MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

Overview of the MVAPICH Project: Latest Status and Future Roadmap

MVAPICH2 User Group (MUG) Meeting

by

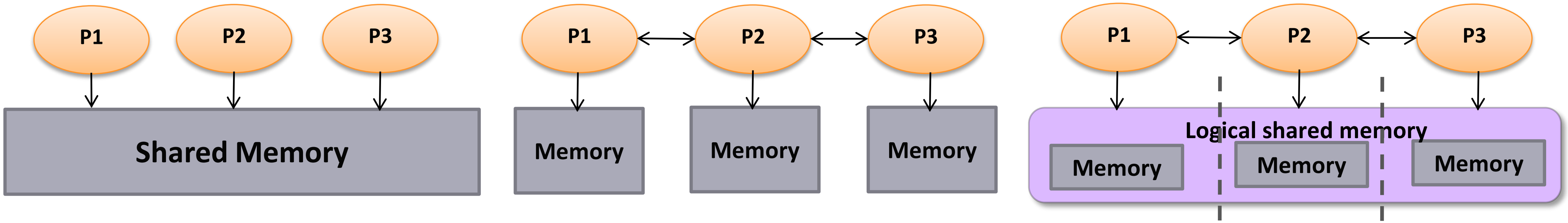
Dhabaleswar K. (DK) Panda

The Ohio State University

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

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

Parallel Programming Models Overview



Shared Memory Model

SHMEM, DSM

Distributed Memory Model

MPI (Message Passing Interface)

Partitioned Global Address Space (PGAS)

Global Arrays, UPC, UPC++, Chapel, , CAF, …

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

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

Application Kernels/Applications

Middleware

Programming Models

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

Communication Library or Runtime for Programming Models

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

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

Multi-/Many-core Architectures

Accelerators (GPU and MIC)

Co-Design Opportunities and Challenges across Various Layers

Performance

Scalability

Resilience

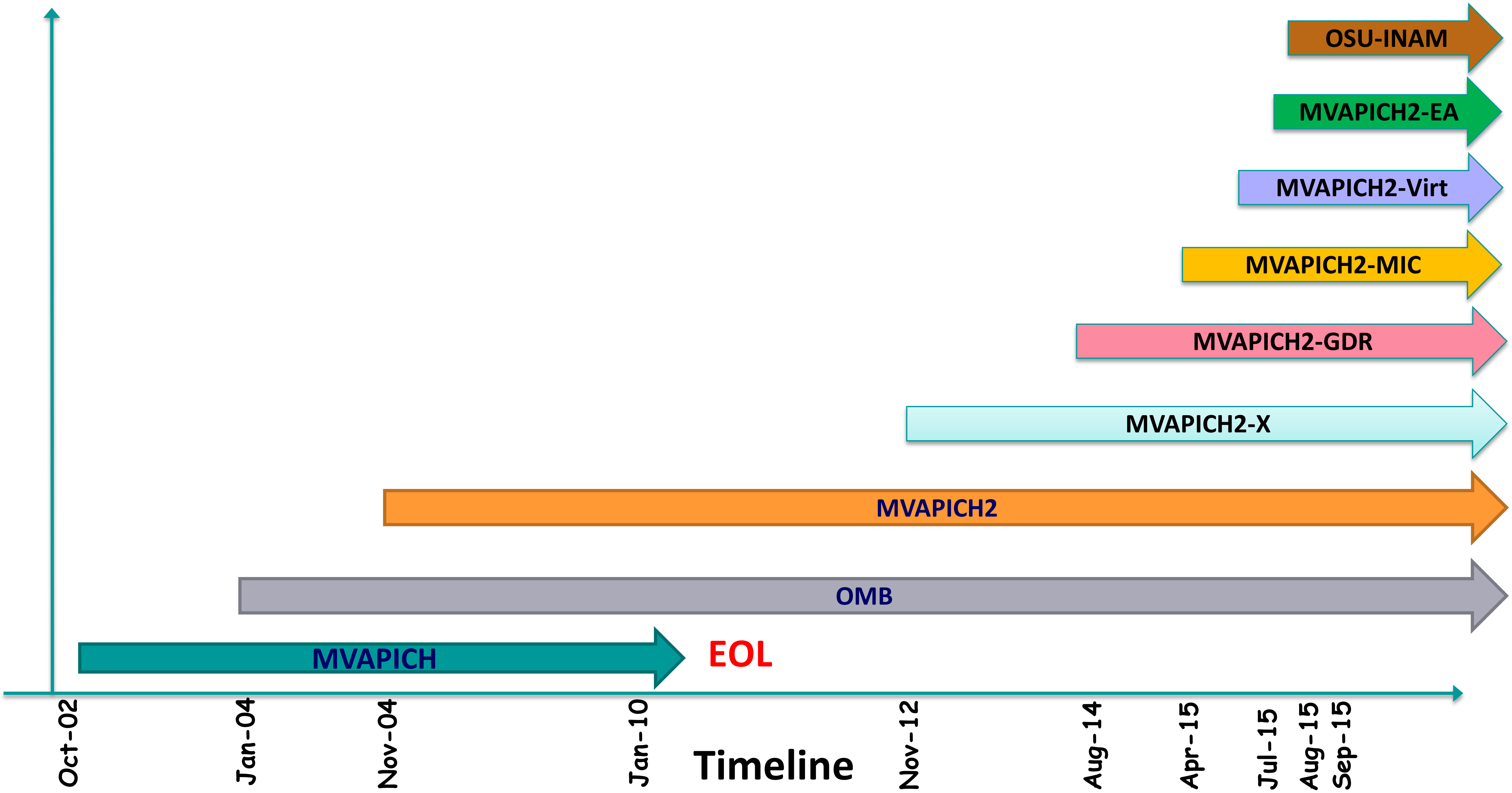
Designing (MPI+X) at Exascale

- 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

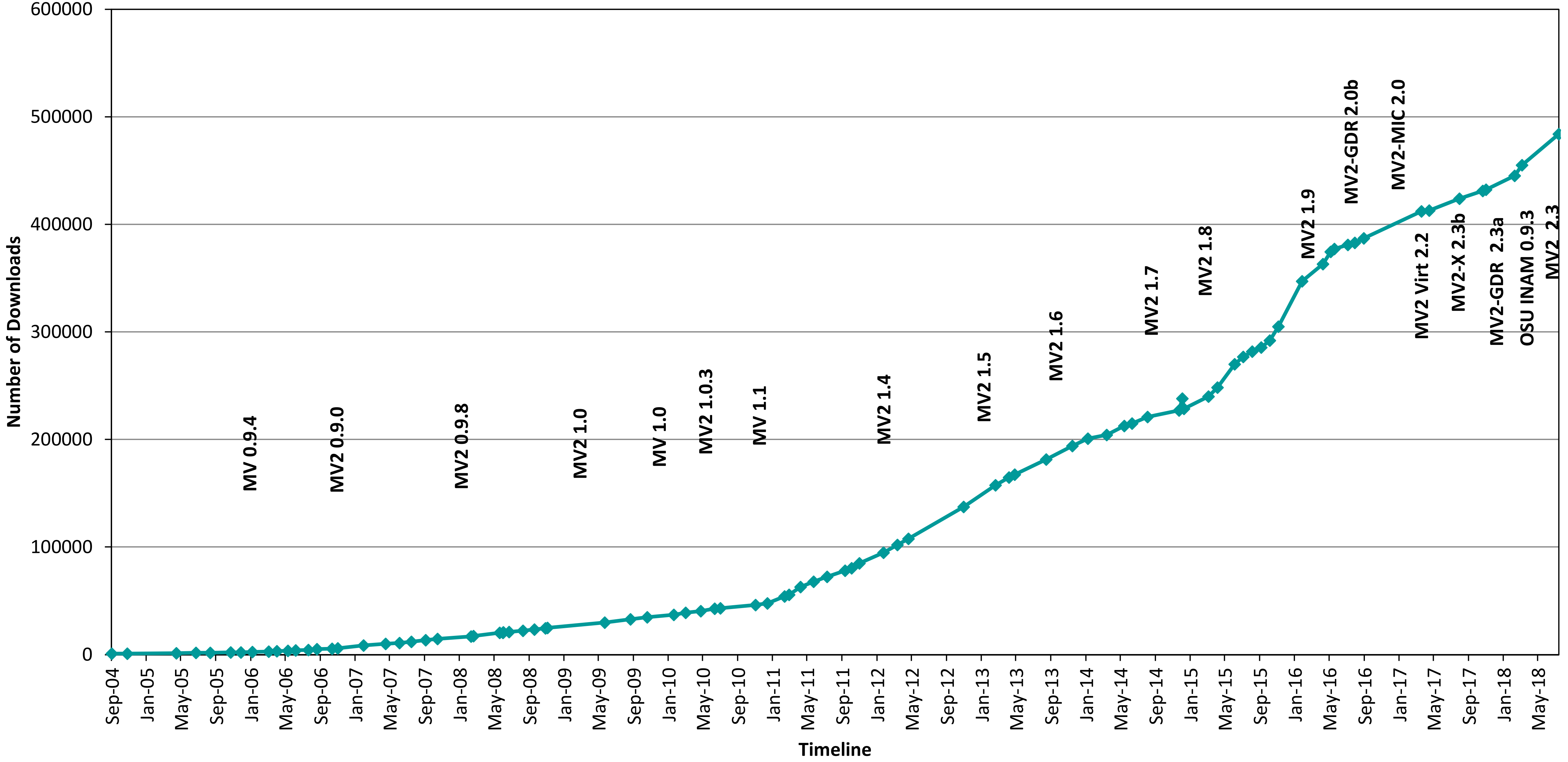
Overview of the MVAPICH2 Project

- 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 2,925 organizations in 86 countries**
  - **More than 485,000 (> 0.48 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Jul '18 ranking)
    - 2nd ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, China
    - 12th, 556,104 cores (Oakforest-PACS) in Japan
    - 15th, 367,024 cores (Stampede2) at TACC
    - 24th, 241,108-core (Pleiades) at NASA and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat and SuSE)
  - **http://mvapich.cse.ohio-state.edu**
- Empowering Top500 systems for over a decade

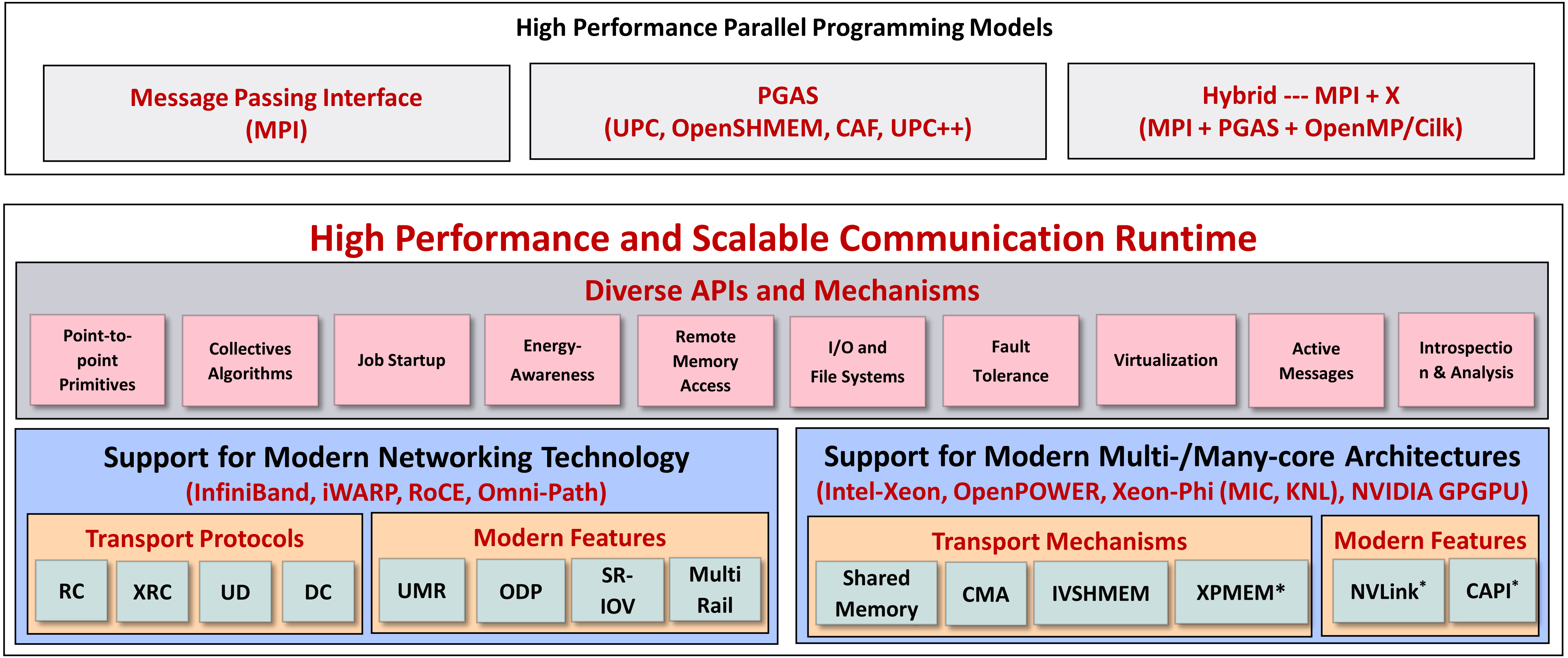
MVAPICH Project Timeline



MVAPICH2 Release Timeline and Downloads



Architecture of MVAPICH2 Software Family



* 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 IB, iWARP, Omni-Path, and RoCE	MVAPICH2
Advanced MPI Features/Support, OSU INAM, PGAS and MPI+PGAS with IB, Omni-Path, and RoCE	MVAPICH2-X
MPI with IB, RoCE & GPU and Support for Deep Learning	MVAPICH2-GDR
HPC Cloud with MPI & IB	MVAPICH2-Virt
Energy-aware MPI with IB, iWARP and RoCE	MVAPICH2-EA
MPI Energy Monitoring Tool	OEMT
InfiniBand Network Analysis and Monitoring	OSU INAM
Microbenchmarks for Measuring MPI and PGAS Performance	OMB

