Send(s_devbuf, size, …);
```

At Receiver:

```
MPI_Recv(r_devbuf, size, …);
```



High Performance and High Productivity

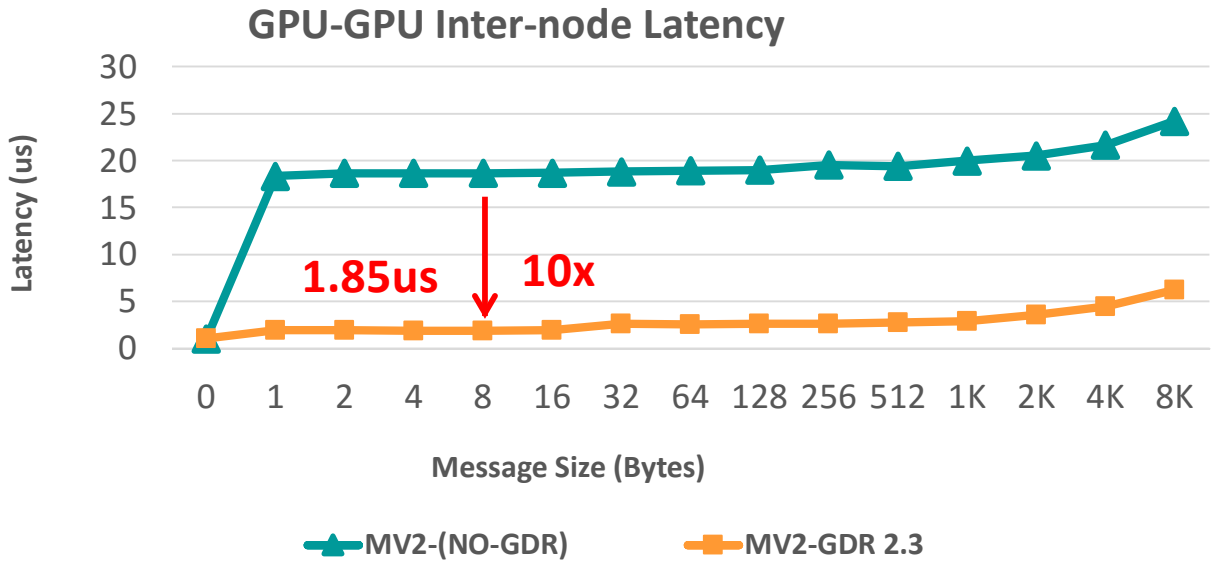
CUDA-Aware MPI: MVAPICH2-GDR 1.8-2.3.2 Releases

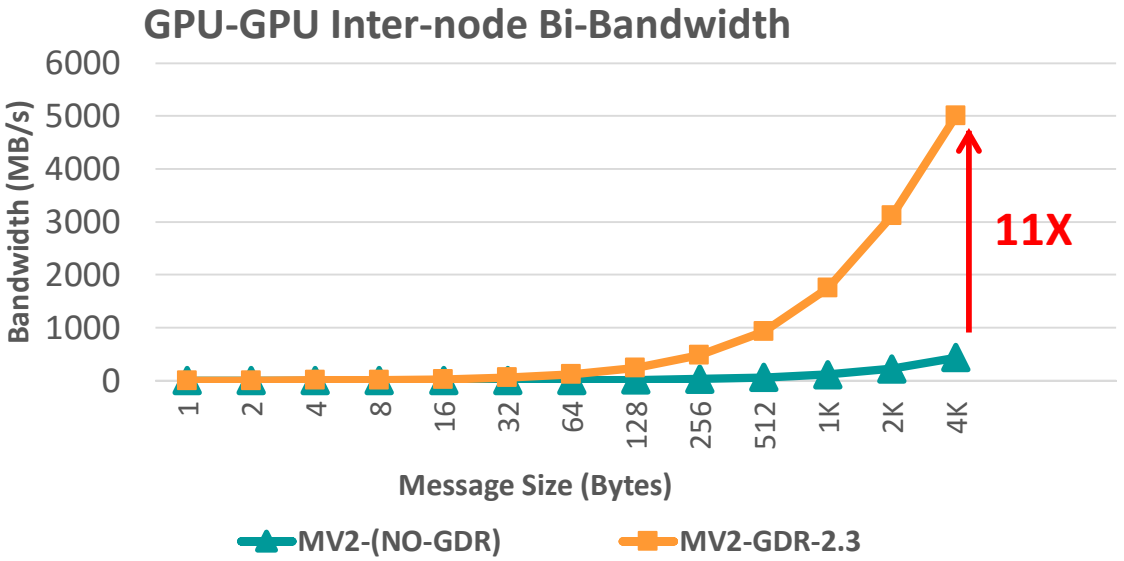
- Support for MPI communication from NVIDIA GPU device memory
- High performance RDMA-based inter-node point-to-point communication (GPU-GPU, GPU-Host and Host-GPU)
- High performance intra-node point-to-point communication for multi-GPU adapters/node (GPU-GPU, GPU-Host and Host-GPU)
- Taking advantage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

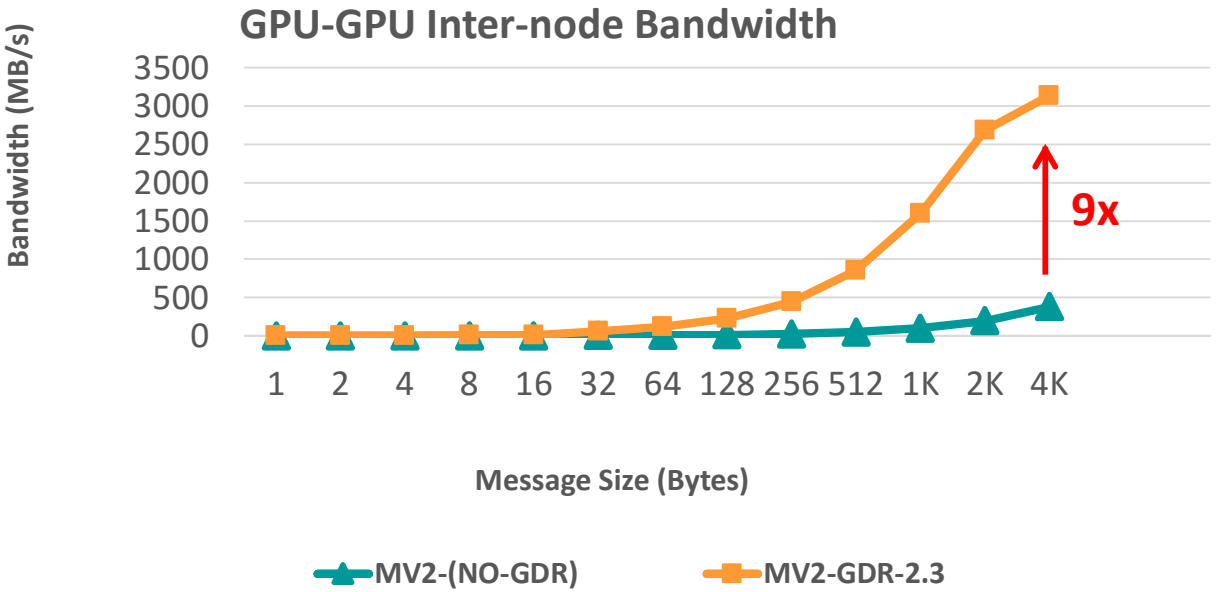
MVAPICH2-GDR 2.3.2

- Released on 08/08/2019
- Major Features and Enhancements
  - Based on MVAPICH2 2.3.1
  - Support for CUDA 10.1
  - Support for PGI 19.x
  - Enhanced intra-node and inter-node point-to-point performance
  - Enhanced MPI_Allreduce performance for DGX-2 system
  - Enhanced GPU communication support in MPI_THREAD_MULTIPLE mode
  - Enhanced performance of datatype support for GPU-resident data
    - Zero-copy transfer when P2P access is available between GPUs through NVLink/PCIe
  - Enhanced GPU-based point-to-point and collective tuning
    - OpenPOWER systems such as ORNL Summit and LLNL Sierra ABCI system @AIST, Owens and Pitzer systems @Ohio Supercomputer Center
  - Scaled Allreduce to 24,576 Volta GPUs on Summit
  - Enhanced intra-node and inter-node point-to-point performance for DGX-2 and IBM POWER8 and IBM POWER9 systems
  - Enhanced Allreduce performance for DGX-2 and IBM POWER8/POWER9 systems
  - Enhanced small message performance for CUDA-Aware MPI_Put and MPI_Get