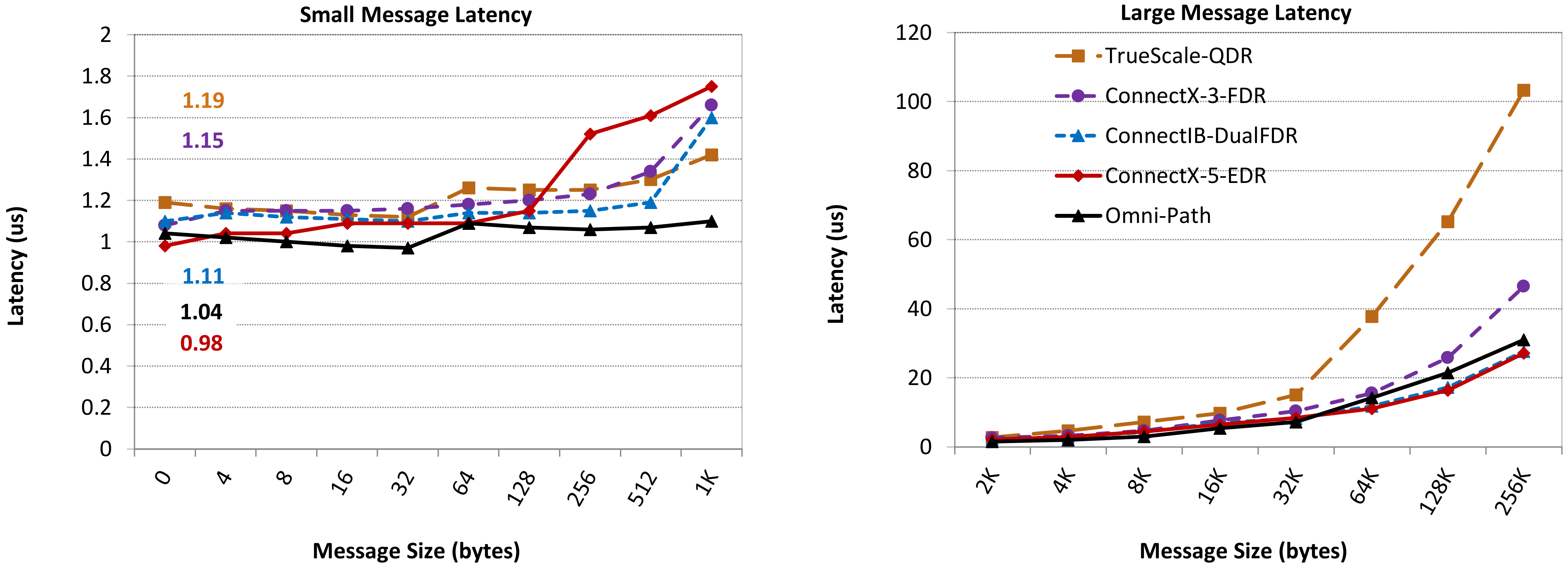
MVAPICH2 2.3-GA

- Released on 07/23/2018
- Major Features and Enhancements
  - Based on MPICH v3.2.1
  - Introduce basic support for executing MPI jobs in Singularity
  - Improve performance for MPI-3 RMA operations
  - **Enhancements for Job Startup**
    - Improved job startup time for OFA-IB-CH3, PSM-CH3, and PSM2-CH3
    - On-demand connection management for PSM-CH3 and PSM2-CH3 channels
    - Enhance PSM-CH3 and PSM2-CH3 job startup to use non-blocking PMI calls
    - Introduce capability to run MPI jobs across multiple InfiniBand subnets
  - **Enhancements to point-to-point operations**
    - Enhance performance of point-to-point operations for CH3-Gen2 (InfiniBand), CH3-PSM, and CH3-PSM2 (Omni-Path) channels
    - Improve performance for Intra- and Inter-node communication for OpenPOWER architecture
    - Enhanced tuning for OpenPOWER, Intel Skylake and Cavium ARM (ThunderX) systems
    - Improve performance for host-based transfers when CUDA is enabled
    - Improve support for large processes per node and hugepages on SMP systems
  - **Enhancements to collective operations**
    - Enhanced performance for Allreduce, Reduce_scatter_block, Allgather, Allgatherv
      - Thanks to Danielle Sikich and Adam Moody @ LLNL for the patch
    - Add support for non-blocking Allreduce using Mellanox SHARP
      - Enhance tuning framework for Allreduce using SHArP
    - Enhanced collective tuning for IBM POWER8, IBM POWER9, Intel Skylake, Intel KNL, Intel Broadwell
  - **Enhancements to process mapping strategies and automatic architecture/network detection**
    - Improve performance of architecture detection on high core-count systems
    - Enhanced architecture detection for OpenPOWER, Intel Skylake and Cavium ARM (ThunderX) systems
    - New environment variable MV2_THREADS_BINDING_POLICY for multi-threaded MPI and MPI+OpenMP applications
    - Support 'spread', 'bunch', 'scatter', 'linear' and 'compact' placement of threads
      - Warn user if oversubscription of core is detected
    - Enhance MV2_SHOW_CPU_BINDING to enable display of CPU bindings on all nodes
    - Added support for MV2_SHOW_CPU_BINDING to display number of OMP threads
    - Added logic to detect heterogeneous CPU/HFI configurations in PSM-CH3 and PSM2-CH3 channels
      - Thanks to Matias Cabral@Intel for the report
    - Enhanced HFI selection logic for systems with multiple Omni-Path HFIs
    - Introduce run time parameter MV2_SHOW_HCA_BINDING to show process to HCA bindings
  - **Miscellaneous enhancements and improved debugging and tools support**
    - Enhance support for MPI_T PVARs and CVARs
    - Enhance debugging support for PSM-CH3 and PSM2-CH3 channels
    - Update to hwloc version 1.11.9
    - Tested with CLANG v5.0.0

Highlights of MVAPICH2 2.3-GA Release

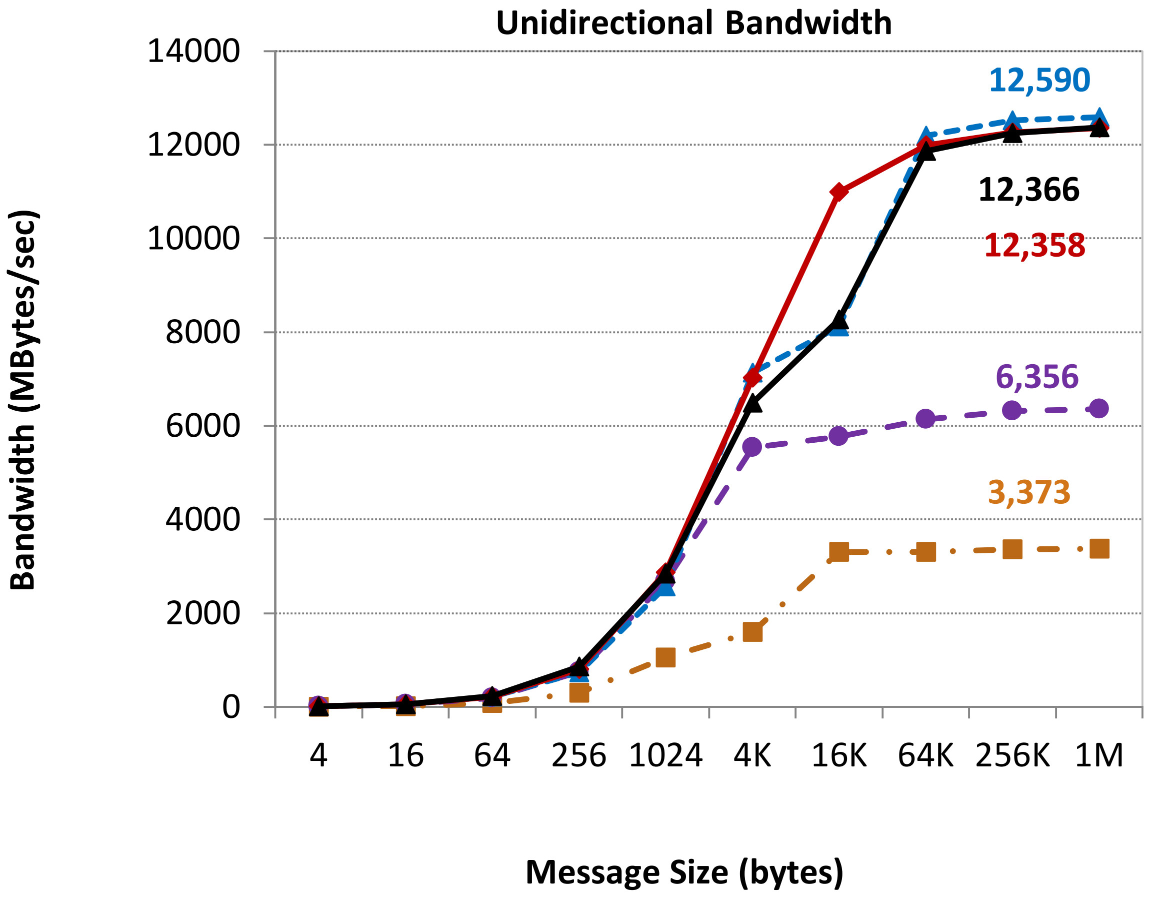
- Support for highly-efficient inter-node and intra-node communication
- Scalable Start-up
- Optimized Collectives using SHArP and Multi-Leaders
- Support for OpenPOWER and ARM architectures
- Performance Engineering with MPI-T
- Application Scalability and Best Practices

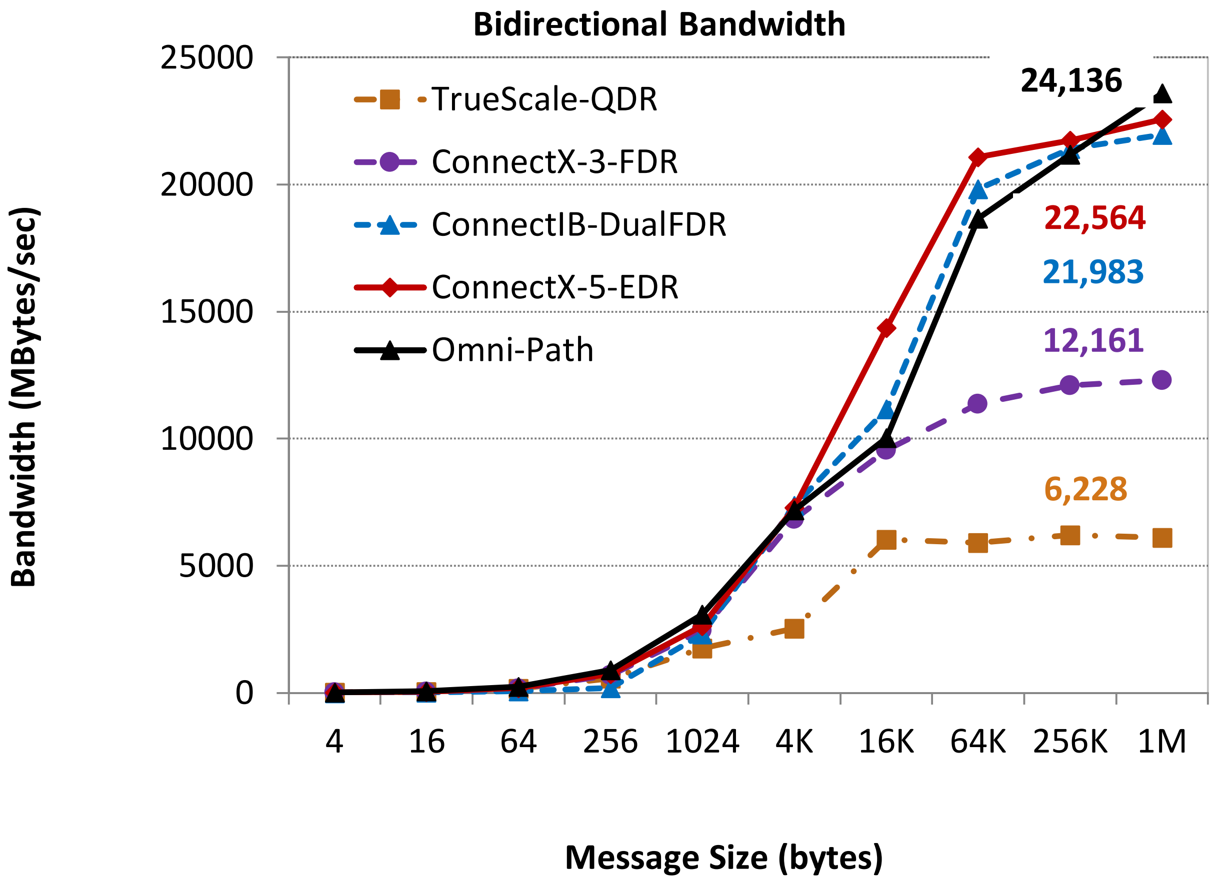
One-way Latency: MPI over IB with MVAPICH2



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-5-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB Switch
Omni-Path - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with Omni-Path switch

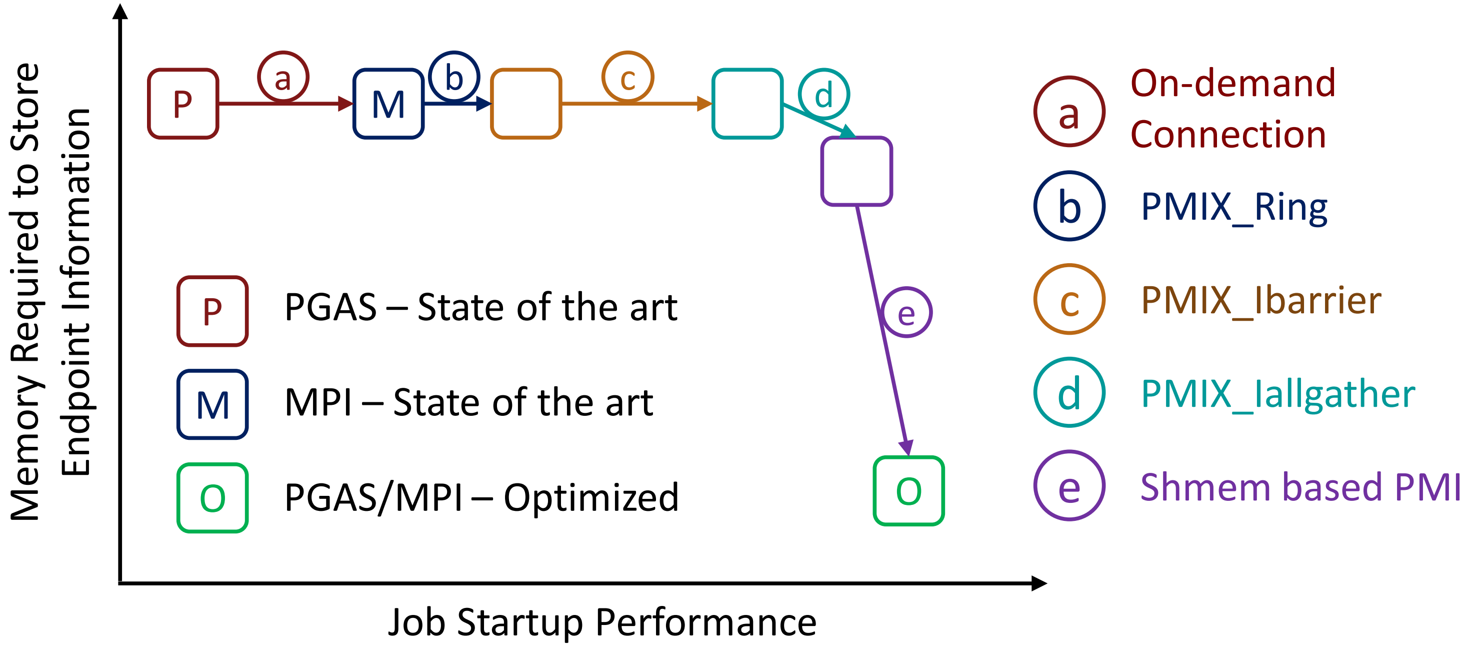
Bandwidth: MPI over IB with MVAPICH2





TrueScale-QDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch
ConnectX-5-EDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 IB switch
Omni-Path - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with Omni-Path switch

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

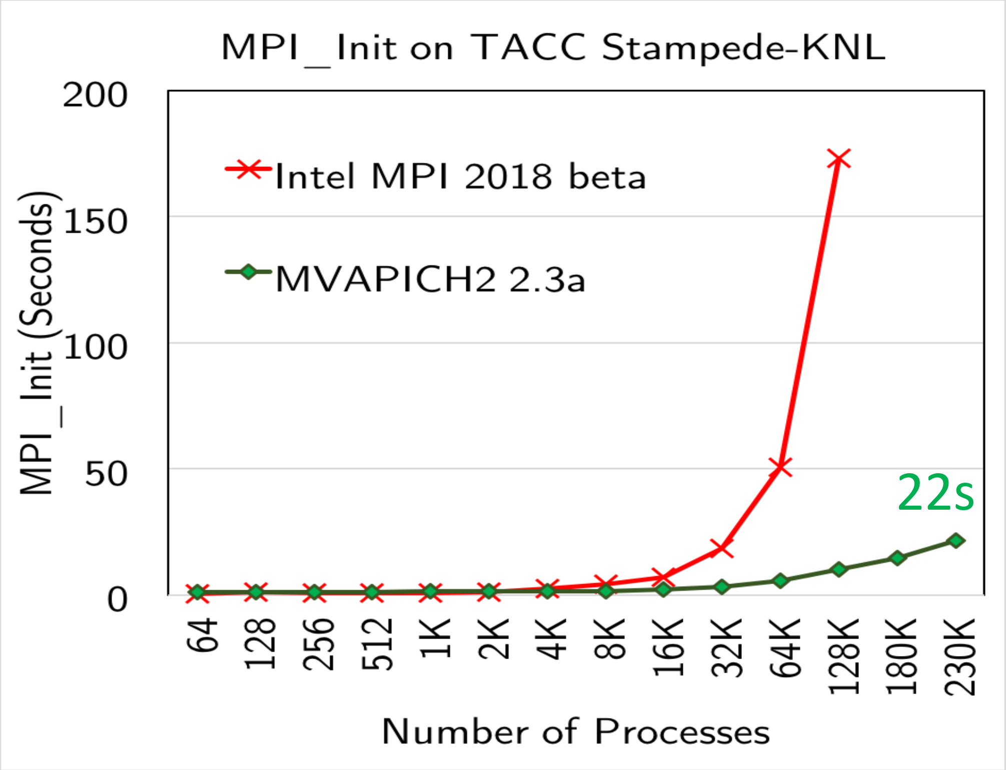
(a) **On-demand Connection Management for OpenSHMEM and OpenSHMEM+MPI.** S. Chakraborty, H. Subramoni, J. Perkins, A. A. Awan, and D K Panda, 20th International Workshop on High-level Parallel Programming Models and Supportive Environments (HIPS ’15)

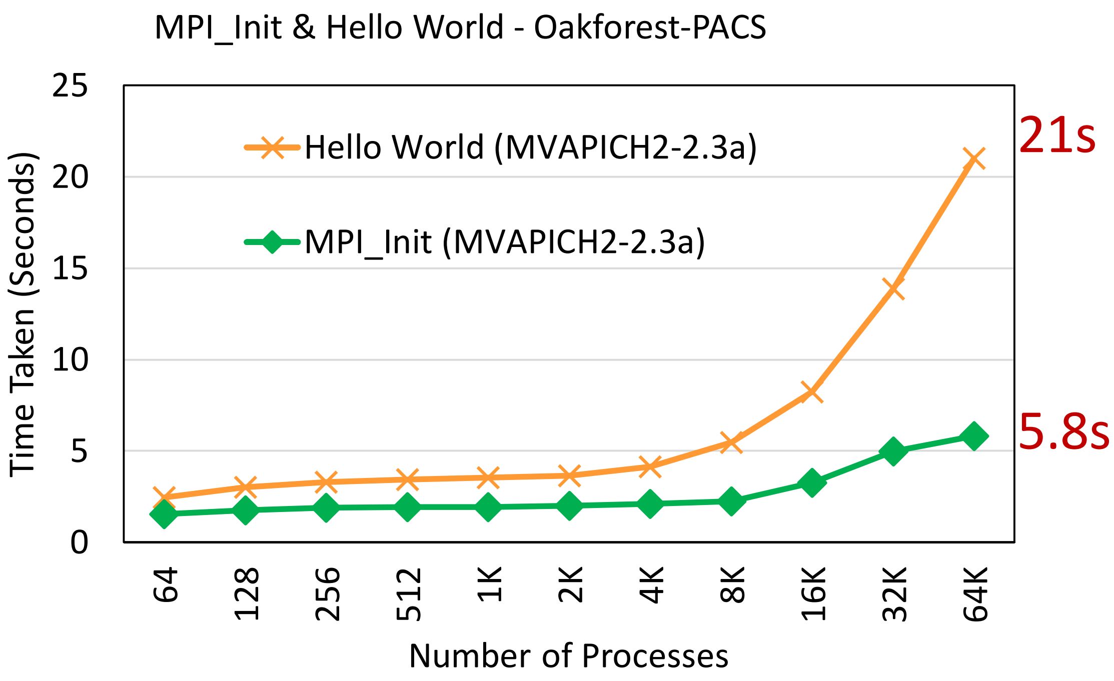
(b) **PMI Extensions for Scalable MPI Startup.** S. Chakraborty, H. Subramoni, A. Moody, J. Perkins, M. Arnold, and D K Panda, Proceedings of the 21st European MPI Users' Group Meeting (EuroMPI/Asia ’14)

(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

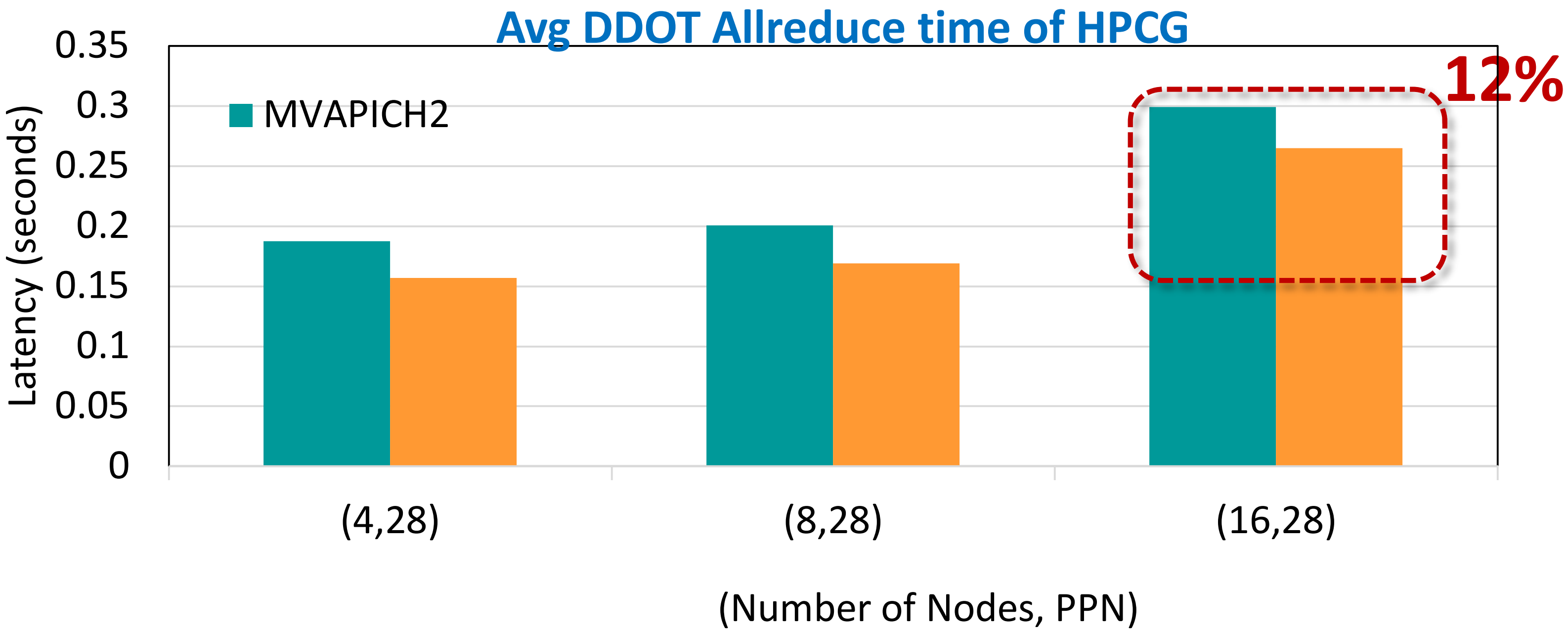


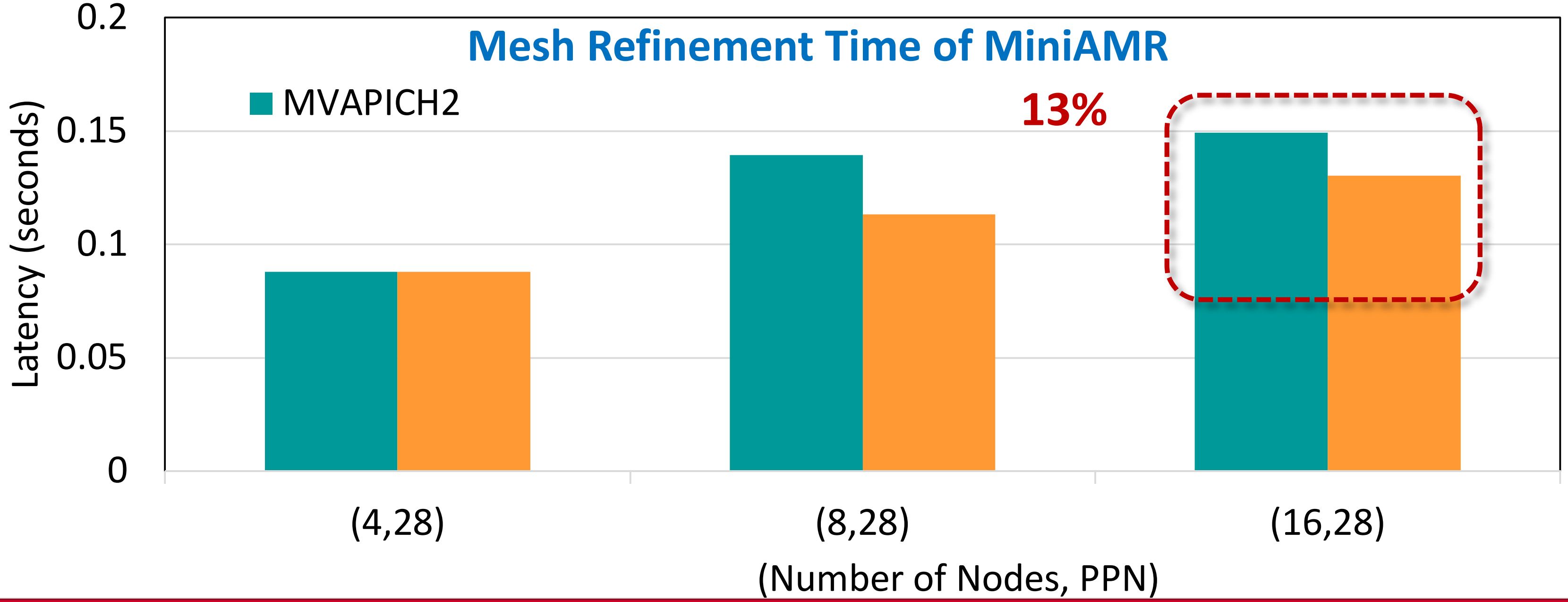


- MPI_Init takes 22 seconds on 229,376 processes on 3,584 KNL nodes (Stampede2 – Full scale)
- 8.8 times faster than Intel MPI at 128K processes (Courtesy: TACC)
- At 64K processes, MPI_Init and Hello World takes 5.8s and 21s respectively (Oakforest-PACS)
- All numbers reported with 64 processes per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

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

Advanced Allreduce Collective Designs Using SHArP



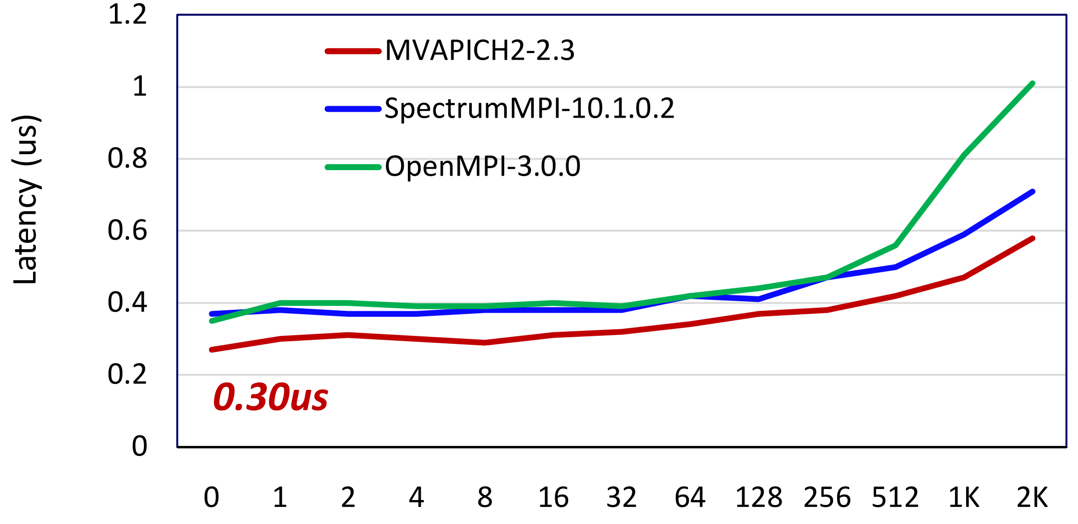


SHArP Support (blocking and non-blocking) is available since MVAPICH2 2.3a

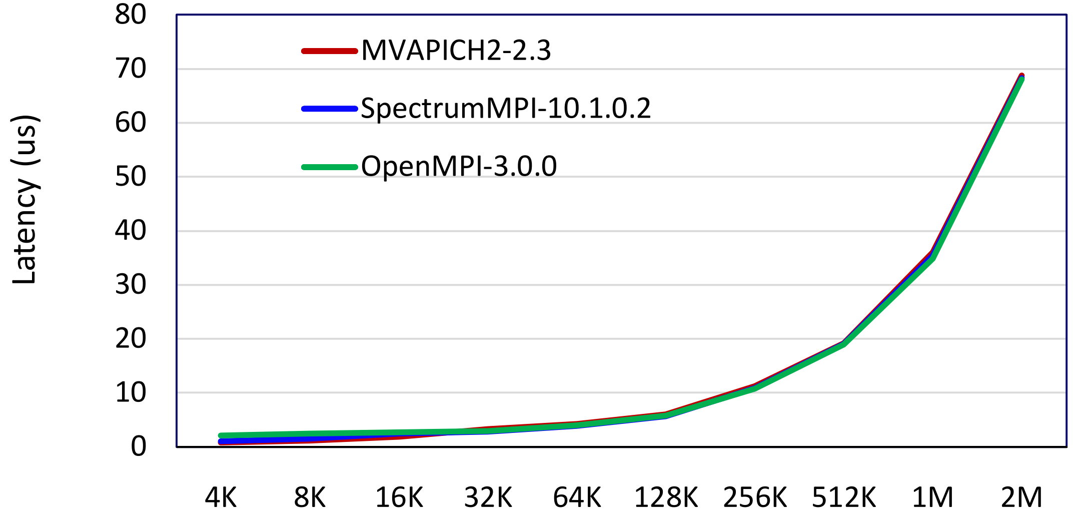
M. Bayatpour, S. Chakraborty, H. Subramoni, X. Lu, and D. K. Panda, Scalable Reduction Collectives with Data Partitioning-based Multi-Leader Design, Supercomputing '17.