  - Flexible support for running TensorFlow (Horovod) jobs

Optimized MVAPICH2-GDR Design

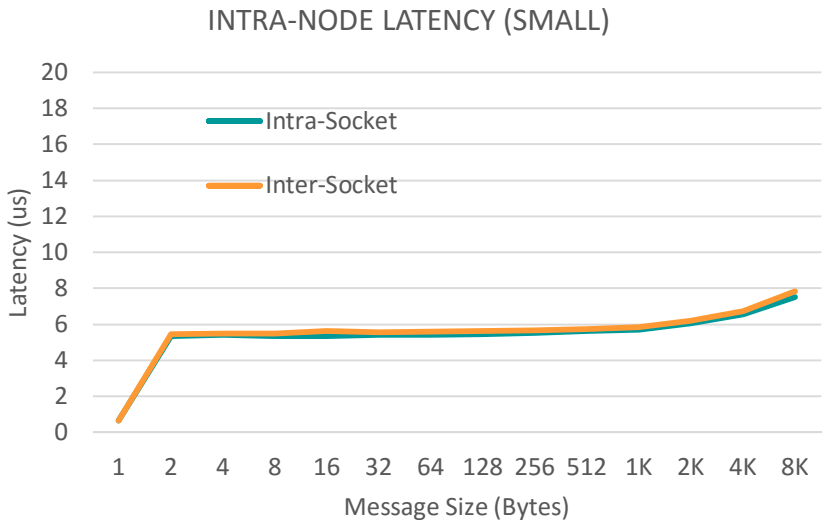


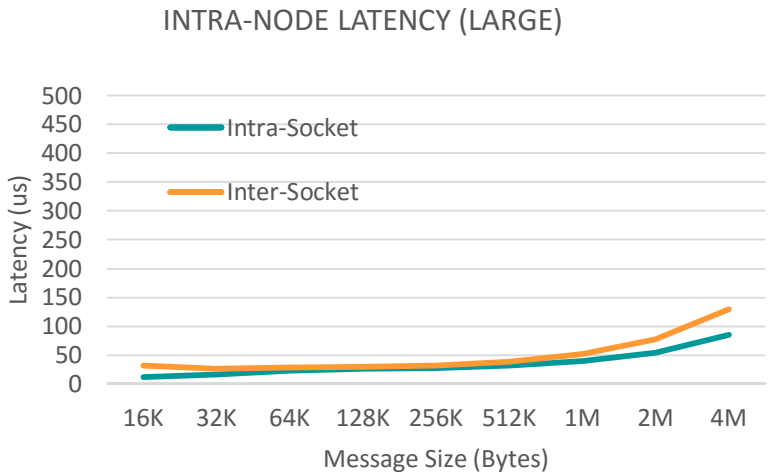


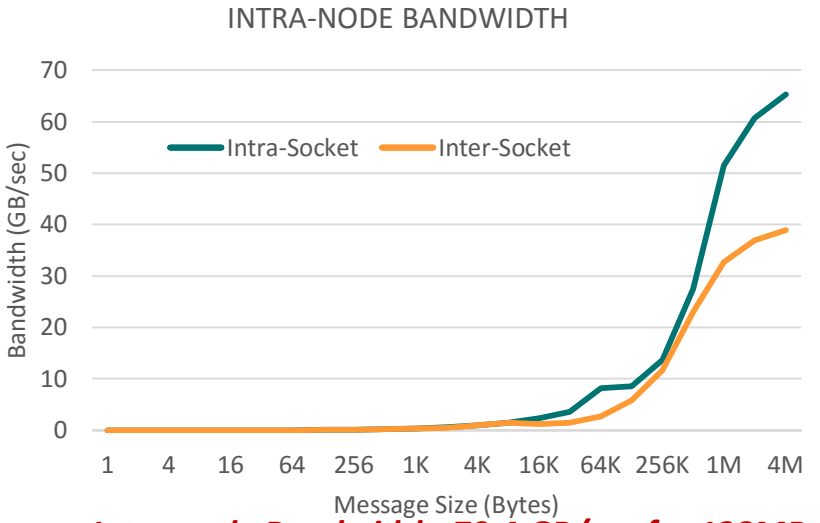


MVAPICH2-GDR-2.3
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

Device-to-Device Performance on OpenPOWER (NVLink2 + Volta)

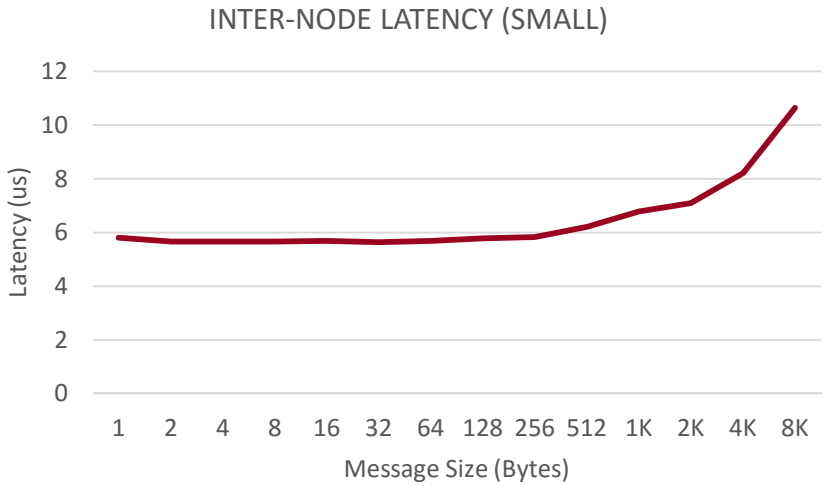


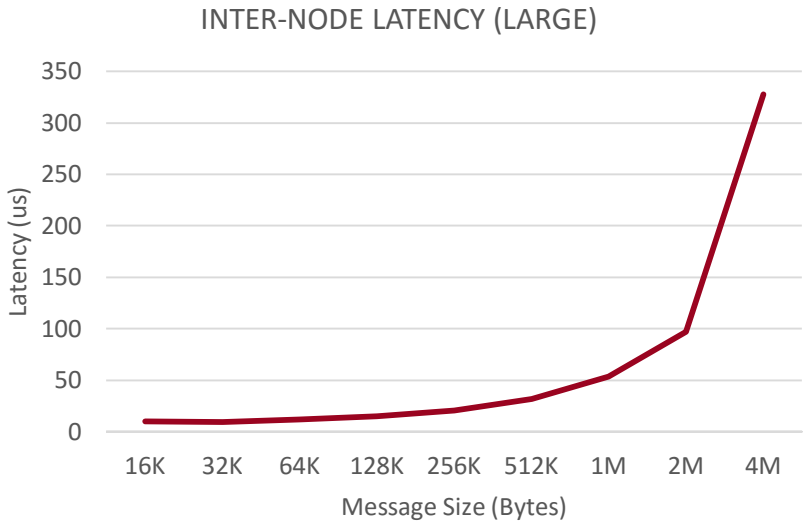


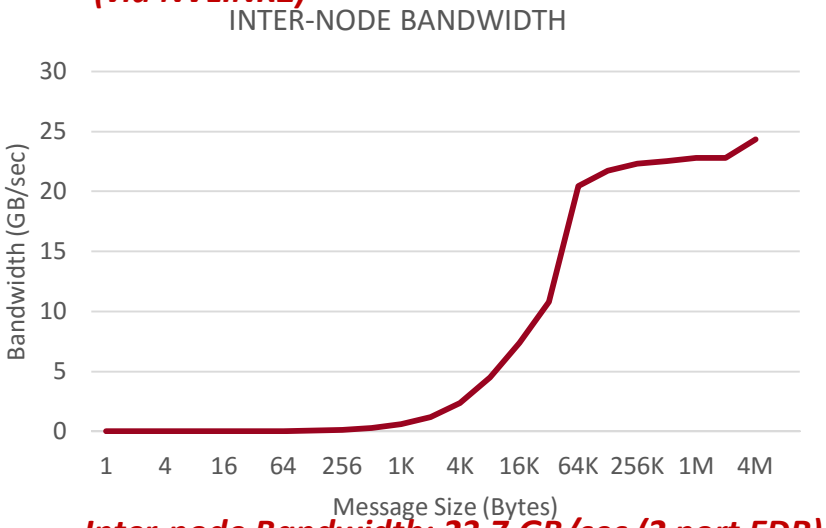


Intra-node Latency: 5.36 us (without GDRCopy)

Intra-node Bandwidth: 70.4 GB/sec for 128MB (via NVLINK2)







Inter-node Latency: 5.66 us (without GDRCopy)

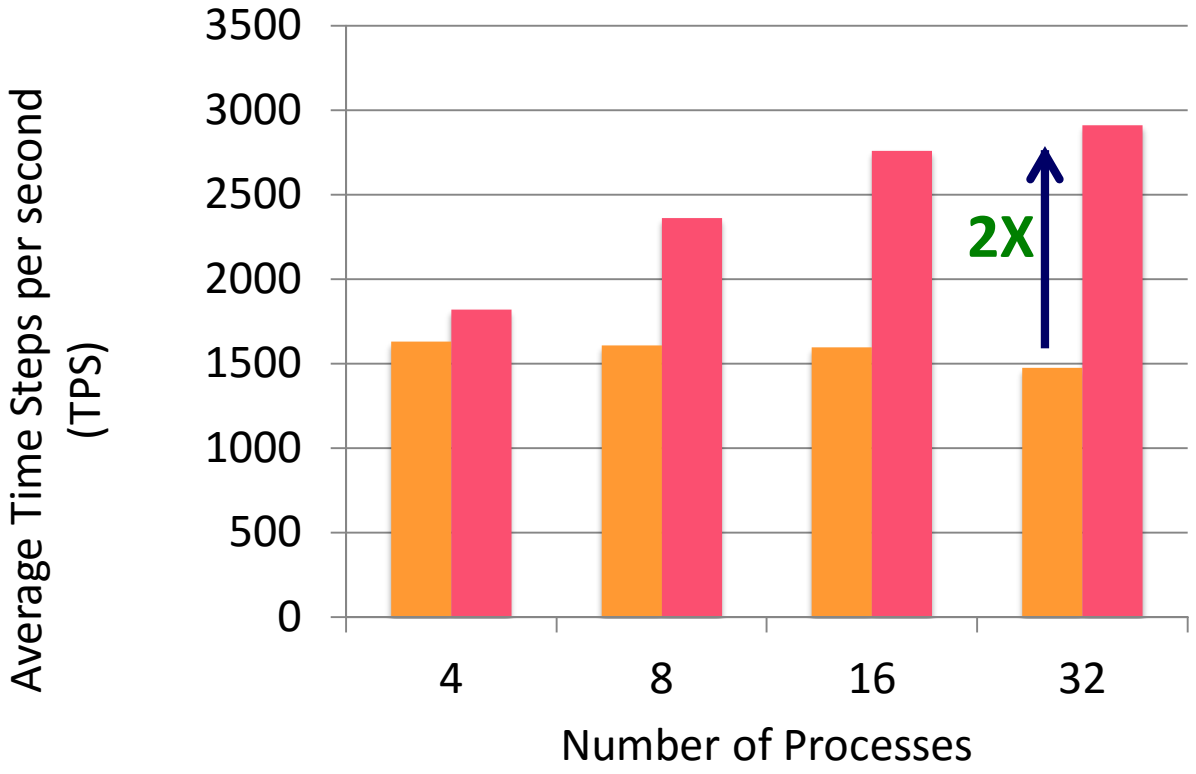
Available since MVAPICH2-GDR 2.3a

Inter-node Bandwidth: 23.7 GB/sec (2 port EDR)

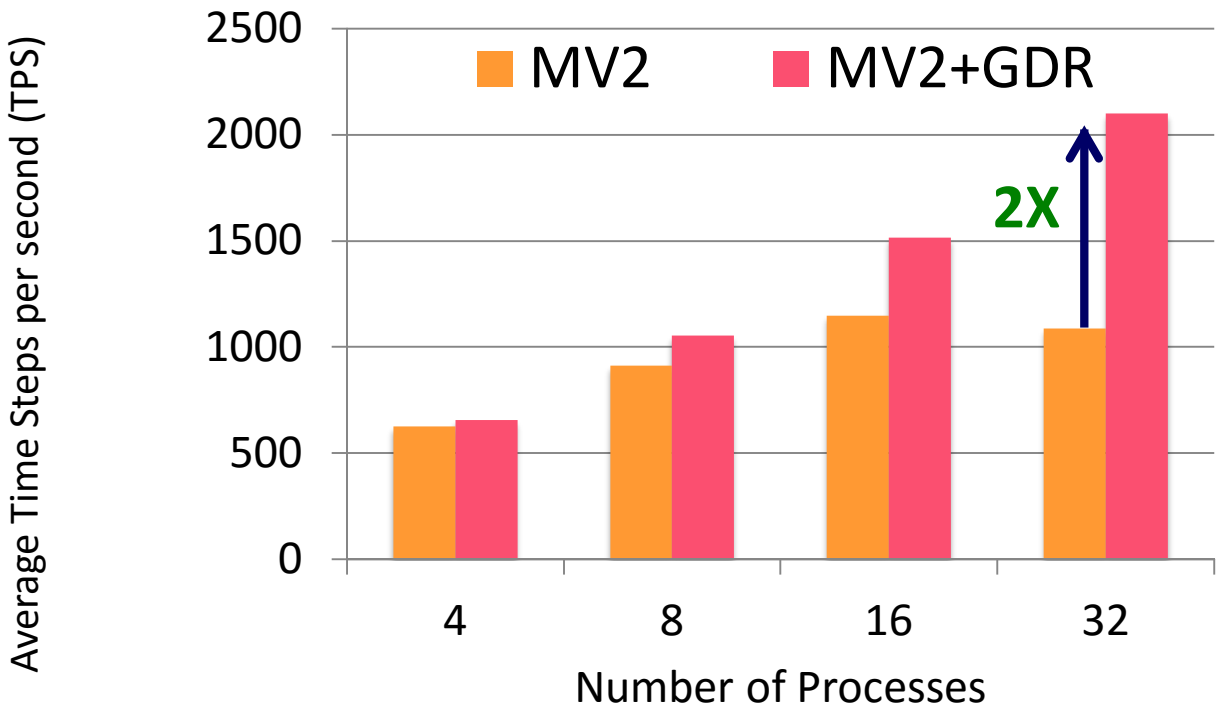
Platform: OpenPOWER (POWER9-ppc64le) nodes equipped with a dual-socket CPU, 4 Volta V100 GPUs, and 2port EDR InfiniBand Interconnect

Application-Level Evaluation (HOOMD-blue)

64K Particles

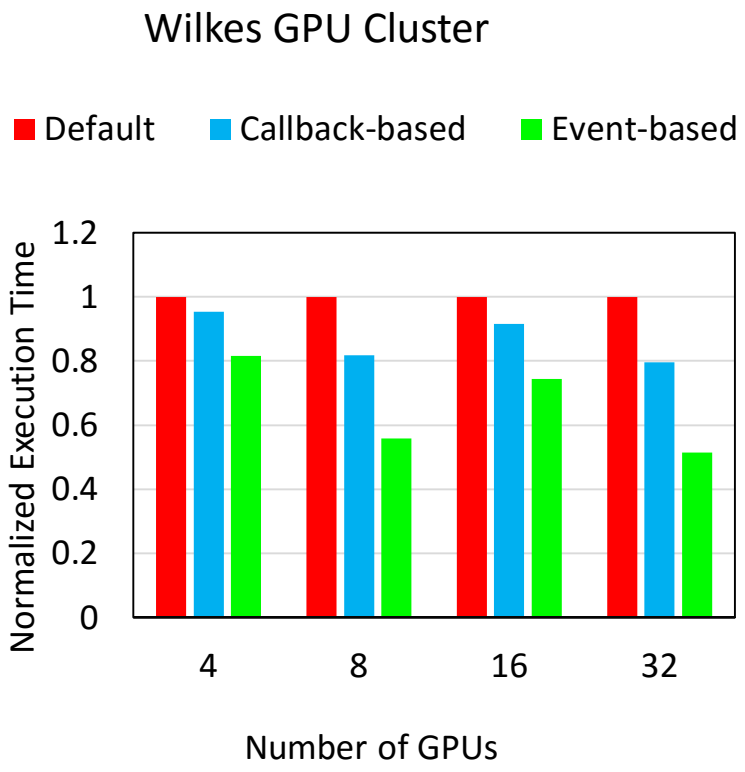


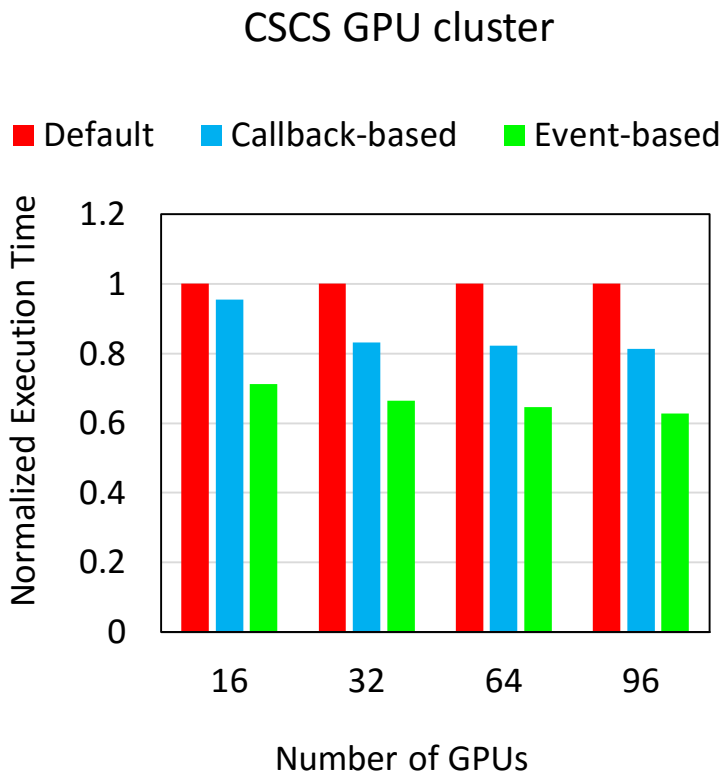
256K Particles

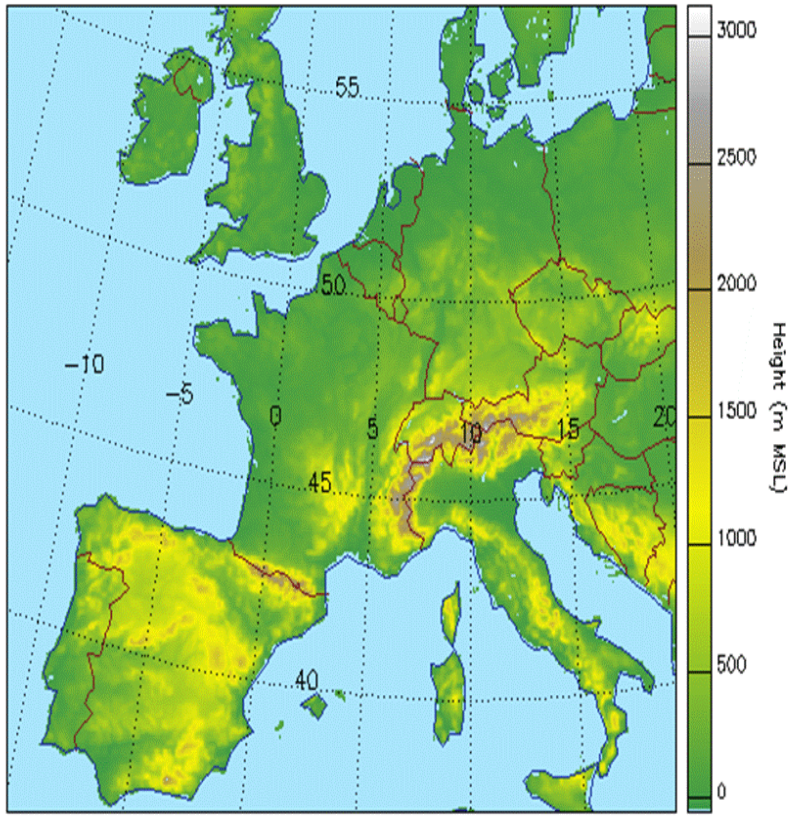


- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- HoomDBlue Version 1.0.5
  - GDRCOPY enabled: MV2_USE_CUDA=1 MV2_IBA_HCA=mlx5_0 MV2_IBA_EAGER_THRESHOLD=32768 MV2_VBUF_TOTAL_SIZE=32768 MV2_USE_GPUDIRECT_LOOPBACK_LIMIT=32768 MV2_USE_GPUDIRECT_GDRCOPY=1 MV2_USE_GPUDIRECT_GDRCOPY_LIMIT=16384

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







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

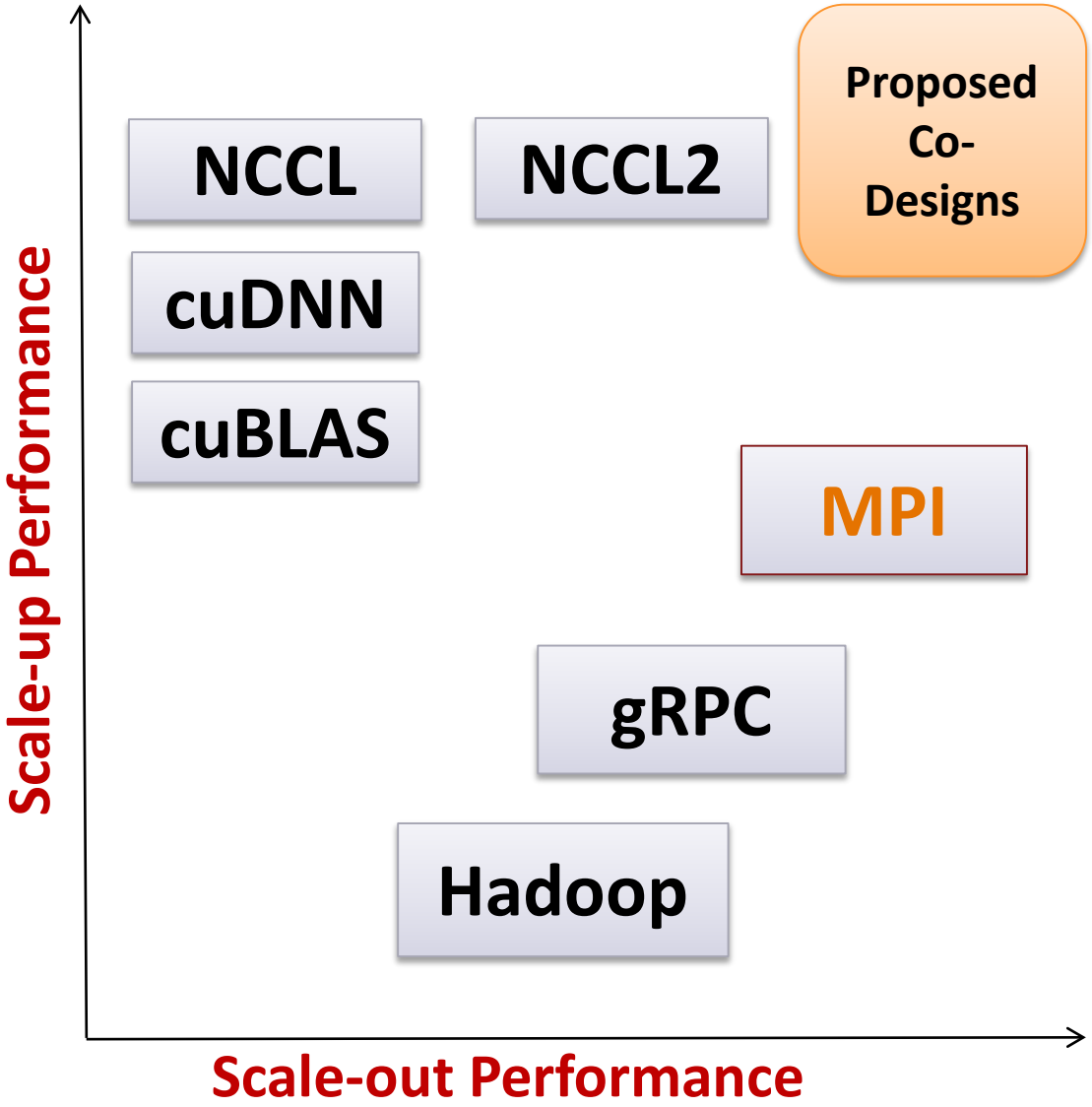
Cosmo model: http://www2.cosmo-model.org/content/tasks/operational/meteoSwiss/

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

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

Deep Learning: New Challenges for MPI Runtimes

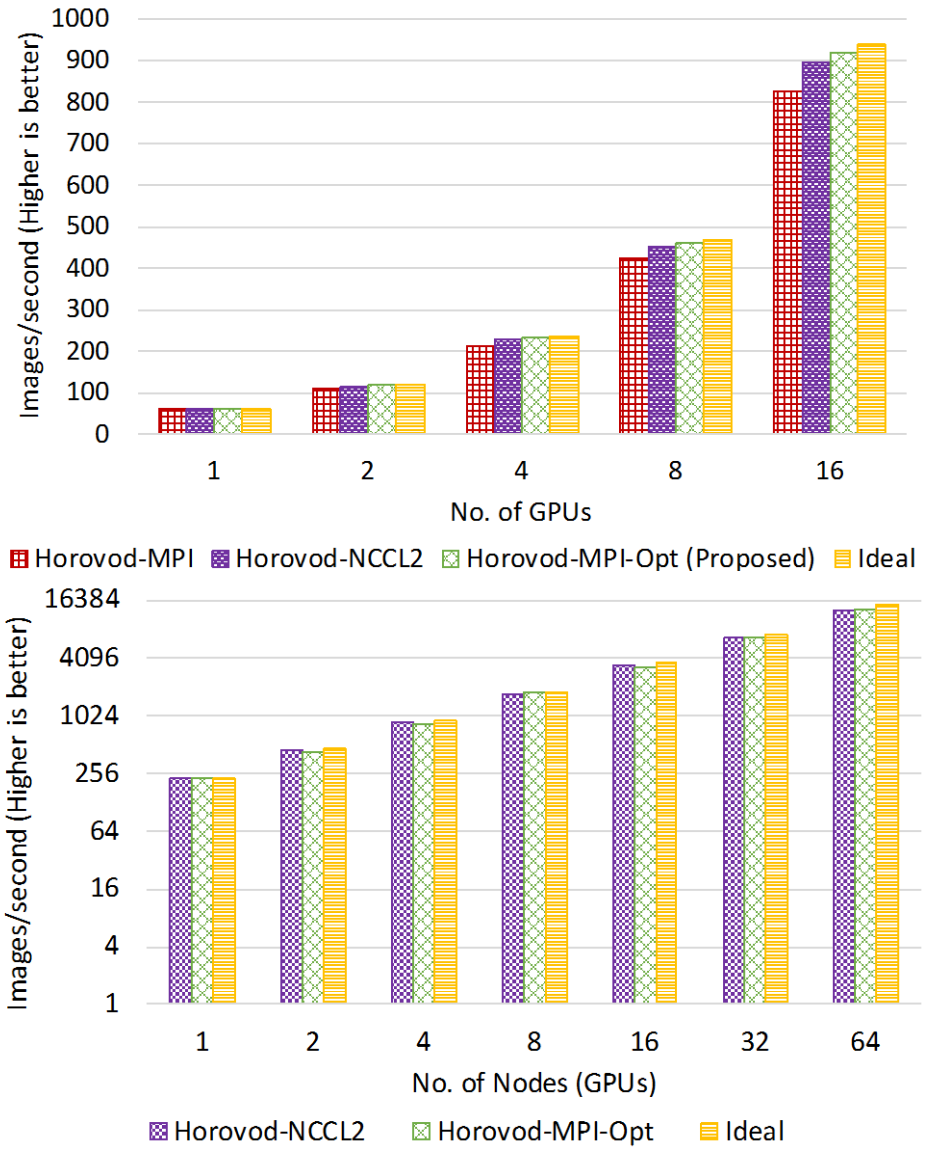
- Deep Learning frameworks are a different game altogether
  - Unusually large message sizes (order of megabytes)
  - Most communication based on GPU buffers
- Existing State-of-the-art
  - cuDNN, cuBLAS, NCCL --> **scale-up** performance
  - NCCL2, CUDA-Aware MPI --> **scale-out** performance
    -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



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

Scalable TensorFlow using Horovod, MPI, and NCCL

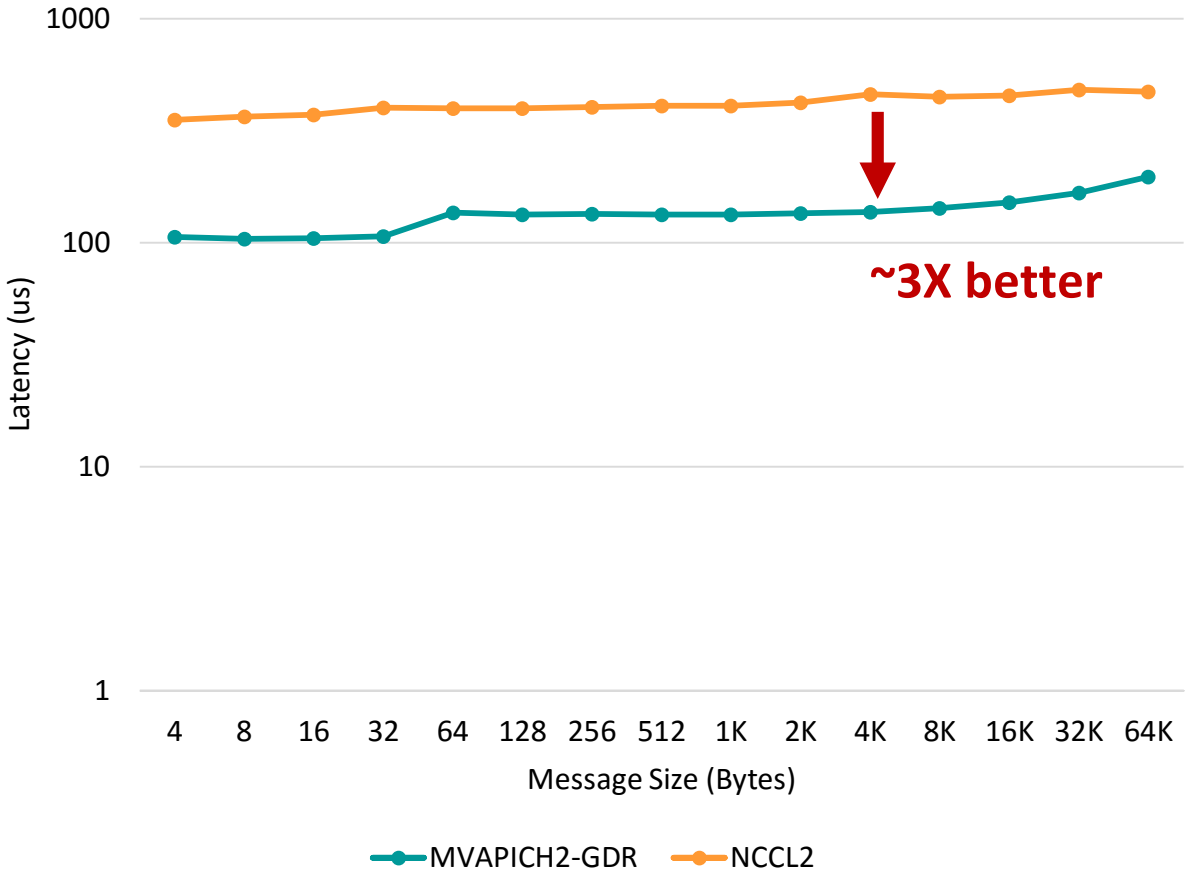
- Efficient Allreduce is crucial for Horovod's overall training performance
  - Both MPI and NCCL designs are available
- We have evaluated Horovod extensively and compared across a wide range of designs using gRPC and gRPC extensions
- MVAPICH2-GDR achieved up to ***90%*** scaling efficiency for ResNet-50 Training on 64 Pascal GPUs

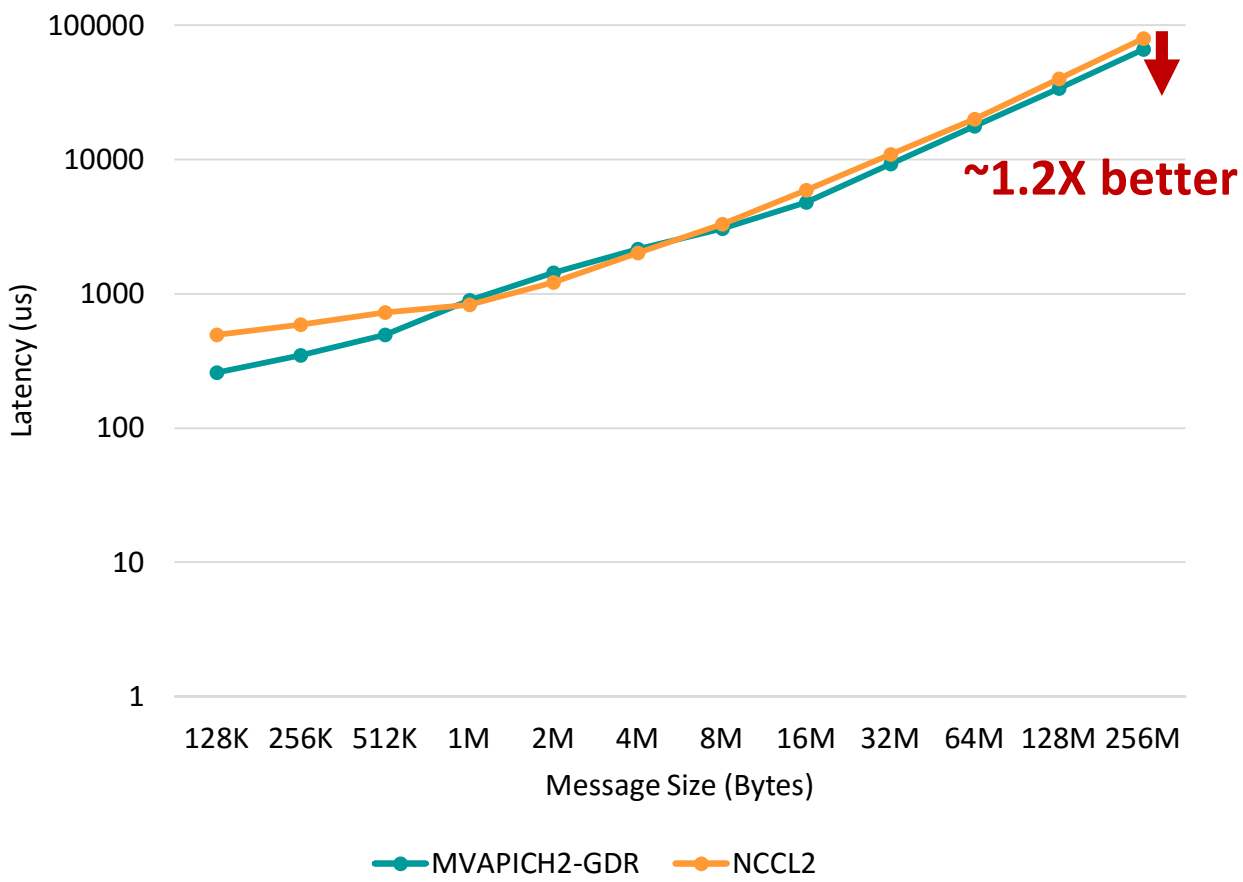


Awan et al., "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs, and Performance Evaluation", CCGrid '19. https://arxiv.org/abs/1810.11112

MVAPICH2-GDR vs. NCCL2 – Allreduce Operation

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

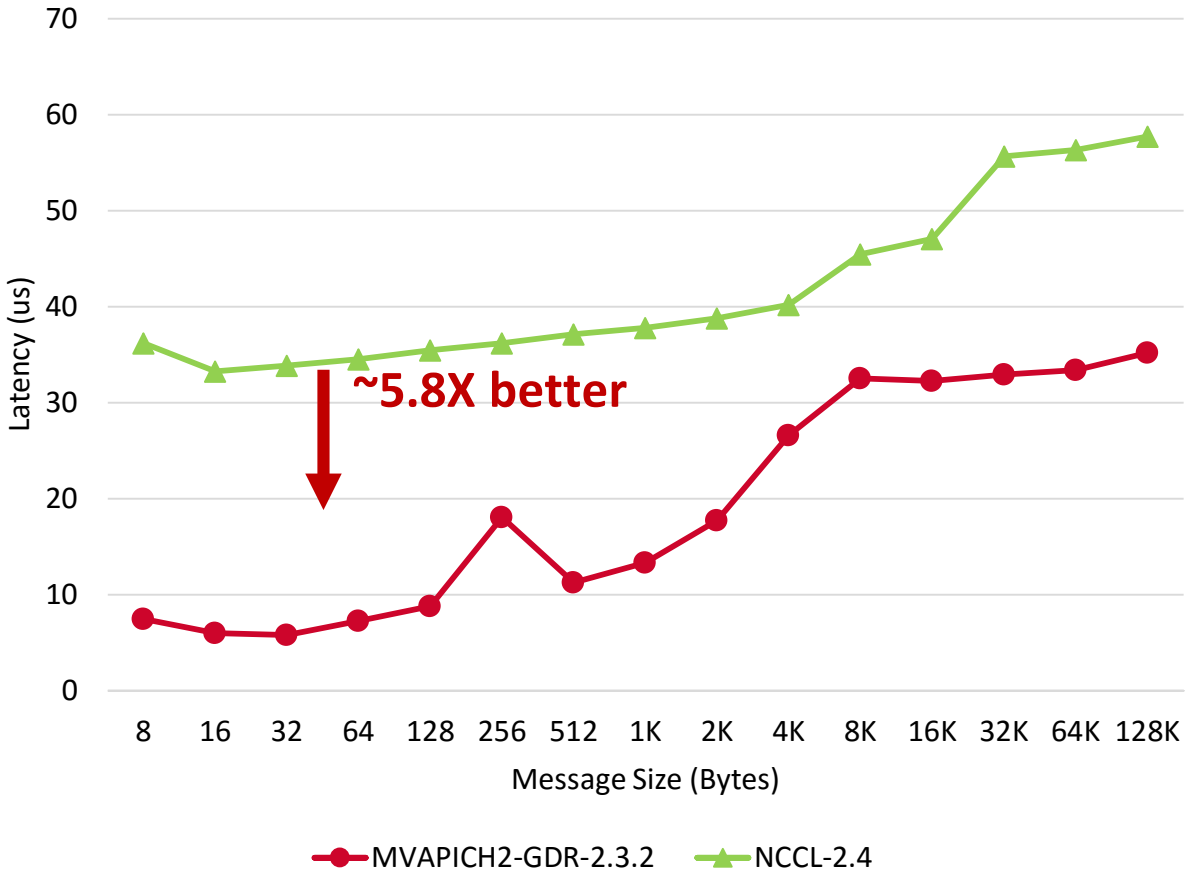


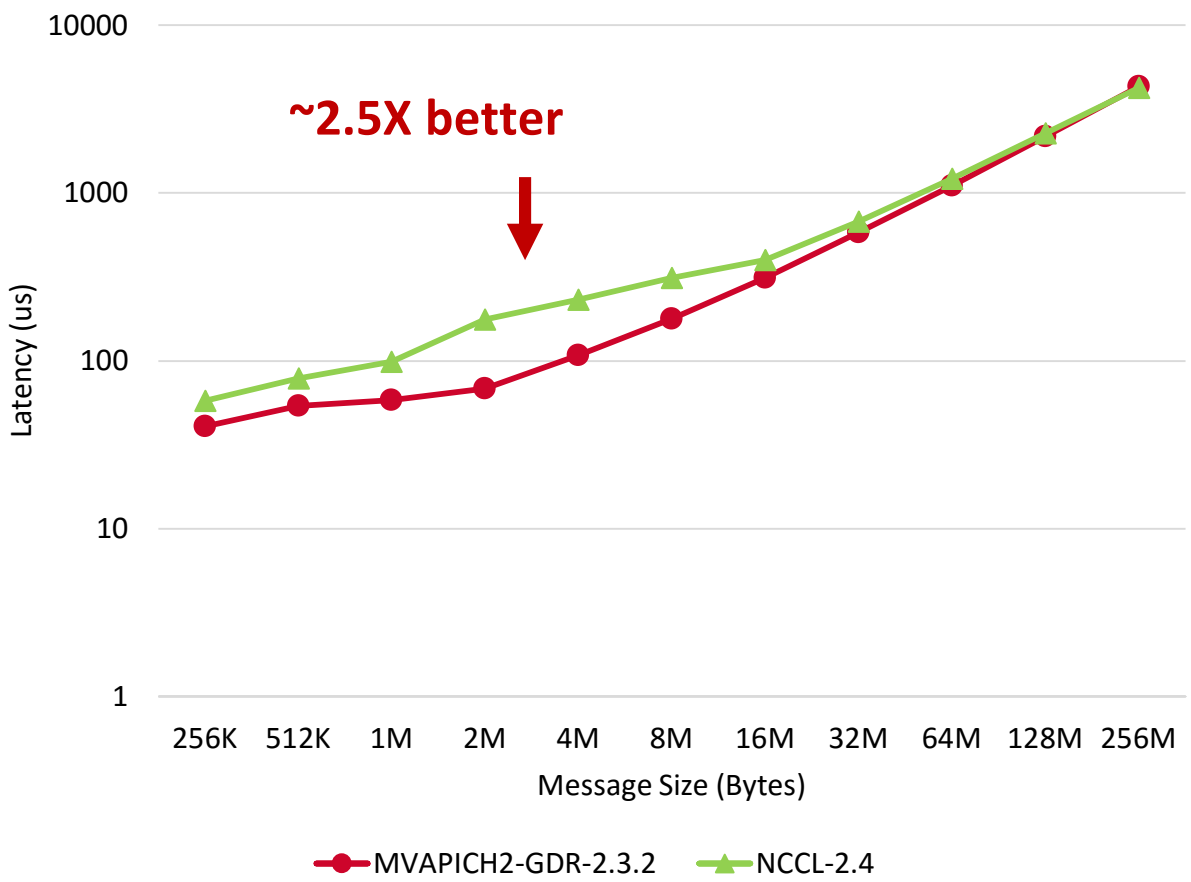


Platform: Intel Xeon (Broadwell) nodes equipped with a dual-socket CPU, 1 K-80 GPUs, and EDR InfiniBand Inter-connect

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

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

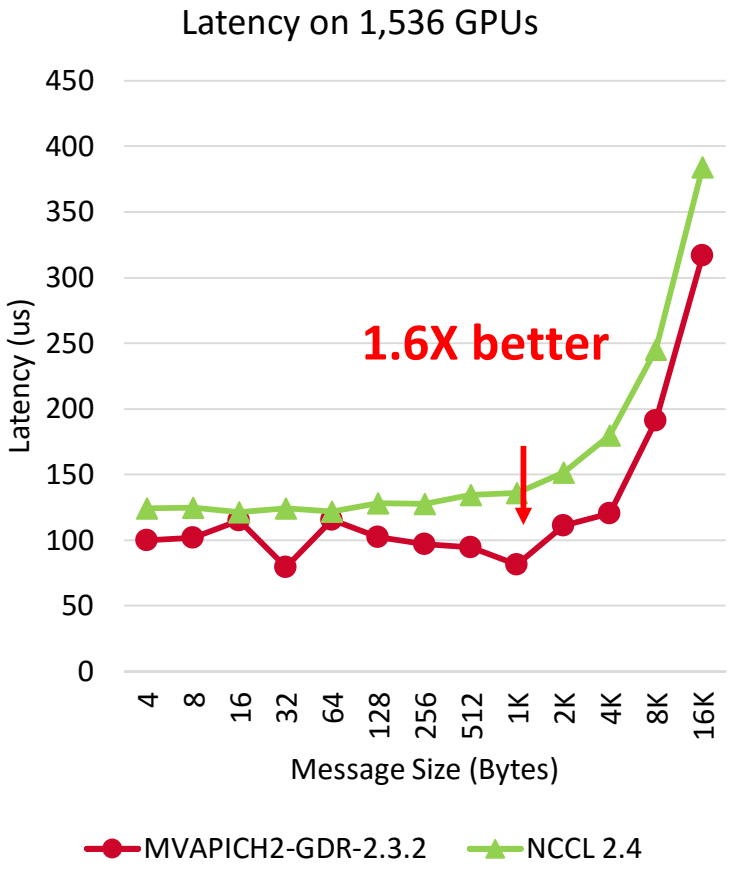


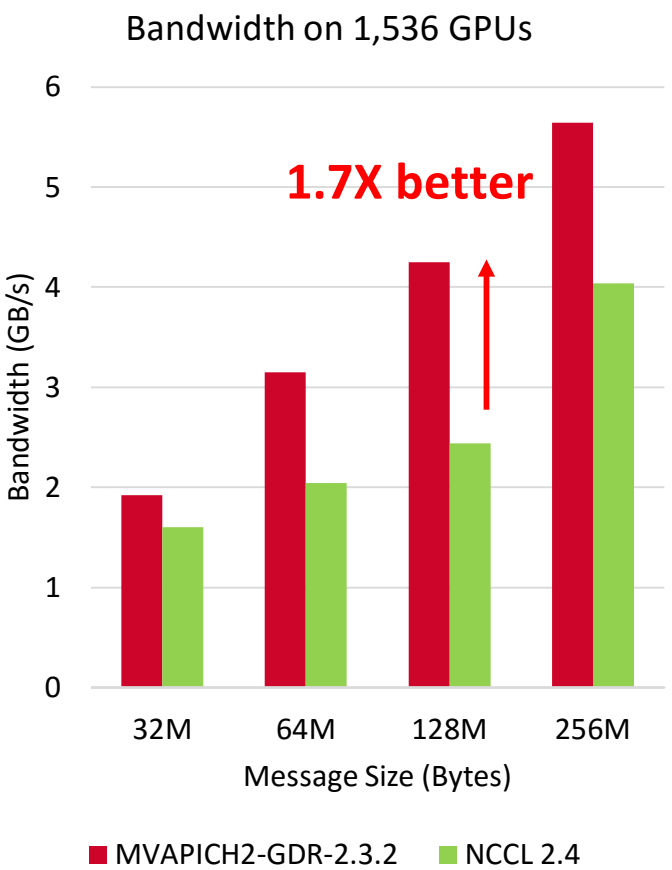