Intra-node Point-to-Point Performance on OpenPOWER

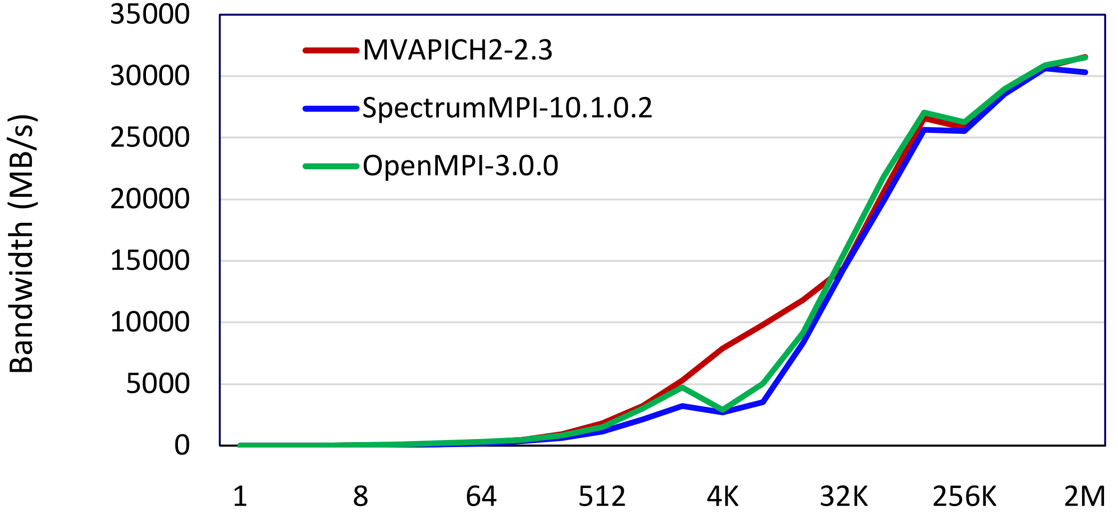
Intra-Socket Small Message Latency



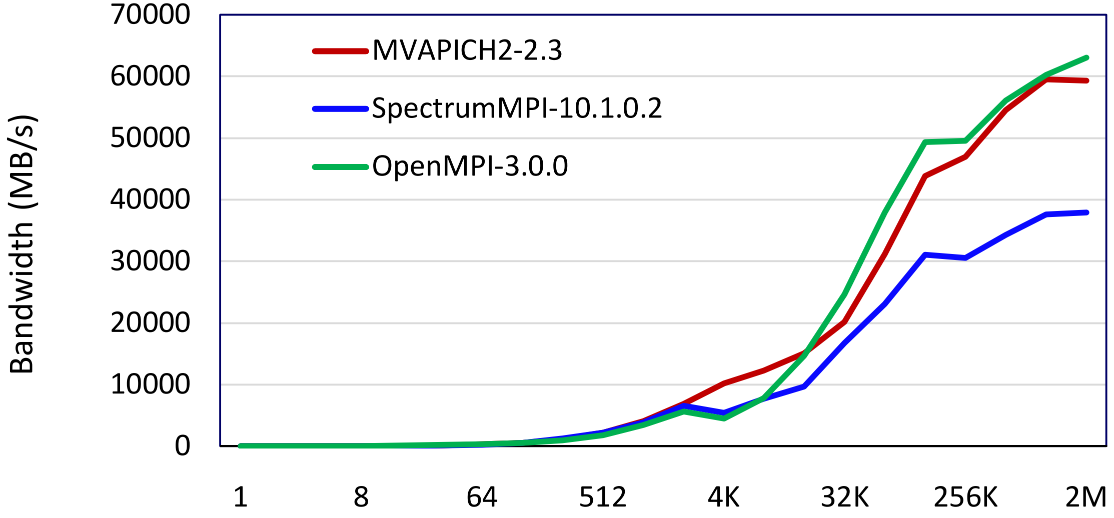
Intra-Socket Large Message Latency



Intra-Socket Bandwidth

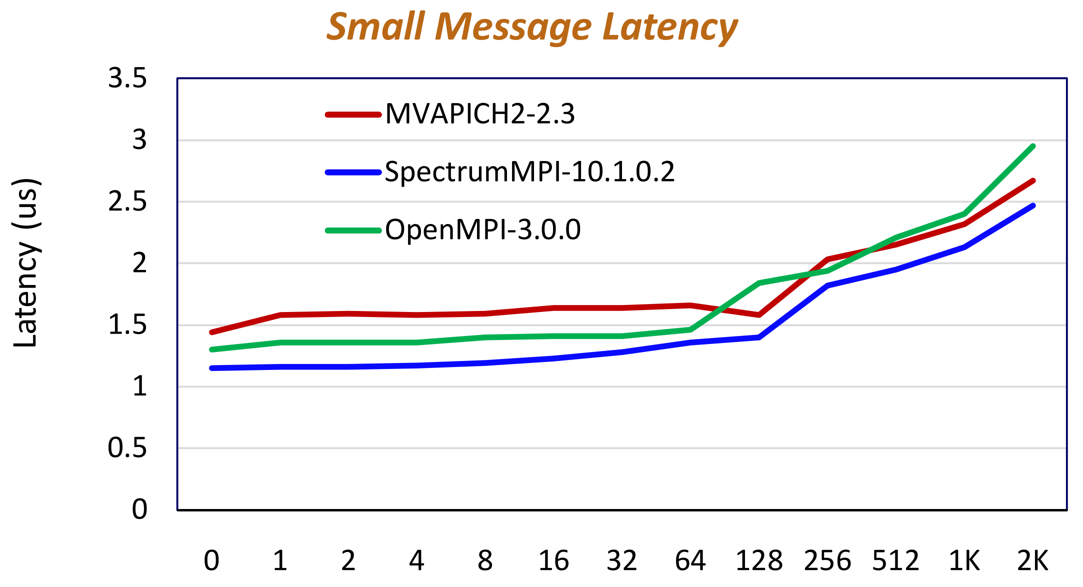


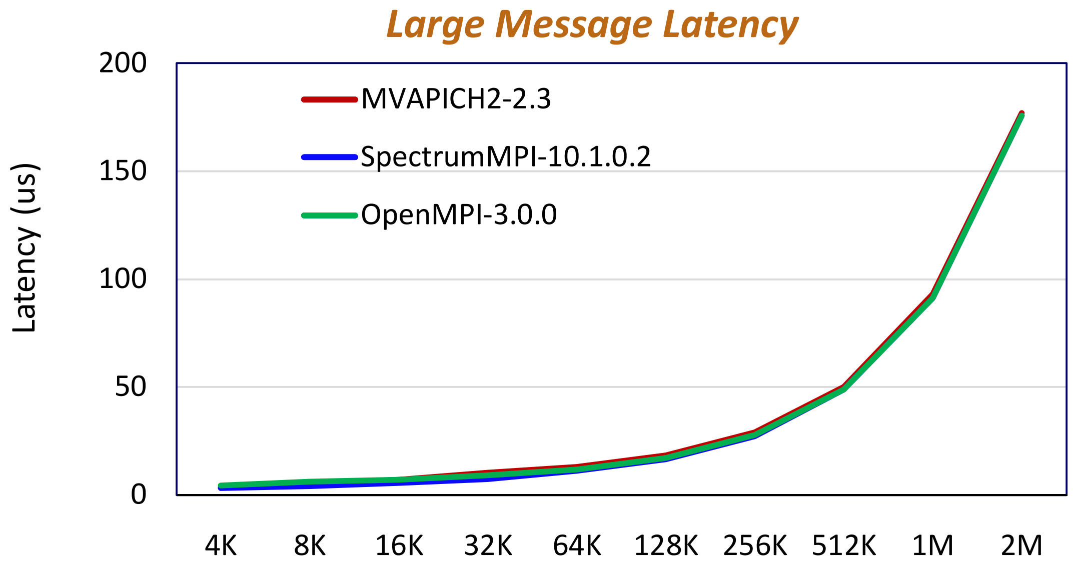
Intra-Socket Bi-directional Bandwidth

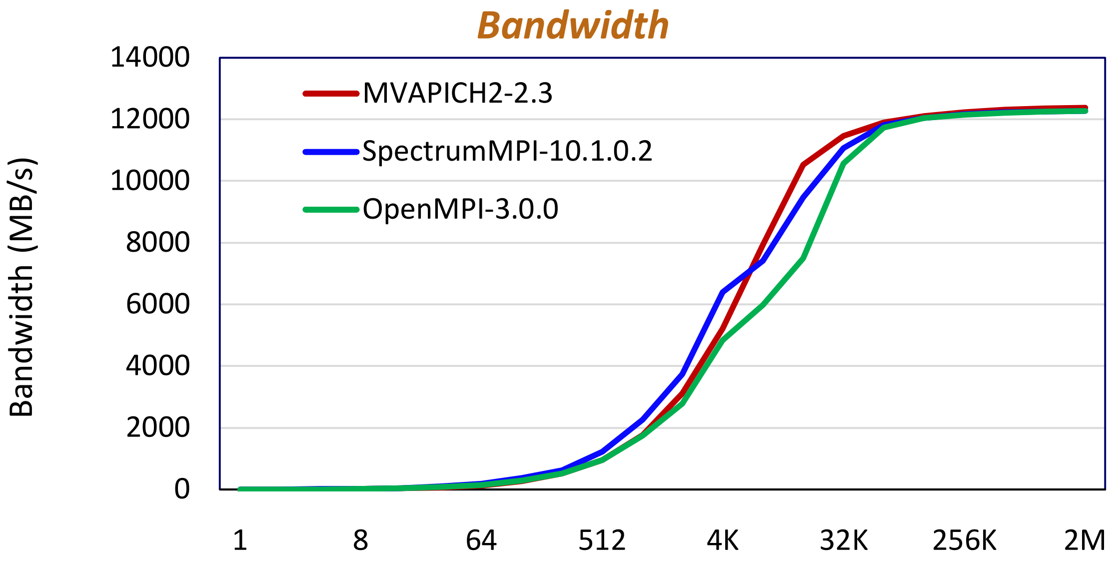


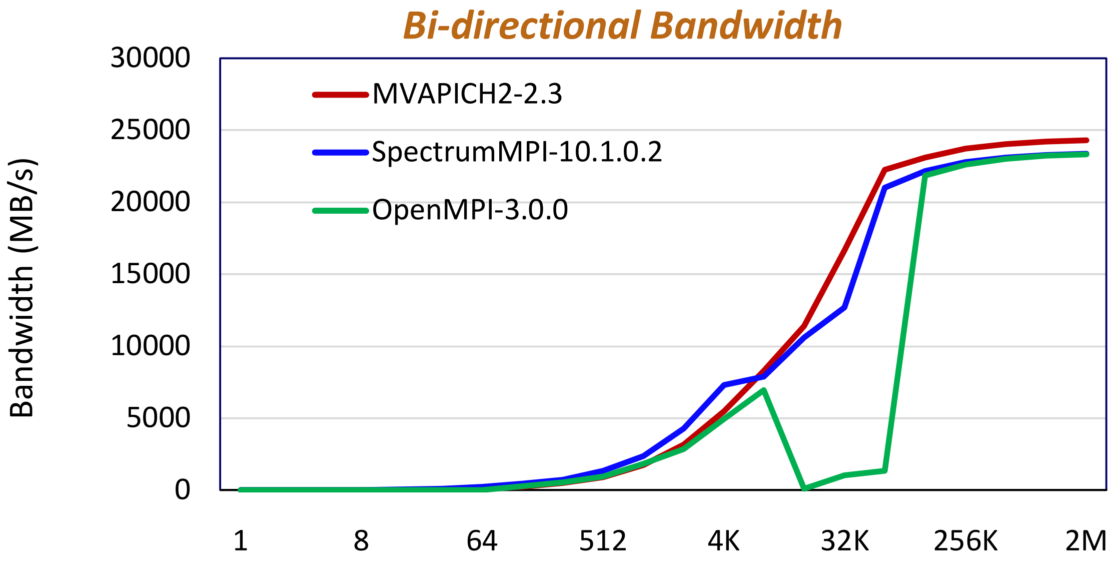
Platform: Two nodes of OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA