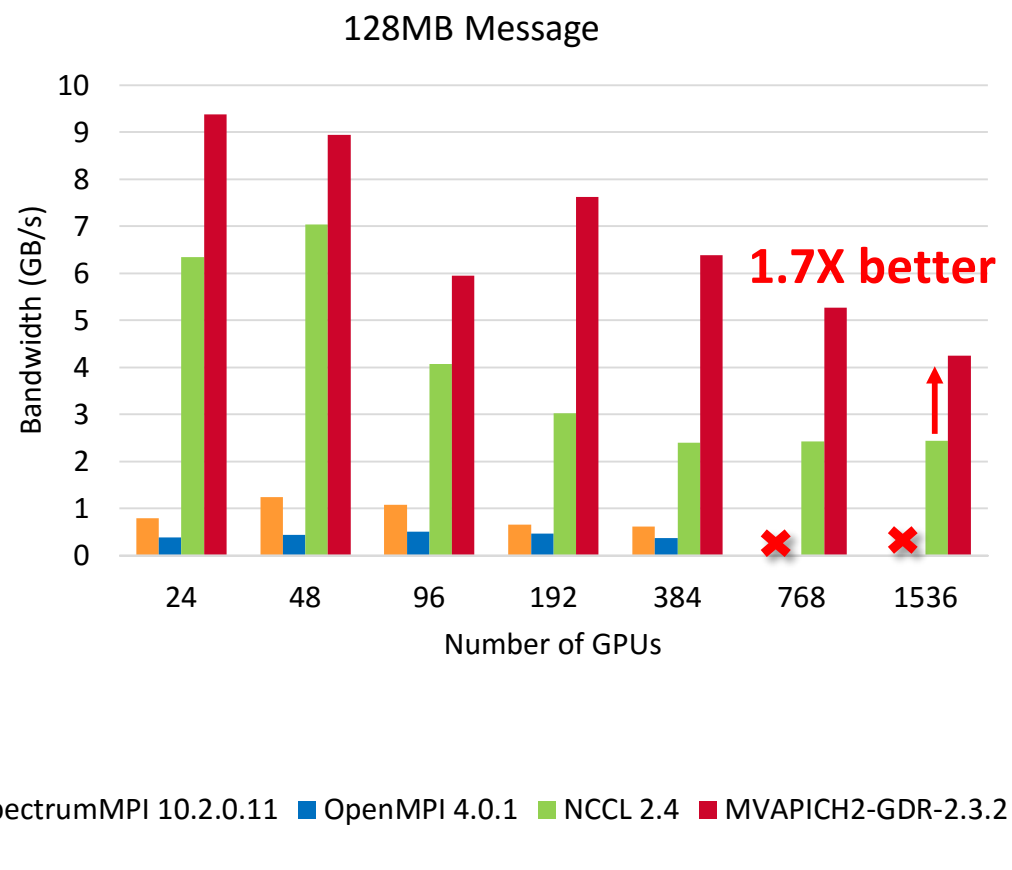
Platform: Nvidia DGX-2 system (16 Nvidia Volta GPUs connected with NVSwitch), CUDA 9.2

MVAPICH2-GDR: Enhanced MPI_Allreduce at Scale

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



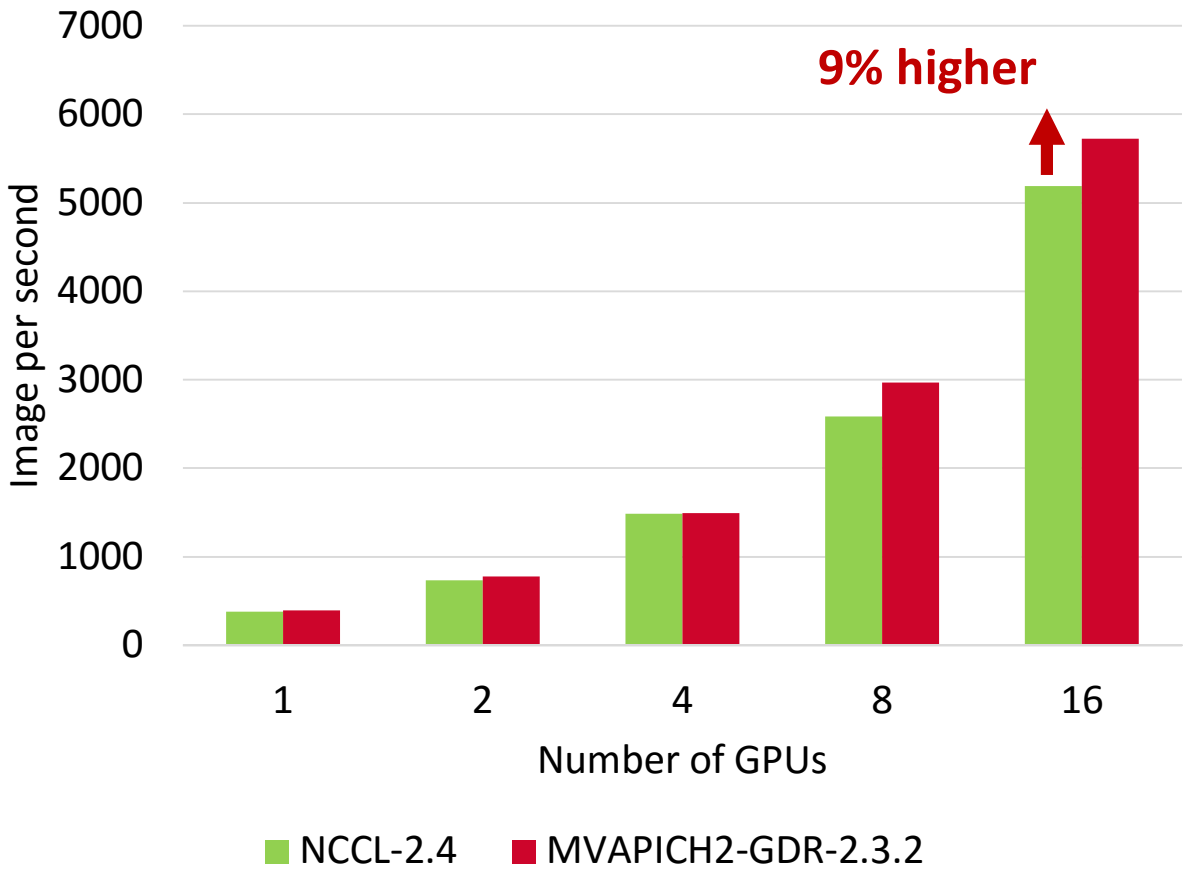




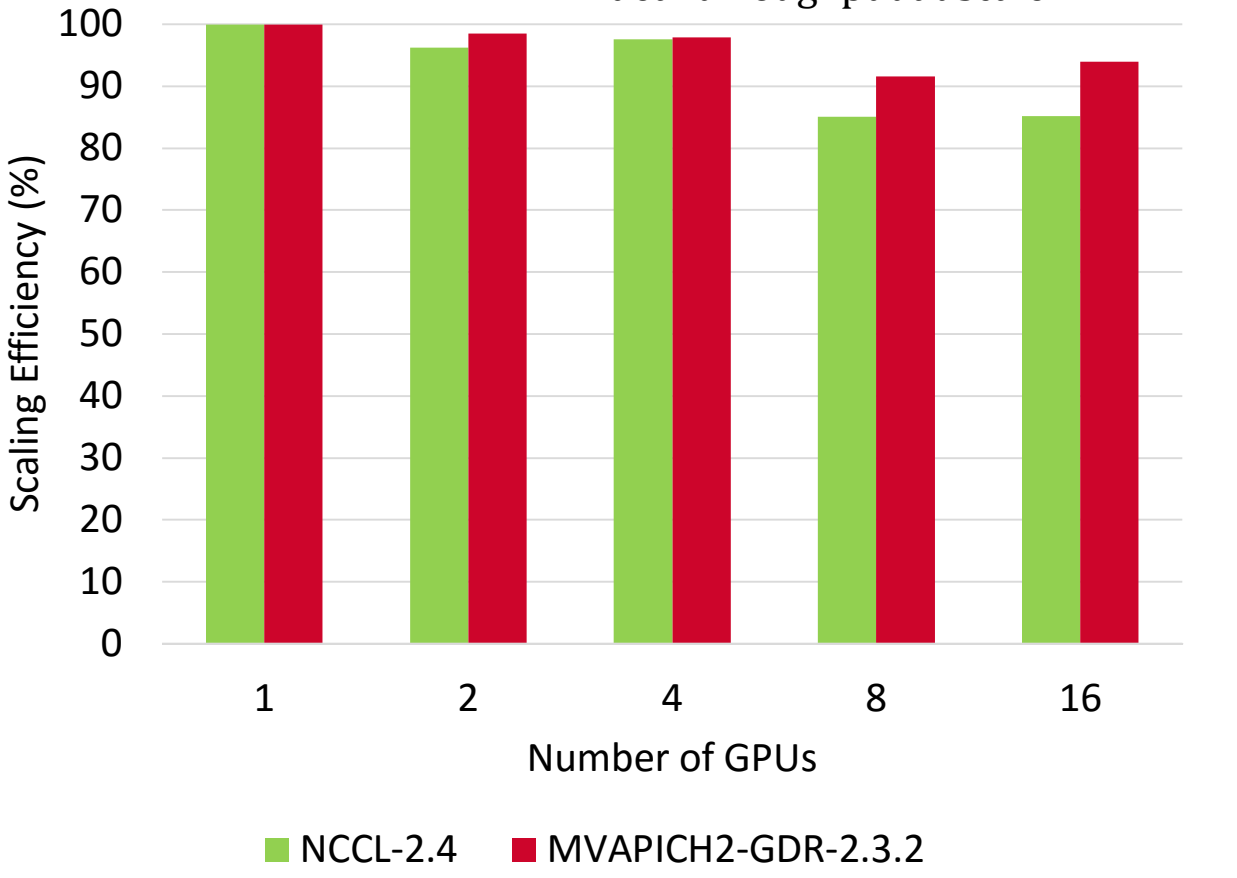
Platform: Dual-socket IBM POWER9 CPU, 6 NVIDIA Volta V100 GPUs, and 2-port InfiniBand EDR Interconnect

Distributed Training with TensorFlow and MVAPICH2-GDR

- **ResNet-50 Training using TensorFlow benchmark on 1 DGX-2 node (16 Volta GPUs)**



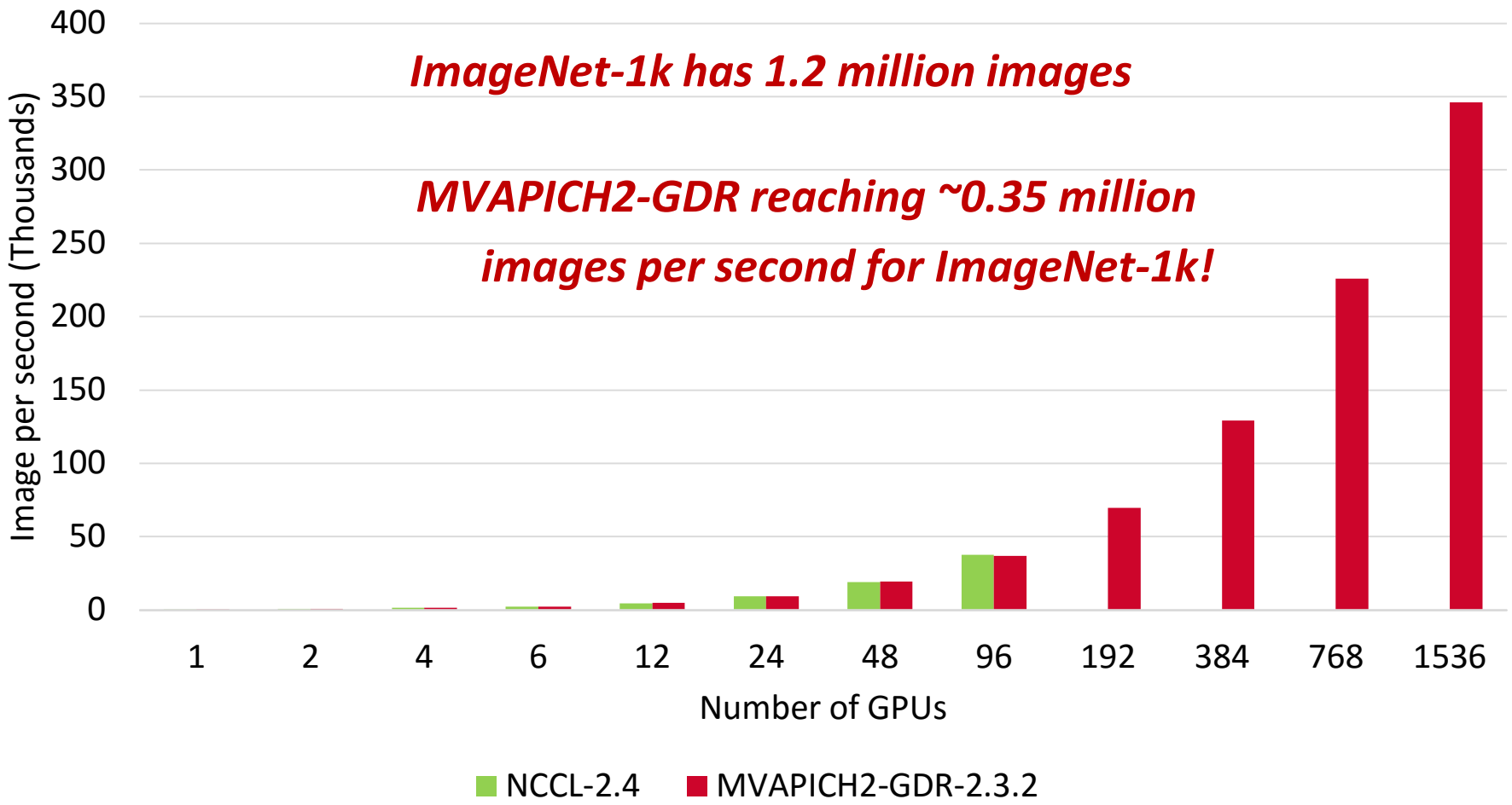
$$\text{Scaling Efficiency} = \frac{\text{Actual throughput}}{\text{Ideal throughput at scale}} \times 100\%$$



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

Distributed Training with TensorFlow and MVAPICH2-GDR

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



*We observed errors for NCCL2 beyond 96 GPUs

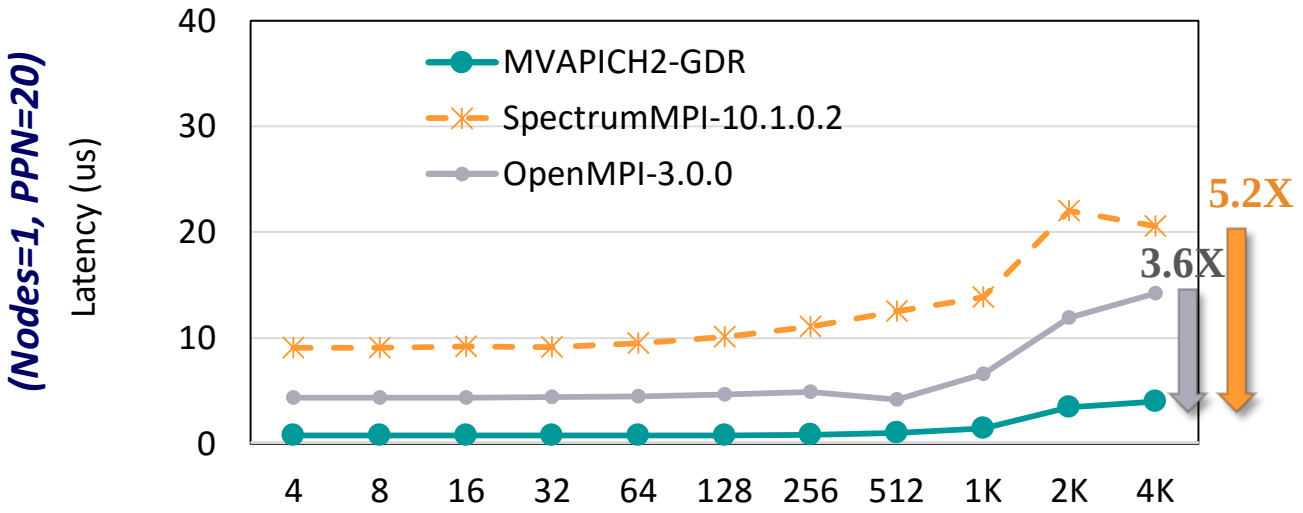
Platform: The Summit Supercomputer (#1 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 9.2

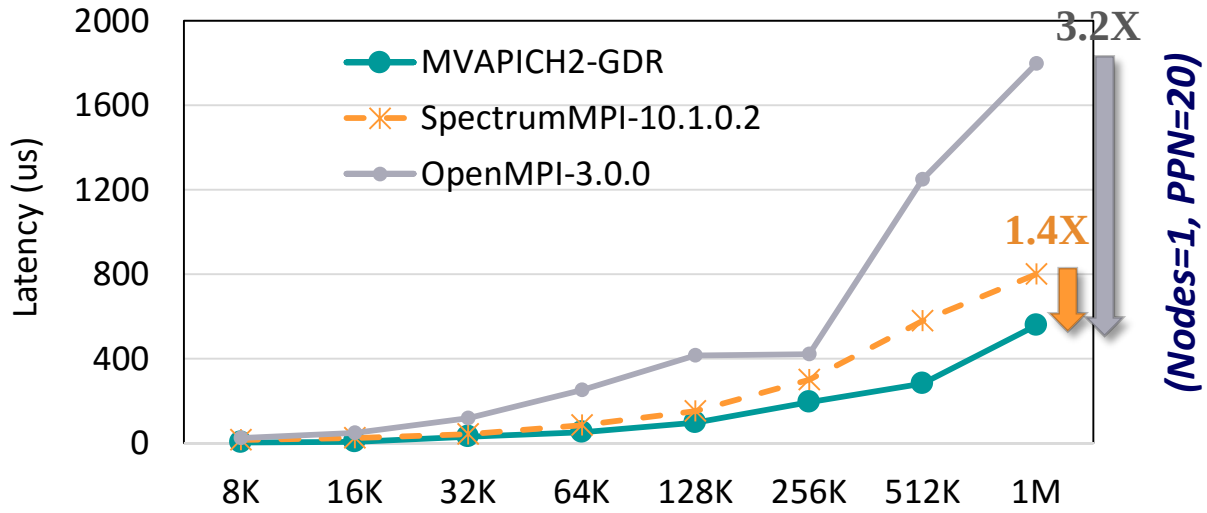
MVAPICH2-GDR Upcoming Features for HPC and DL

- Scalable Host-based Collectives
- Enhanced Derived Datatype
- Integrated Collective Support with SHArP from GPU Buffers
- Optimization for PyTorch and MXNET

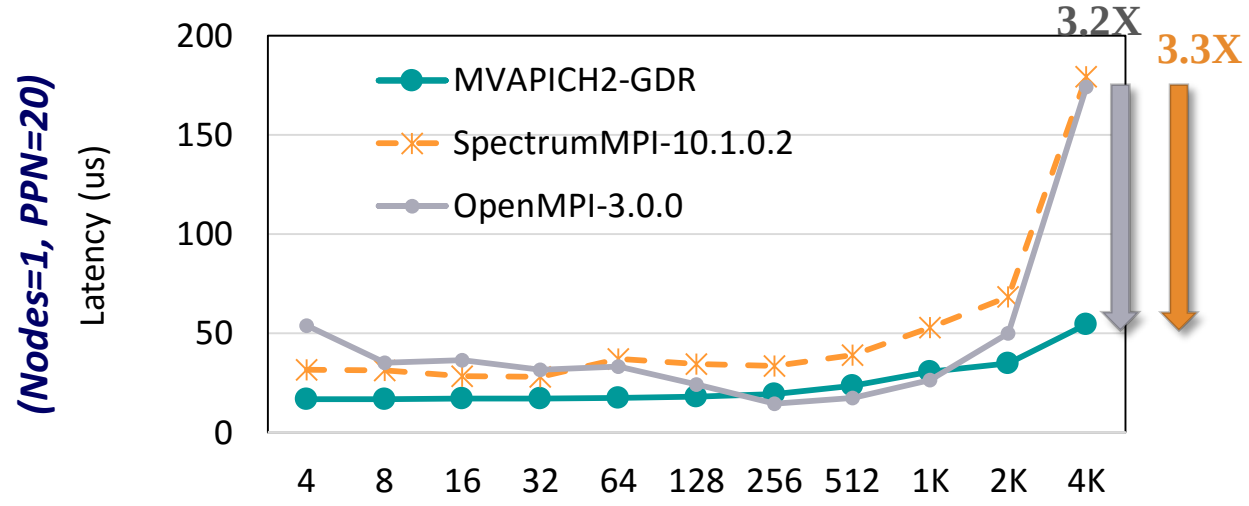
Scalable Host-based Collectives on OpenPOWER (Intra-node Reduce & AlltoAll)

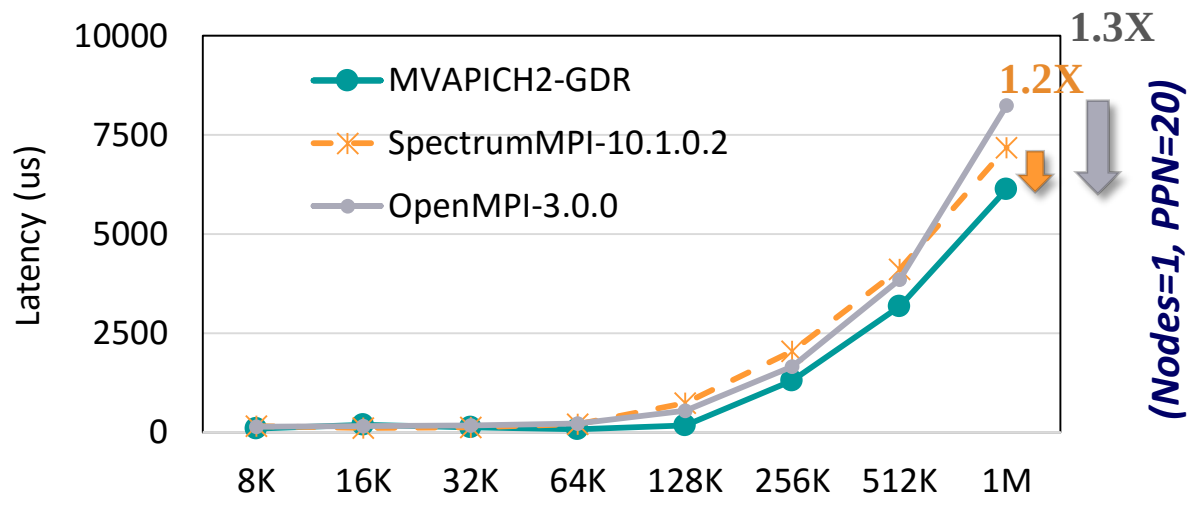
Reduce





Alltoall

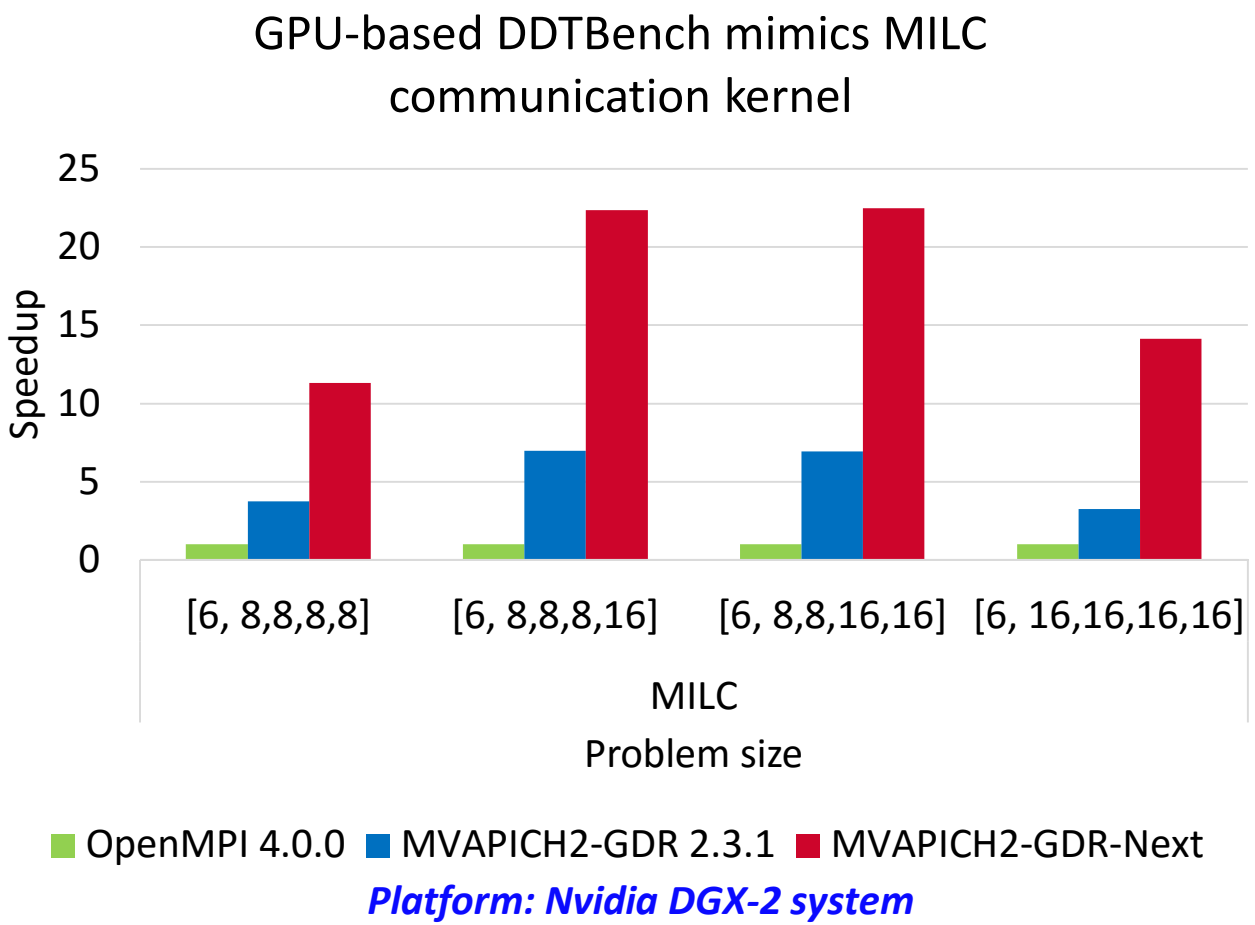




Up to 5X and 3x performance improvement by MVAPICH2 for small and large messages respectively

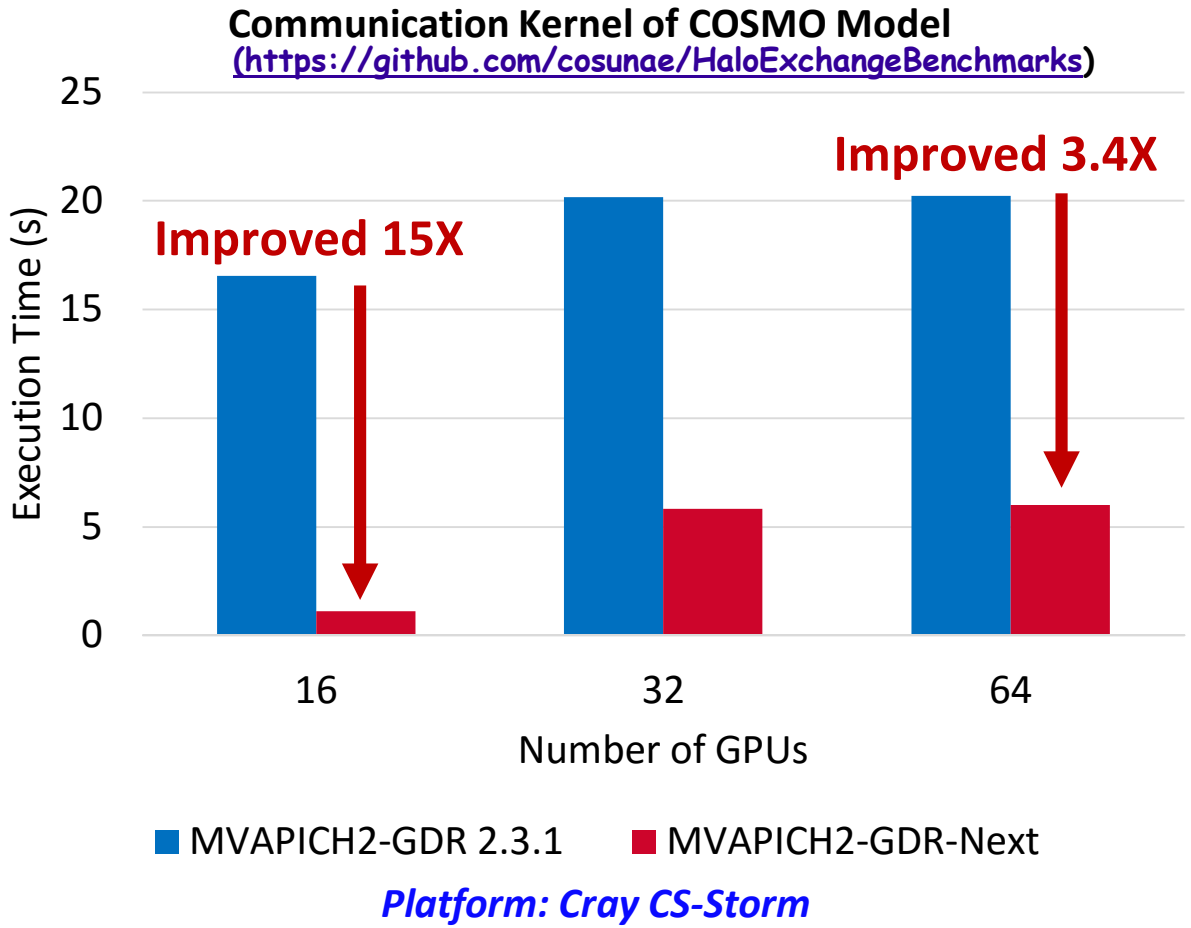