Inter-node Point-to-Point Performance on OpenPOWER



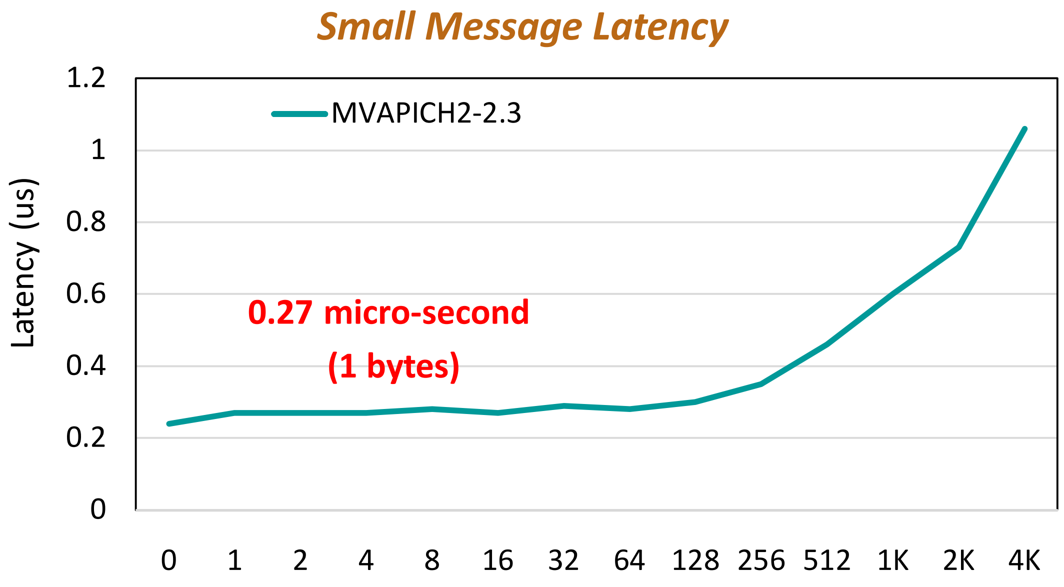


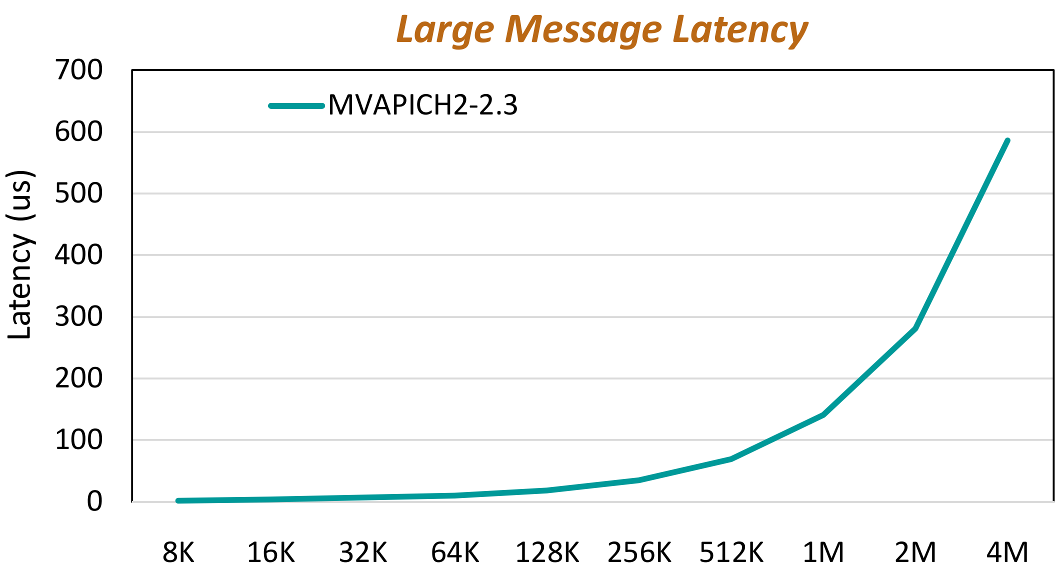


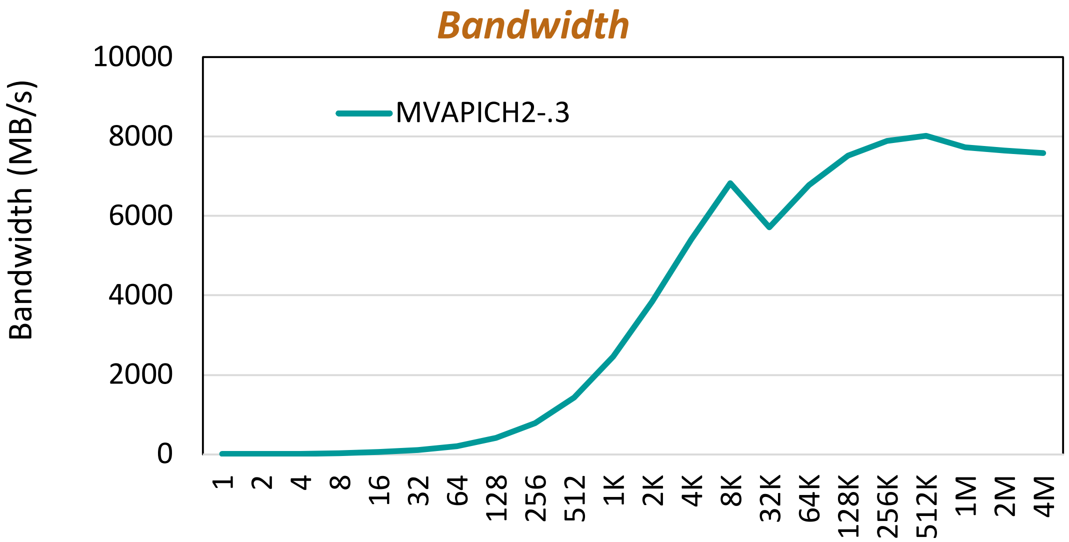


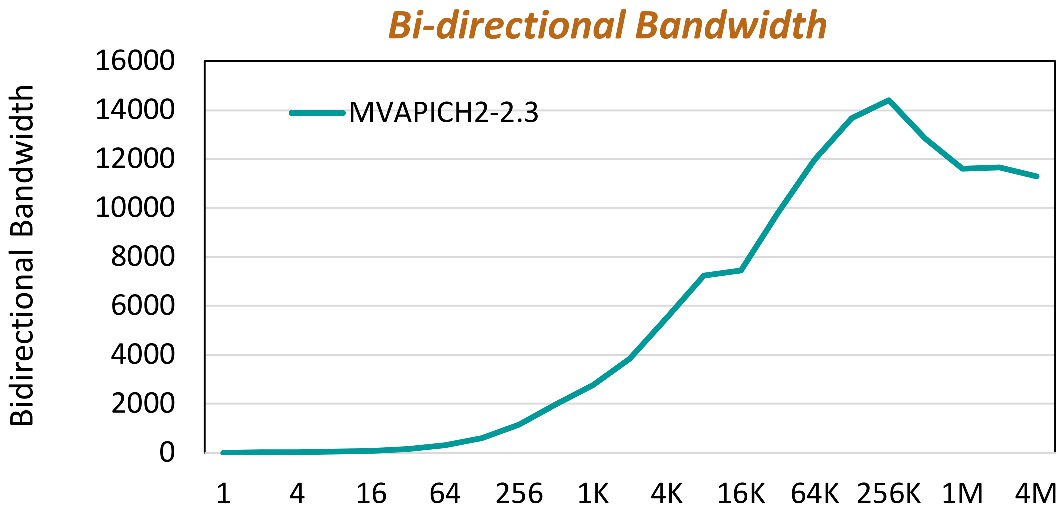
Platform: Two nodes of OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA

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





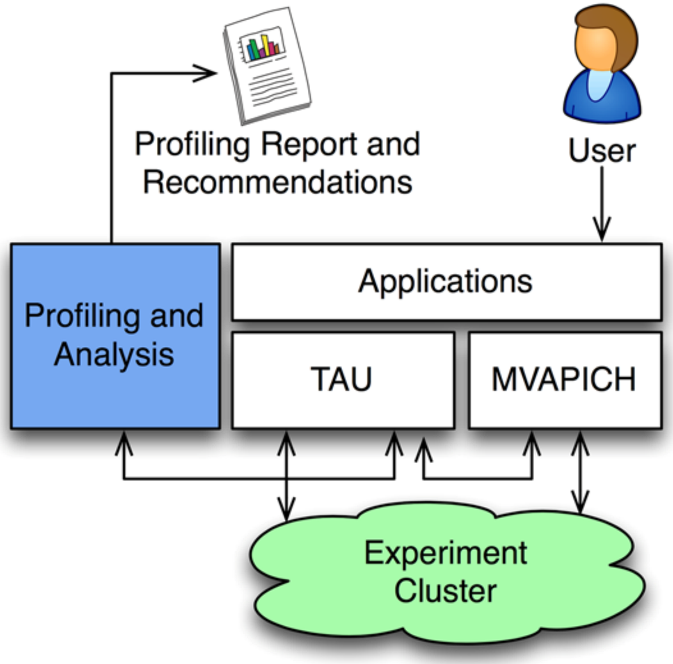




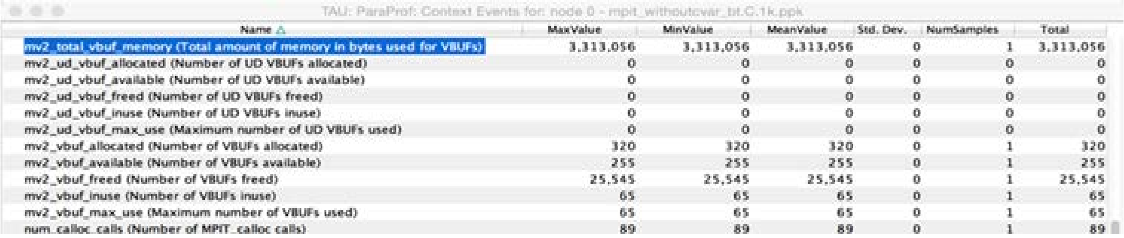
Platform: ARM Cortex A72 (aarch64) MIPS processor with 64 cores dual-socket CPU. Each socket contains 32 cores.

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

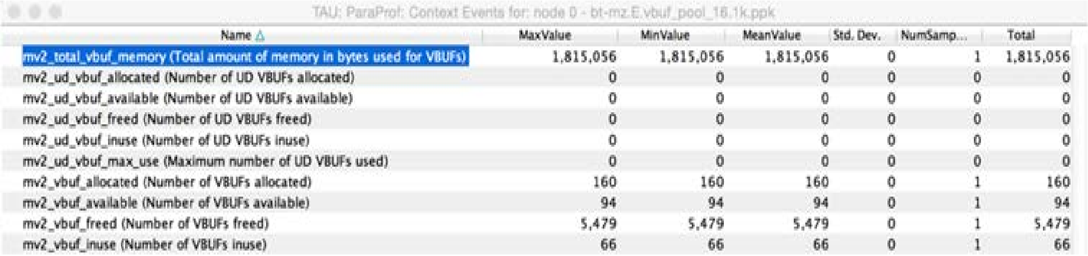


VBUF usage without CVAR based tuning as displayed by ParaProf


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

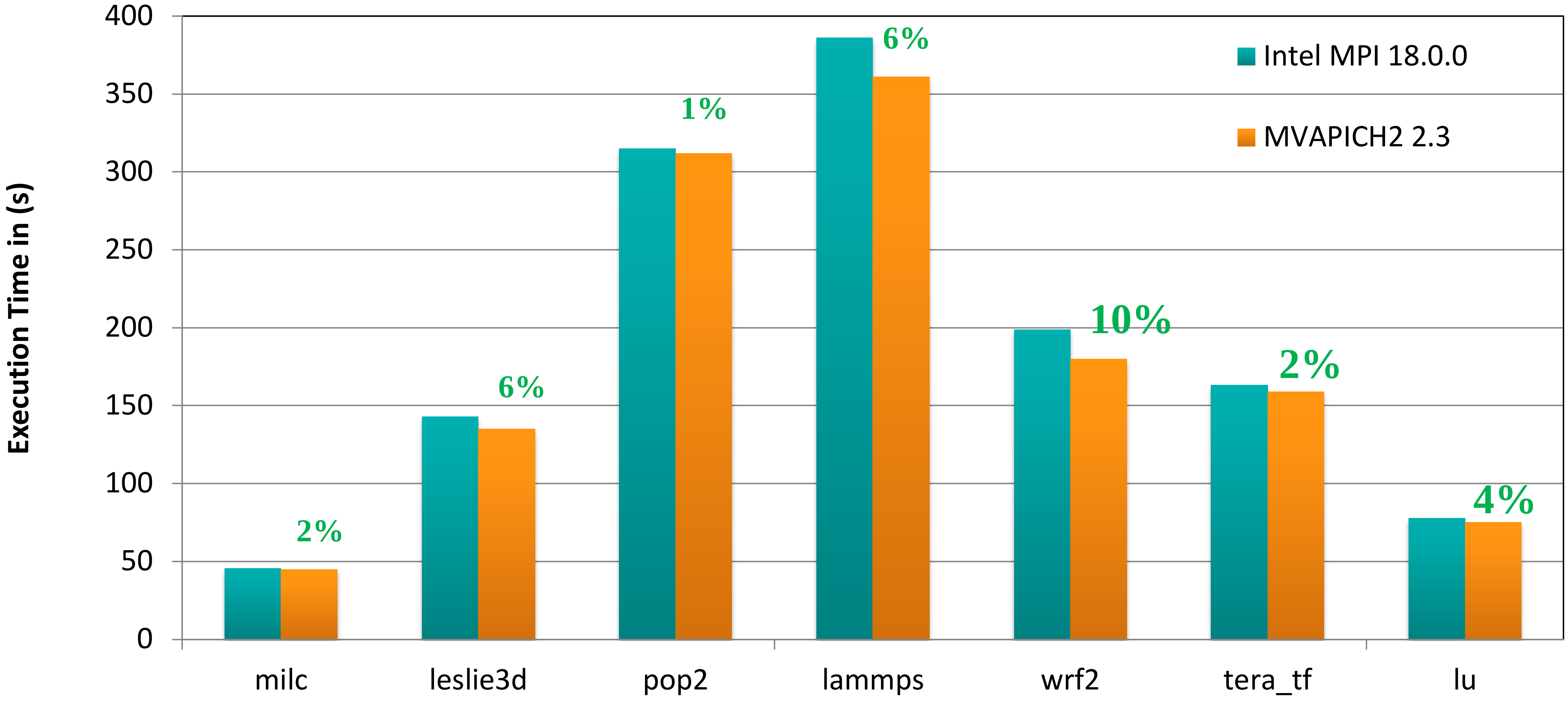
Name	MaxValue	MinValue	MeanValue	Std. Dev.	NumSamples	Total
mv2_total_vbuf_memory (Total amount of memory in bytes used for VBUFs)	3,313,056	3,313,056	3,313,056	0	1	3,313,056
mv2_ud_vbuf_allocated (Number of UD VBUFs allocated)	0	0	0	0	0	0
mv2_ud_vbuf_available (Number of UD VBUFs available)	0	0	0	0	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

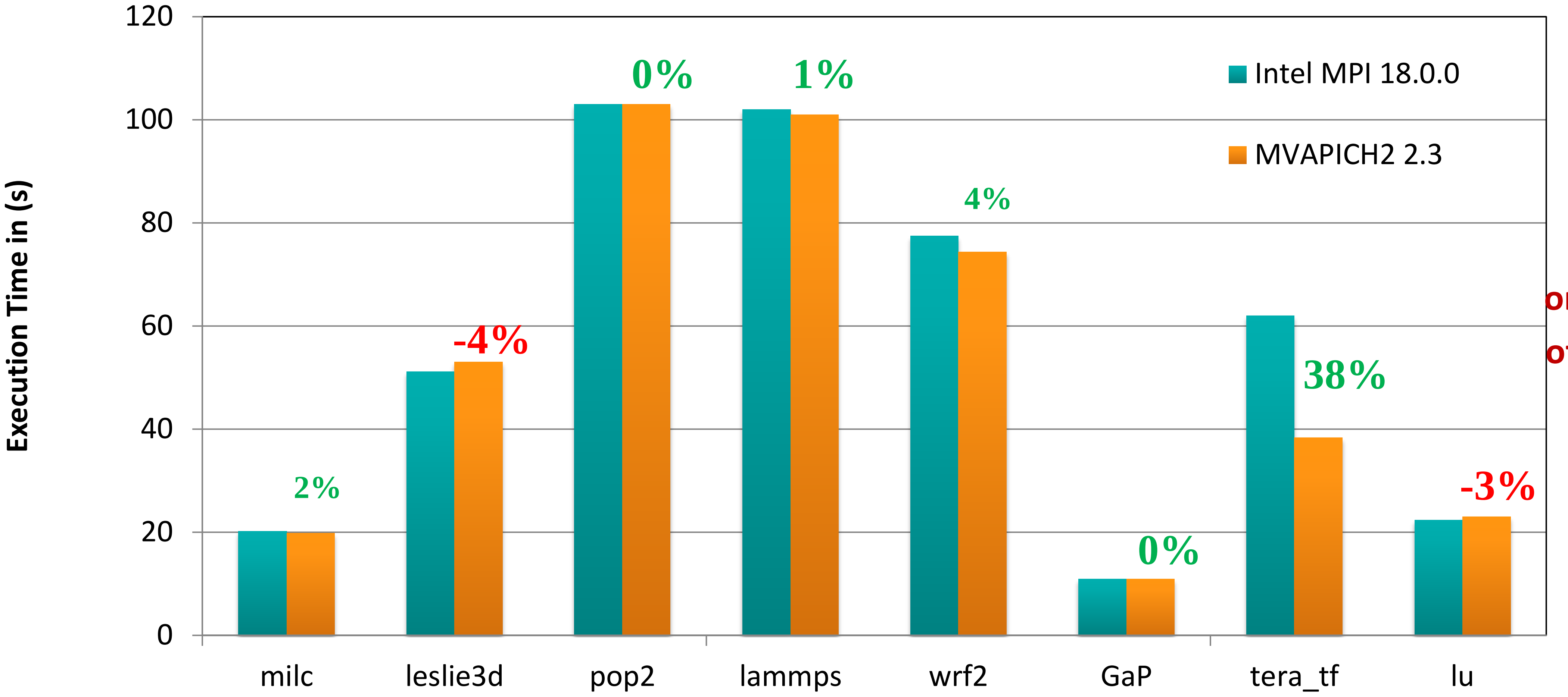
Performance of SPEC MPI 2007 Benchmarks (KNL + Omni-Path)



448 processes on 7 KNL nodes of TACC Stampede2 (64 ppn)