MVAPICH2-GDR: Enhanced Derived Datatype

- **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

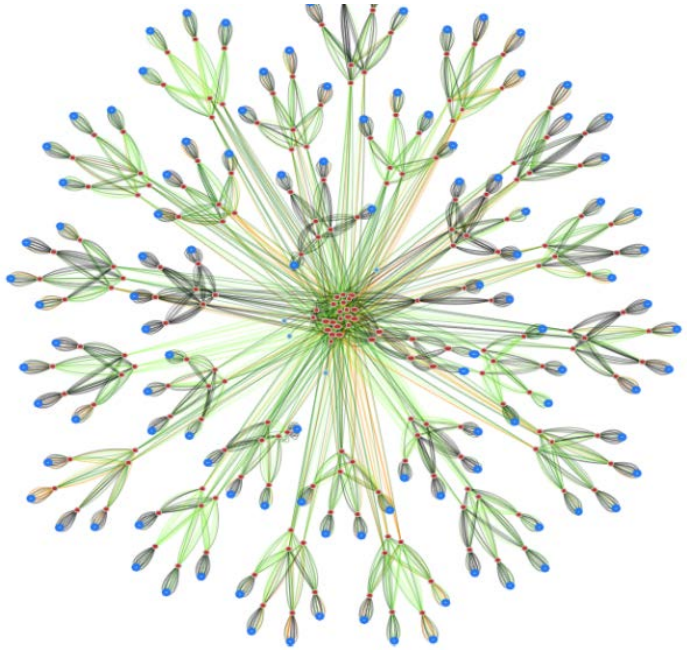
MVAPICH2 Software Family

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

Overview of OSU INAM

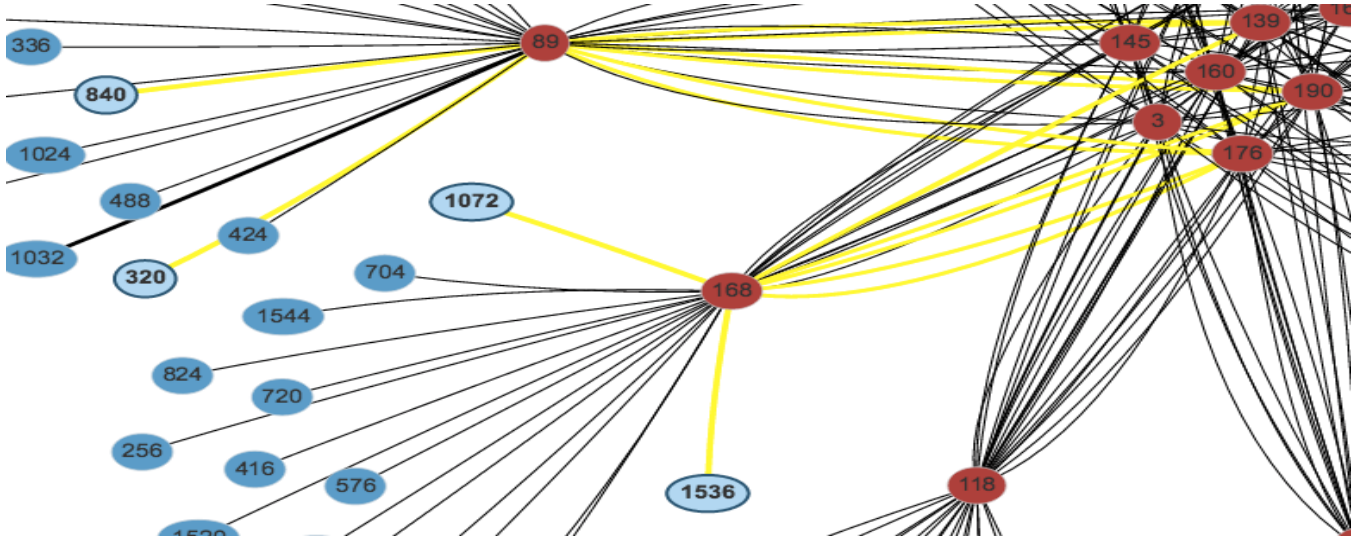
- A network monitoring and analysis tool that is capable of analyzing traffic on the InfiniBand network with inputs from the MPI runtime
  - http://mvapich.cse.ohio-state.edu/tools/osu-inam/
- Monitors IB clusters in real time by querying various subnet management entities and gathering input from the 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
- OSU INAM 0.9.4 released on 11/10/2018
  - Enhanced performance for fabric discovery using optimized OpenMP-based multi-threaded designs
  - Ability to gather InfiniBand performance counters at sub-second granularity for very large (>2000 nodes) clusters
  - Redesign database layout to reduce database size
  - Enhanced fault tolerance for database operations
    - Thanks to Trey Dockendorf @ OSC for the feedback
  - OpenMP-based multi-threaded designs to handle database purge, read, and insert operations simultaneously
  - Improved database purging time by using bulk deletes
  - Tune database timeouts to handle very long database operations
  - Improved debugging support by introducing several debugging levels

OSU INAM Features



Comet@SDSC --- Clustered View

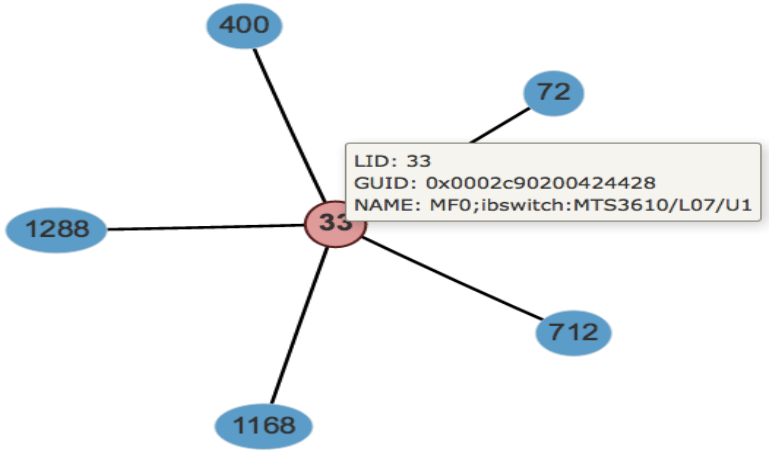
(1,879 nodes, 212 switches, 4,377 network links)



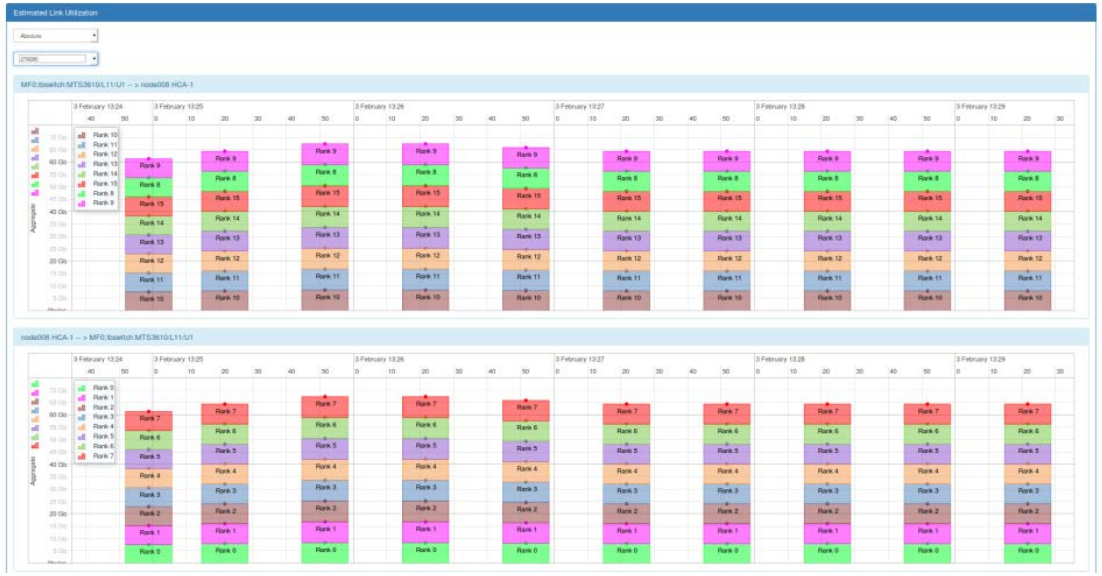
Finding Routes Between Nodes

- Show network topology of large clusters
- Visualize traffic pattern on different links
- Quickly identify congested links/links in error state
- See the history unfold – play back historical state of the network

OSU INAM Features (Cont.)



Visualizing a Job (5 Nodes)



Estimated Process Level Link Utilization

- Job level view
  - Show different network metrics (load, error, etc.) for any live job
  - Play back historical data for completed jobs to identify bottlenecks
- Node level view - details per process or per node
  - CPU utilization for each rank/node
  - Bytes sent/received for MPI operations (pt-to-pt, collective, RMA)
  - Network metrics (e.g. XmitDiscard, RcvError) per rank/node

- Estimated Link Utilization view
  - Classify data flowing over a network link at different granularity in conjunction with MVAPICH2-X 2.2rc1
    - Job level and
    - Process level

More Details in Tutorial/Demo Session Tomorrow

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++)
    - 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
- Please visit the following link for more information
  - http://mvapich.cse.ohio-state.edu/benchmarks/

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 – Plans for Exascale

- Performance and Memory scalability toward 1-10M cores
- Hybrid programming (MPI + OpenSHMEM, MPI + UPC, MPI + CAF …)
  - MPI + Task*
- Enhanced Optimization for GPU Support and Accelerators
- Taking advantage of advanced features of Mellanox InfiniBand
  - Tag Matching*
  - Adapter Memory*
- Enhanced communication schemes for upcoming architectures
  - Intel Optane*
  - BlueField*
  - CAPI*
- 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

Commercial Support for MVAPICH2, HiBD, and HiDL Libraries