MVAPICH2 outperforms Intel MPI by up to 10%

Performance of SPEC MPI 2007 Benchmarks (Skylake + Omni-Path)

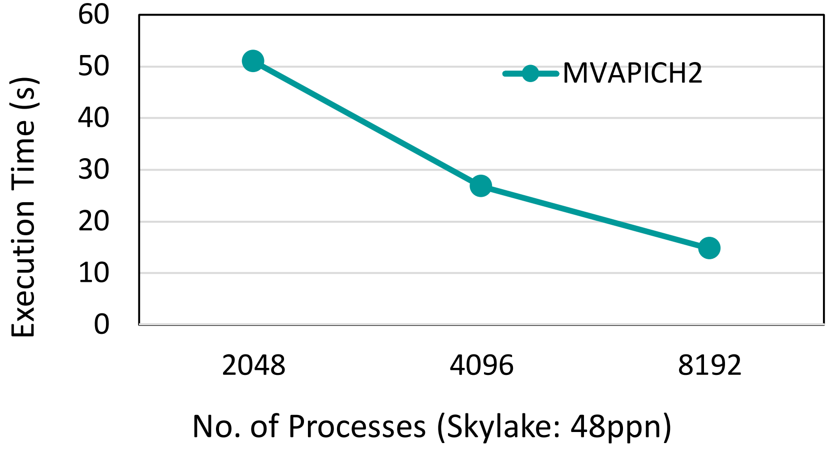


480 processes on 10 Skylake nodes of TACC Stampede2 (48 ppn)

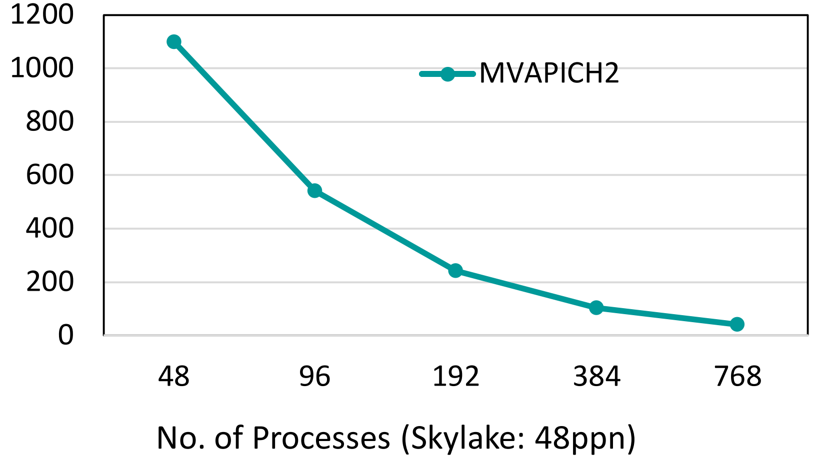
MVAPICH2 outperforms Intel MPI by up to 38%

Application Scalability on Skylake and KNL

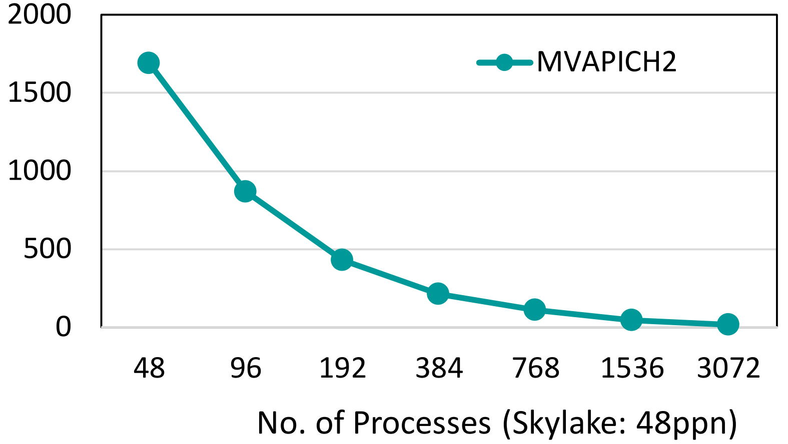
MiniFE (1300x1300x1300 ~ 910 GB)

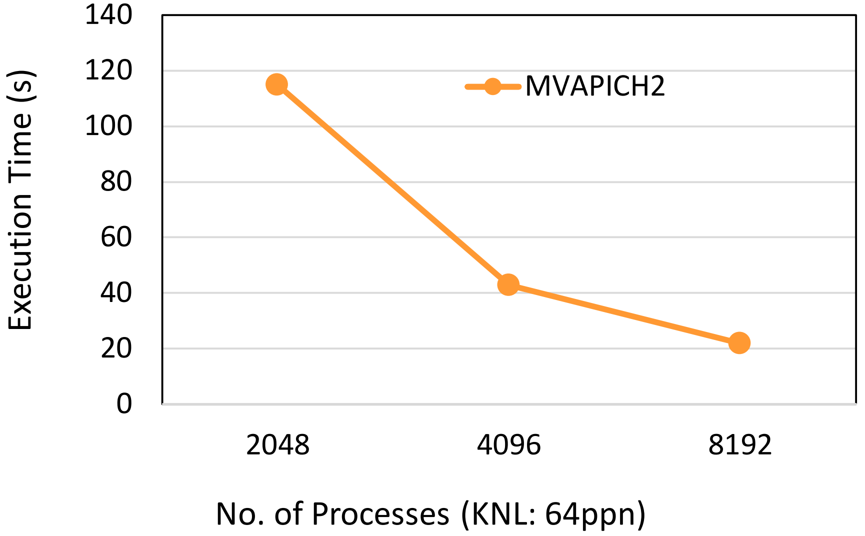


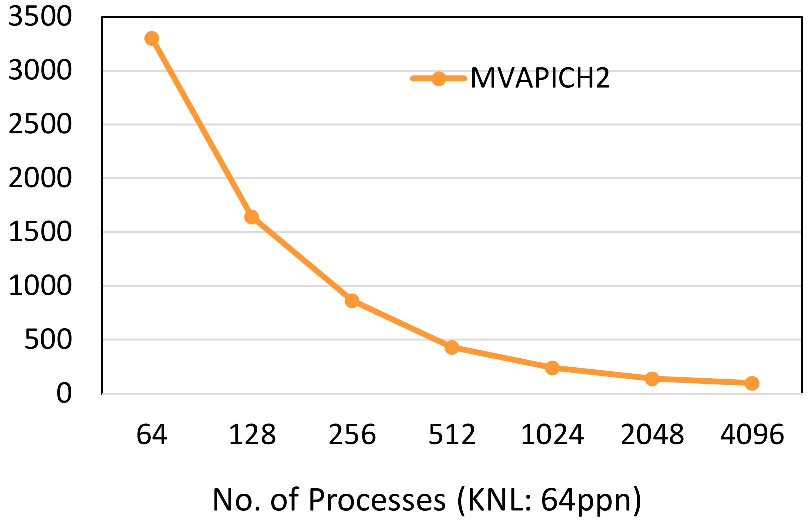
NEURON (YuEtAl2012)

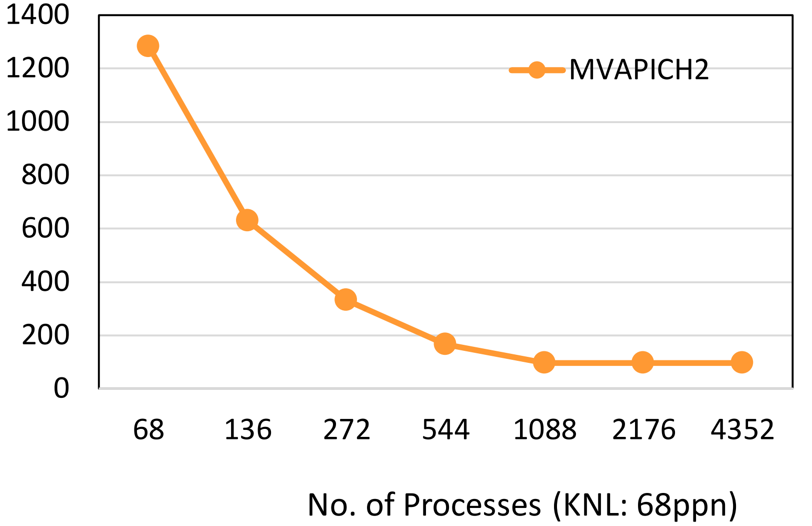


Cloverleaf (bm64) MPI+OpenMP, NUM_OMP_THREADS = 2









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 Upcoming Features

- Dynamic and Adaptive Tag Matching
- Dynamic and Adaptive Communication Protocols
- Support for FPGA-based Accelerators

Dynamic and Adaptive Tag Matching

Challenge

Tag matching is a significant overhead for receivers

Existing Solutions are

- Static and do not adapt dynamically to communication pattern
- Do not consider memory overhead

Solution

A new tag matching design

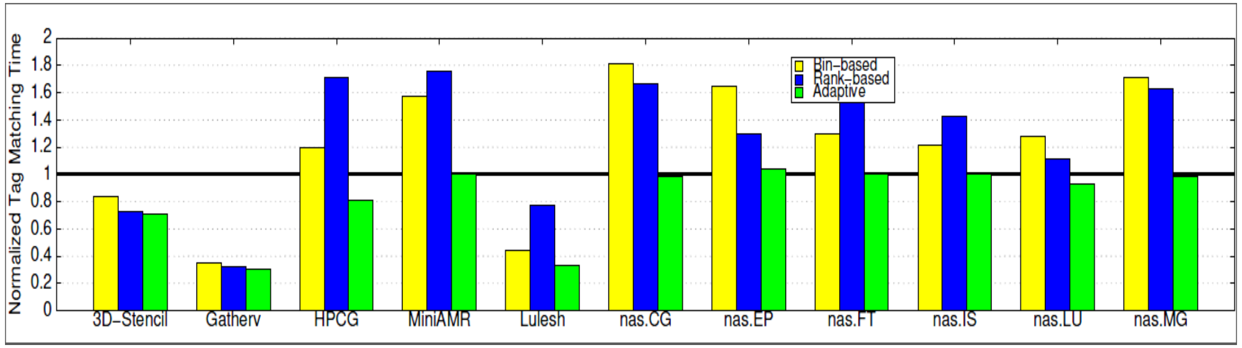
- Dynamically adapt to communication patterns
- Use different strategies for different ranks
- Decisions are based on the number of request object that must be traversed before hitting on the required one

Results

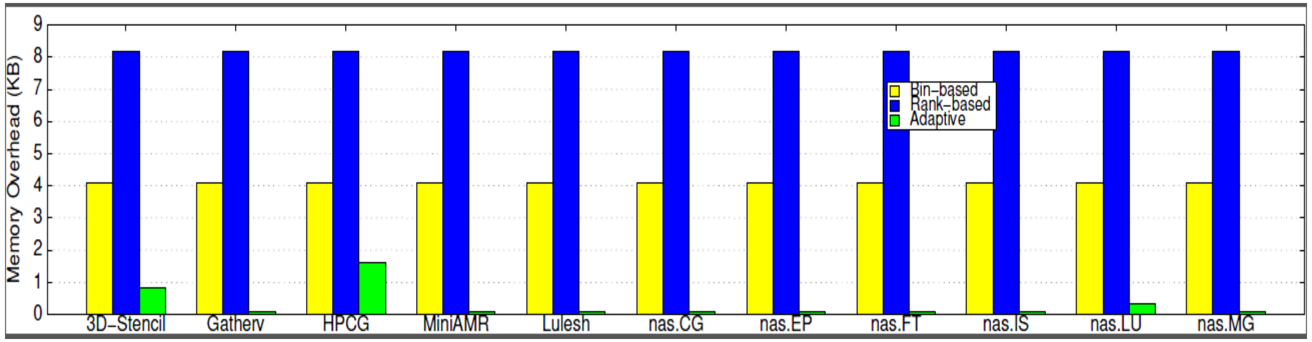
Better performance than other state-of-the art tag-matching schemes

Minimum memory consumption

Will be available in future MVAPICH2 releases



Normalized Total Tag Matching Time at 512 Processes Normalized to Default (Lower is Better)



Normalized Memory Overhead per Process at 512 Processes Compared to Default (Lower is Better)

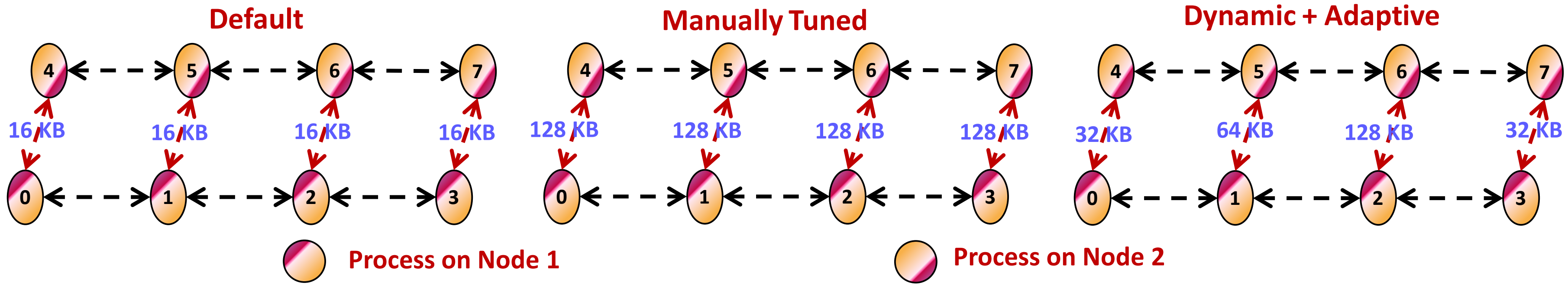
Adaptive and Dynamic Design for MPI Tag Matching; M. Bayatpour, H. Subramoni, S. Chakraborty, and D. K. Panda; IEEE Cluster 2016. [Best Paper Nominee]