- Supported through X-ScaleSolutions (http://x-scalesolutions.com)
- Benefits:
  - Help and guidance with installation of the library
  - Platform-specific optimizations and tuning
  - Timely support for operational issues encountered with the library
  - Web portal interface to submit issues and tracking their progress
  - Advanced debugging techniques
  - Application-specific optimizations and tuning
  - Obtaining guidelines on best practices
  - Periodic information on major fixes and updates
  - Information on major releases
  - Help with upgrading to the latest release
  - Flexible Service Level Agreements
- Support provided to Lawrence Livermore National Laboratory (LLNL) for the last two years

X-ScaleSolutions

Silver ISV Member for the OpenPOWER Consortium + Products

- Has joined the OpenPOWER Consortium as a silver ISV member
- Provides flexibility:
  - To have MVAPICH2, HiDL and HiBD libraries getting integrated into the OpenPOWER software stack
  - A part of the OpenPOWER ecosystem
  - Can participate with different vendors for bidding, installation and deployment process
- Introduced two new integrated products with support for OpenPOWER systems (Presented yesterday at the OpenPOWER North America Summit)
  - X-ScaleHPC
  - X-ScaleAI
  - Send an e-mail to contactus@x-scalesolutions.com for free trial!!

X-ScaleSolutions

Funding Acknowledgments

Funding Support by

Equipment Support by

Personnel Acknowledgments

Current Students (Graduate)

- A. Awan (Ph.D.)
- M. Bayatpour (Ph.D.)
- C.-H. Chu (Ph.D.)
- J. Hashmi (Ph.D.)
- A. Jain (Ph.D.)
- K. S. Kandadi (M.S.)
- Kamal Raj (M.S.)
- K. S. Khorassani (Ph.D.)
- P. Kousha (Ph.D.)
- A. Quentin (Ph.D.)
- B. Ramesh (M. S.)