Dynamic and Adaptive MPI Point-to-point Communication Protocols

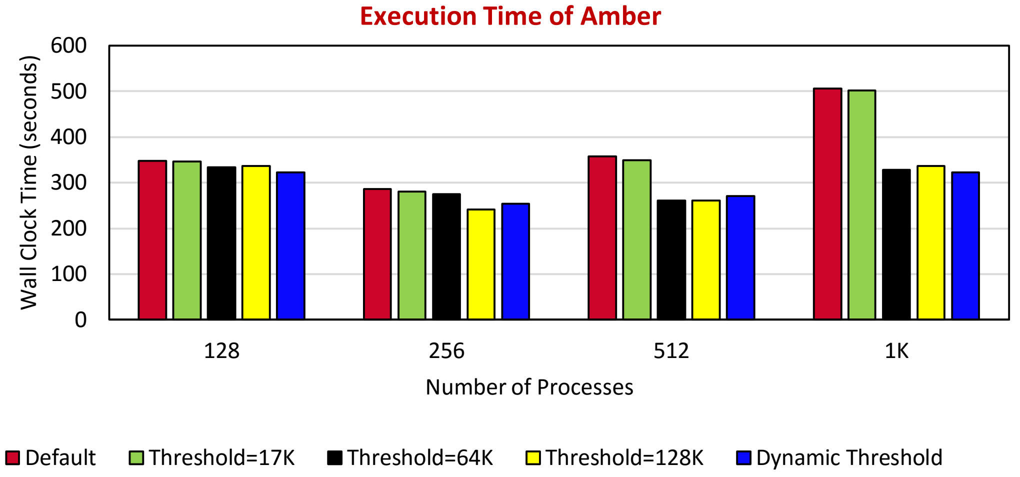
Desired Eager Threshold

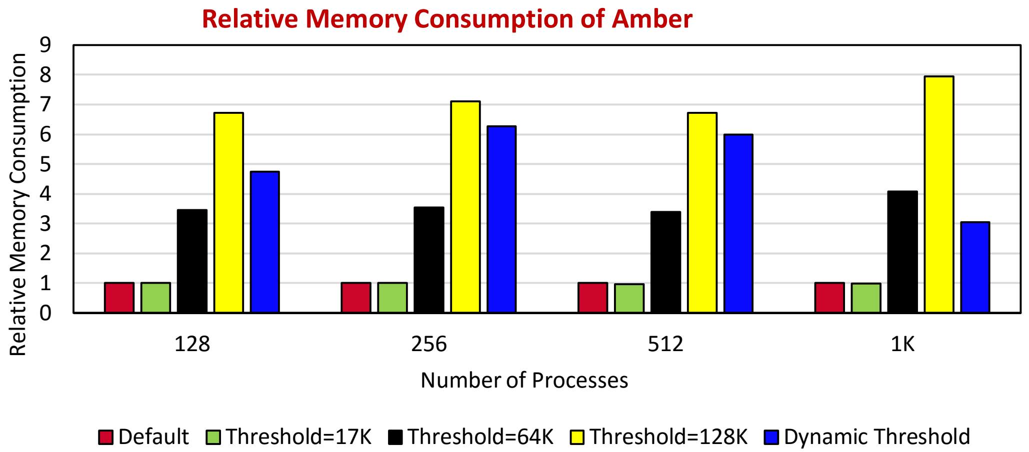
Process Pair	Eager Threshold (KB)
0 – 4	32
1 – 5	64
2 – 6	128
3 – 7	32

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

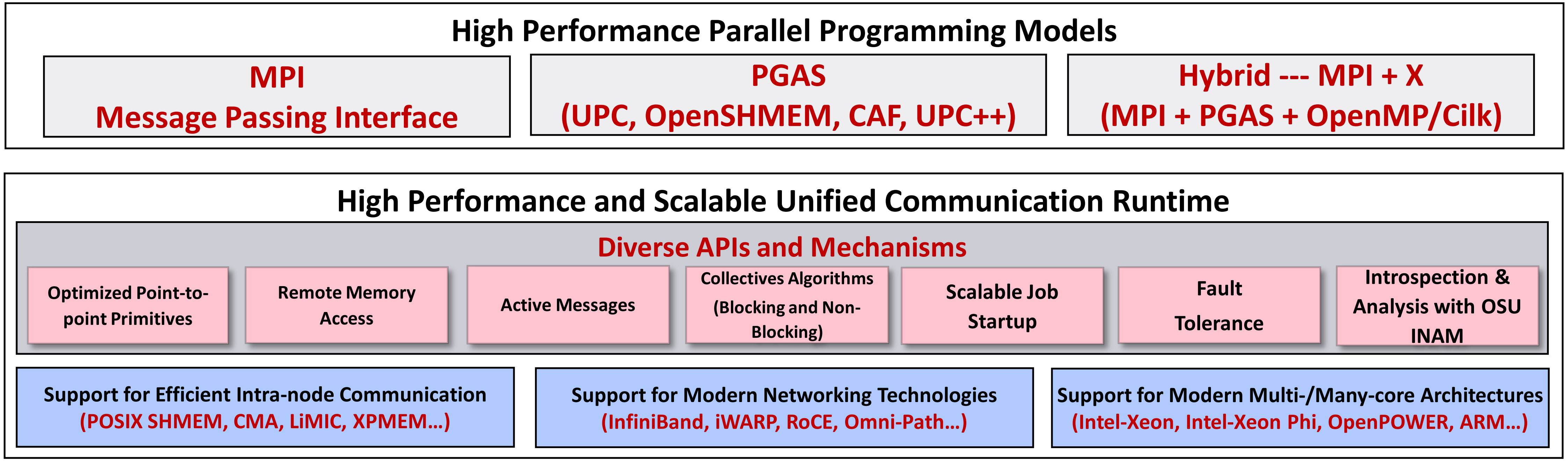
Support for FPGA-Based Accelerators

- FPGAs are emerging in the market for HPC and Deep Learning
- How to exploit FPGA functionalities to accelerate MPI library?
- Funded Collaboration with Pattern Computers

MVAPICH2 Software Family

Requirements	Library
MPI with IB, iWARP, Omni-Path, and RoCE	MVAPICH2
Advanced MPI Features/Support, OSU INAM, PGAS and MPI+PGAS with IB, Omni-Path, and RoCE	MVAPICH2-X
MPI with IB, RoCE & GPU and Support for Deep Learning	MVAPICH2-GDR
HPC Cloud with MPI & IB	MVAPICH2-Virt
Energy-aware MPI with IB, iWARP and RoCE	MVAPICH2-EA
MPI Energy Monitoring Tool	OEMT
InfiniBand Network Analysis and Monitoring	OSU INAM
Microbenchmarks for Measuring MPI and PGAS Performance	OMB

MVAPICH2-X for Hybrid MPI + PGAS Applications

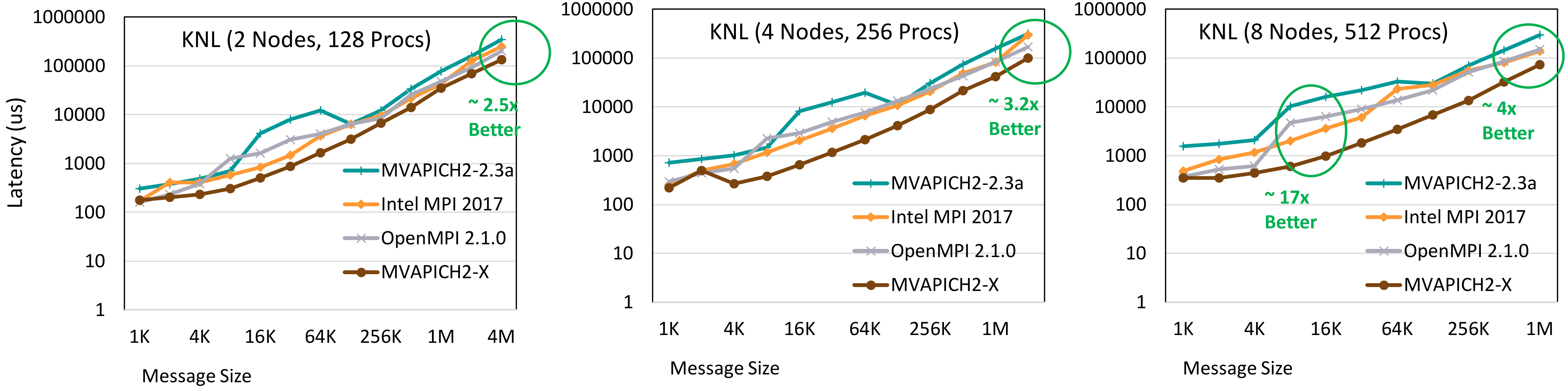


- Current Model – Separate Runtimes for OpenSHMEM/UPC/UPC++/CAF and MPI
  - Possible deadlock if both runtimes are not progressed
  - Consumes more network resource
- Unified communication runtime for MPI, UPC, UPC++, OpenSHMEM, CAF
  - Available with since 2012 (starting with MVAPICH2-X 1.9)
  - http://mvapich.cse.ohio-state.edu

Major Features of MVAPICH2-X

- Advanced MPI Features/Support
  - InfiniBand Features (DC, UMR, ODP, SHArP, and Core-Direct)
  - Kernel-assisted collectives
- Hybrid MPI+PGAS (OpenSHMEM, UPC, UPC++, and CAF)
- Allows to combine programming models (Pure MPI, Pure MPI+OpenMP, MPI+OpenSHMEM, MPI+UPC, MPI+UPC++, ..) in the same program to get best performance and scalability

Optimized CMA-based Collectives for Large Messages



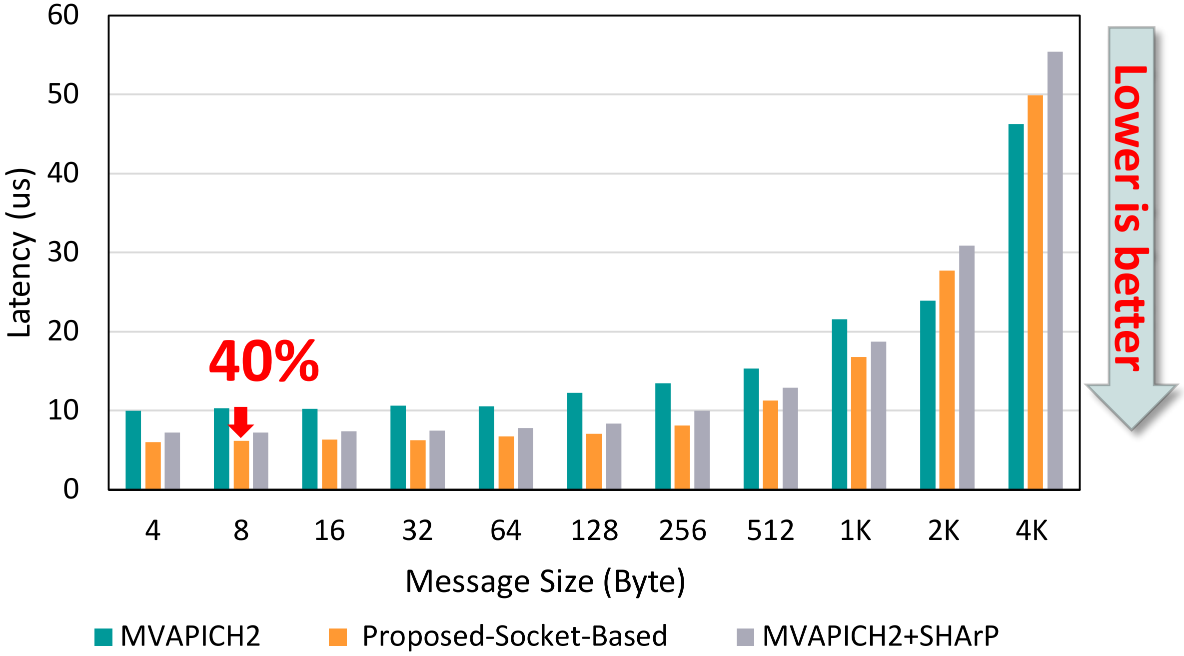
Performance of MPI_Gather on KNL nodes (64PPN)

- 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)

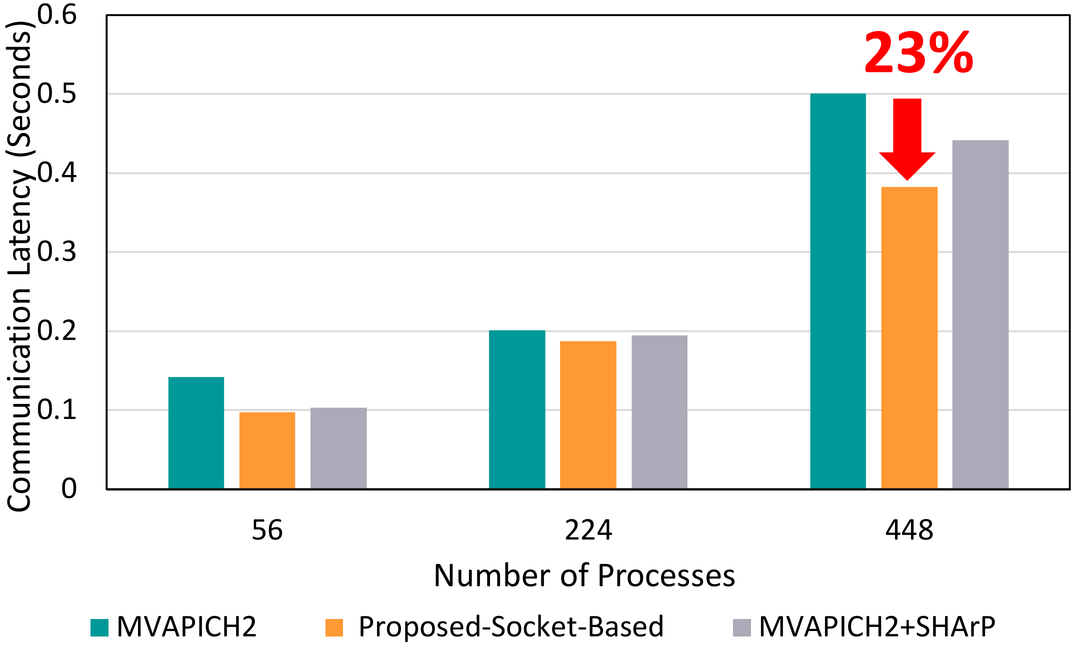
S. Chakraborty, H. Subramoni, and D. K. Panda,* Contention Aware Kernel-Assisted MPI Collectives for Multi/Many-core Systems, *IEEE Cluster '17, BEST Paper Finalist

Available since MVAPICH2-X 2.3b

Advanced Allreduce Collective Designs Using SHArP and Multi-Leaders



OSU Micro Benchmark (16 Nodes, 28 PPN)