- S. Xu (M.S.)
- Q. Zhou (Ph.D.)

Current Research Scientist

- H. Subramoni

Current Post-doc

- M. S. Ghazimeersaeed
- A. Ruhela
- K. Manian

Current Students (Undergraduate)

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

Current Research Specialist

- J. Smith

Past Students

- A. Augustine (M.S.)
- P. Balaji (Ph.D.)
- R. Biswas (M.S.)
- S. Bhagvat (M.S.)
- A. Bhat (M.S.)
- D. Buntinas (Ph.D.)
- L. Chai (Ph.D.)
- B. Chandrasekharan (M.S.)
- S. Chakraborthy (Ph.D.)
- N. Dandapanthula (M.S.)
- V. Dhanraj (M.S.)
- T. Gangadharappa (M.S.)
- K. Gopalakrishnan (M.S.)
- W. Huang (Ph.D.)
- W. Jiang (M.S.)
- J. Jose (Ph.D.)
- S. Kini (M.S.)
- M. Koop (Ph.D.)
- K. Kulkarni (M.S.)
- R. Kumar (M.S.)
- S. Krishnamoorthy (M.S.)
- K. Kandalla (Ph.D.)
- M. Li (Ph.D.)
- P. Lai (M.S.)
- J. Liu (Ph.D.)
- M. Luo (Ph.D.)
- A. Mamidala (Ph.D.)
- G. Marsh (M.S.)
- V. Meshram (M.S.)
- A. Moody (M.S.)
- S. Naravula (Ph.D.)
- R. Noronha (Ph.D.)
- X. Ouyang (Ph.D.)
- S. Pai (M.S.)
- S. Potluri (Ph.D.)
- R. Rajachandrasekar (Ph.D.)
- D. Shankar (Ph.D.)
- G. Santhanaraman (Ph.D.)
- A. Singh (Ph.D.)
- J. Sridhar (M.S.)
- S. Sur (Ph.D.)
- H. Subramoni (Ph.D.)
- K. Vaidyanathan (Ph.D.)
- A. Vishnu (Ph.D.)
- J. Wu (Ph.D.)
- W. Yu (Ph.D.)
- J. Zhang (Ph.D.)

Past Research Scientist

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

Past Programmers

- D. Bureddy
- J. Perkins

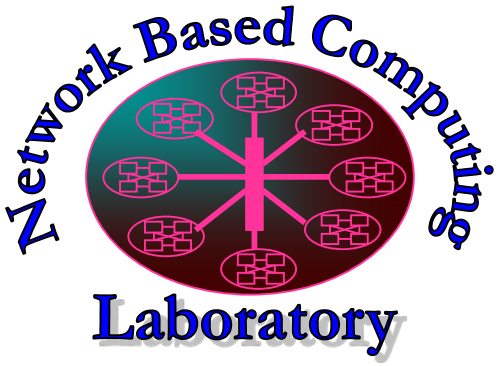
Past Research Specialist

- M. Arnold

Past Post-Docs

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

Thank You!



Network-Based Computing Laboratory
http://nowlab.cse.ohio-state.edu/

The MVAPICH2 Project
http://mvapich.cse.ohio-state.edu/