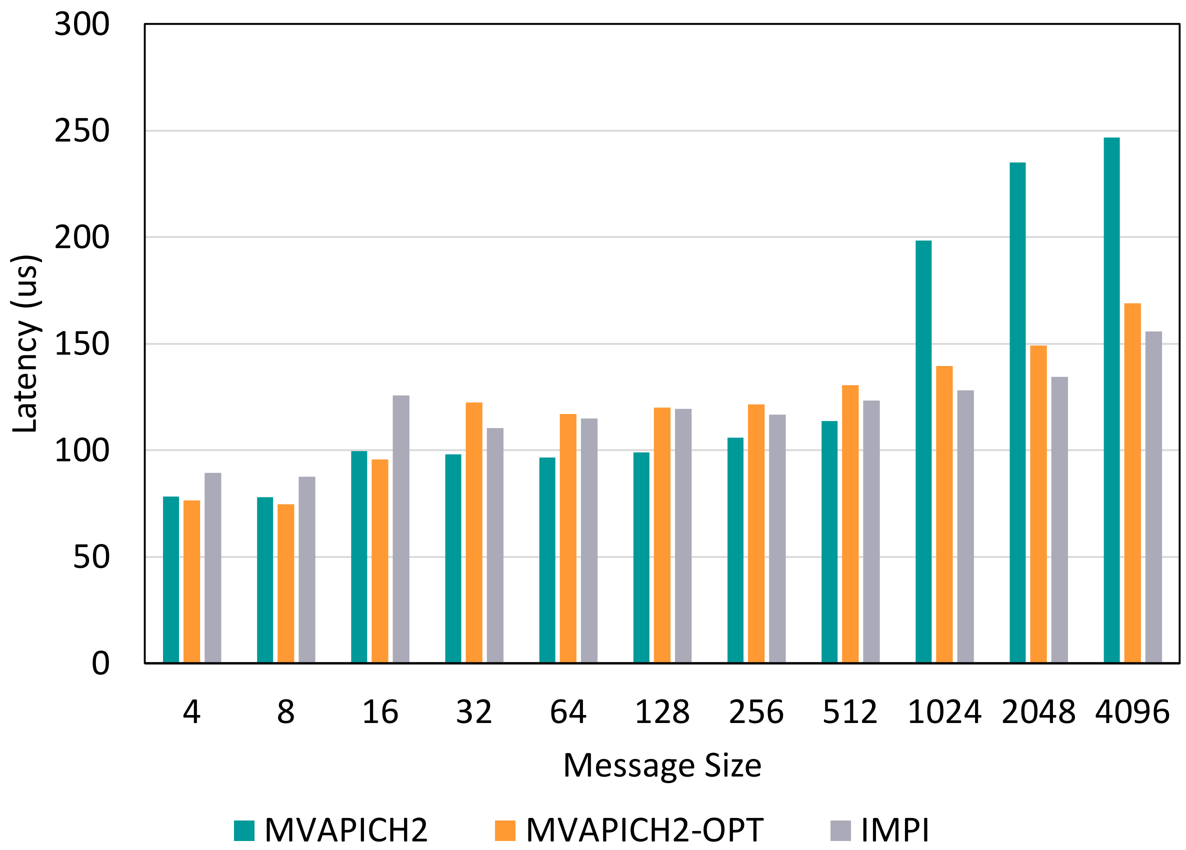


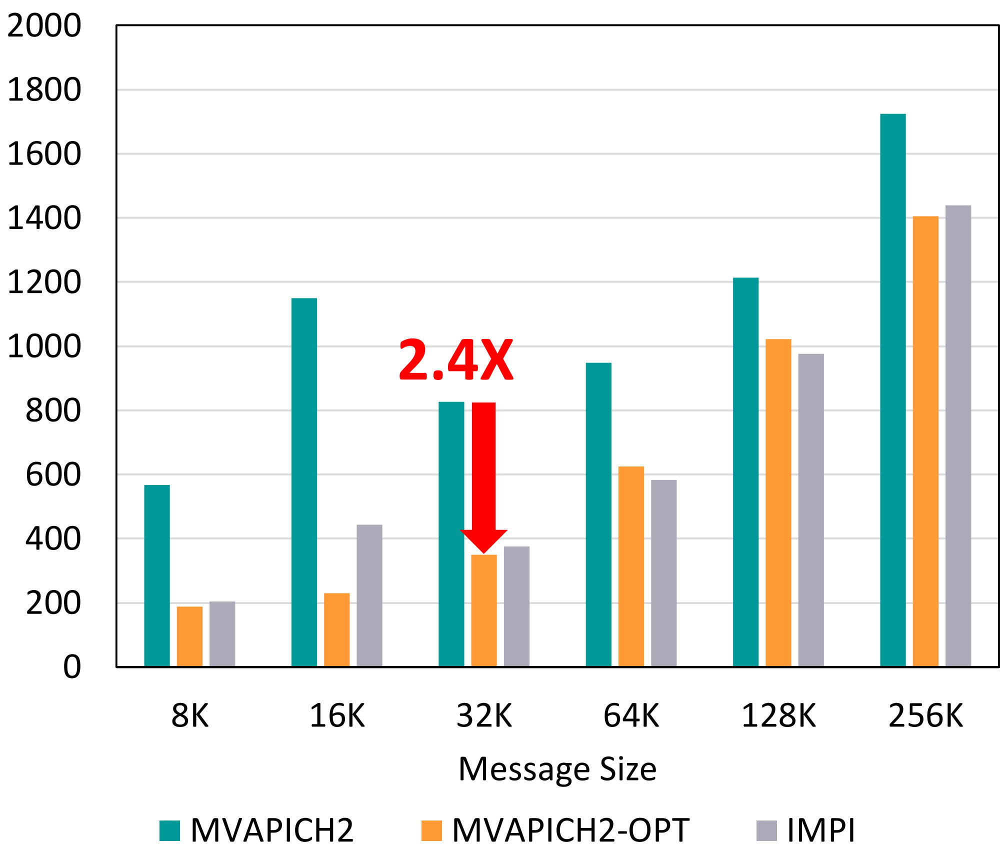
HPCG (28 PPN)

- 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)

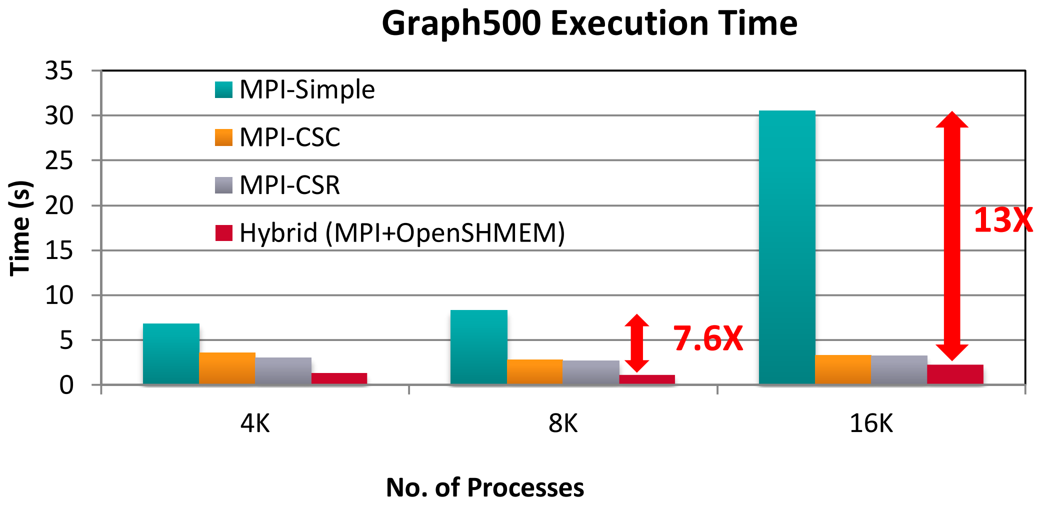


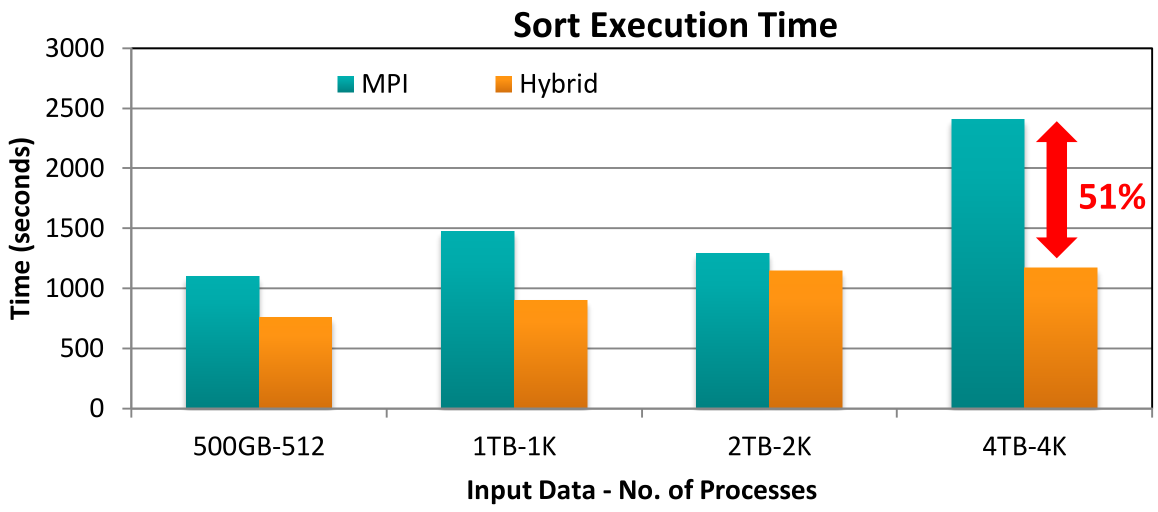


OSU Micro Benchmark 64 PPN

- MPI_Allreduce latency with 32K bytes reduced by 2.4X

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.

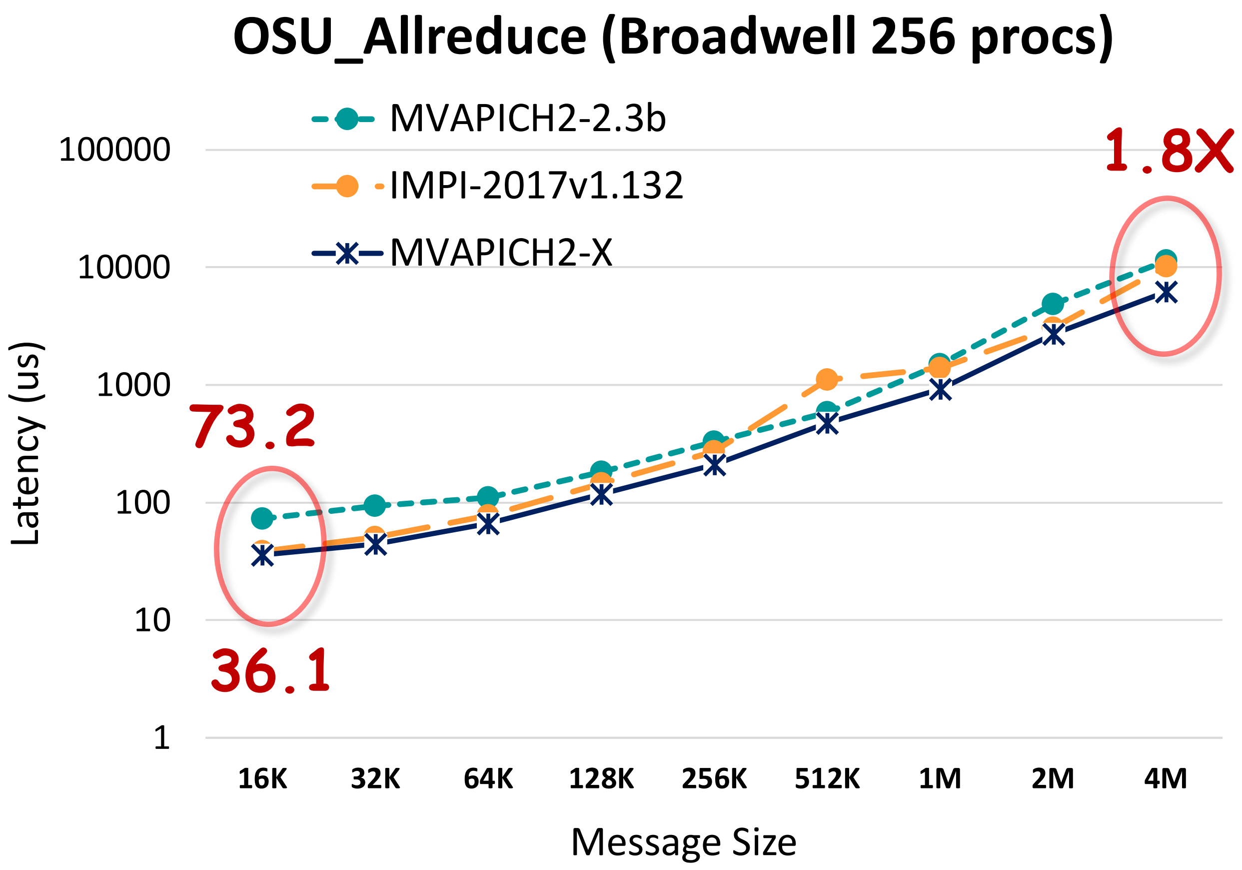
J. Jose, S. Potluri, K. Tomko and D. K. Panda, Designing Scalable Graph500 Benchmark with Hybrid MPI+OpenSHMEM Programming Models, International Supercomputing Conference (ISC'13), June 2013

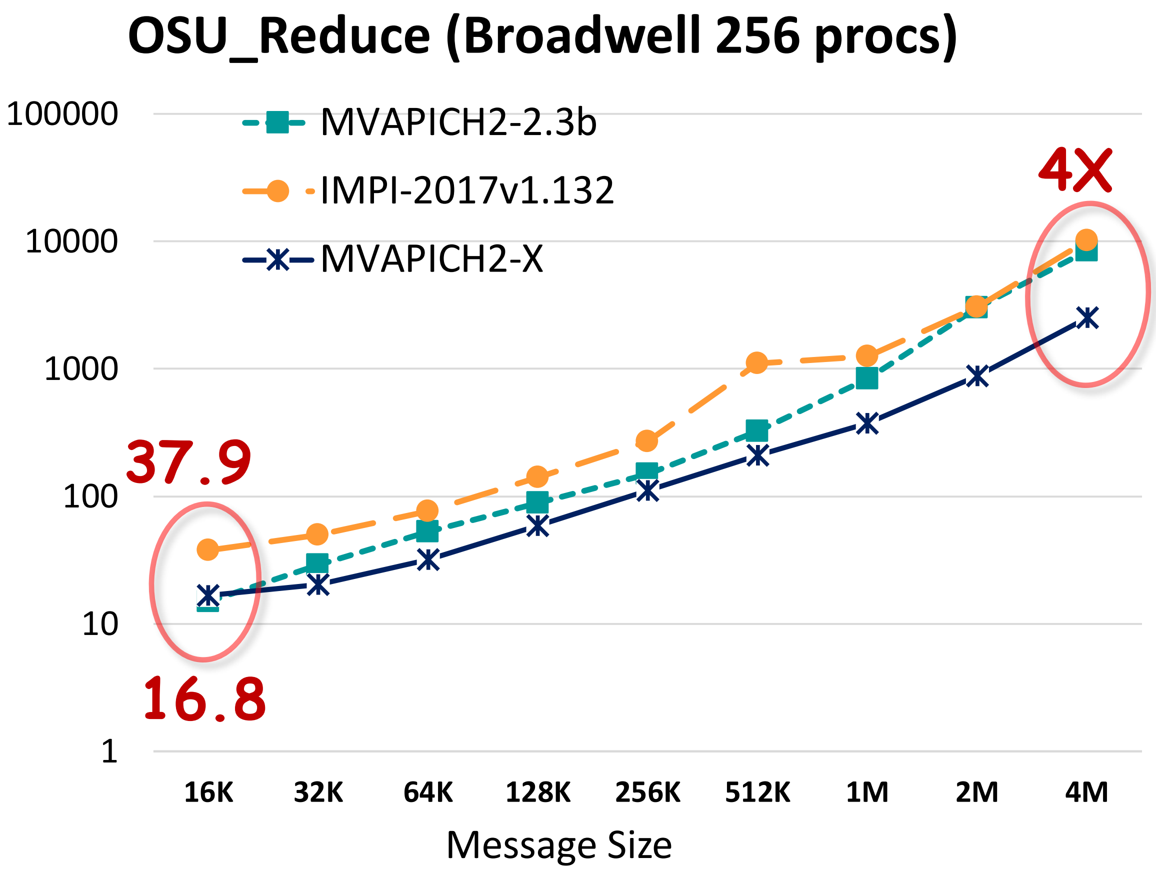
J. Jose, K. Kandalla, M. Luo and D. K. Panda, Supporting Hybrid MPI and OpenSHMEM over InfiniBand: Design and Performance Evaluation, Int'l Conference on Parallel Processing (ICPP '12), September 2012

MVAPICH2-X Upcoming Features

- MVAPICH2-X 2.3RC1 will be released soon with the following features
  - Support for XPMEM (Point-to-point, Reduce, and All-reduce)
  - Efficient Asynchronous Progress with Full Subscription
  - Contention-aware CMA-based collective support for OpenSHMEM, UPC, and UPC++
- Implicit On-Demand Paging (ODP) Support
- Other collectives with XPMEM-based Support

Shared Address Space (XPMEM)-based Collectives Design

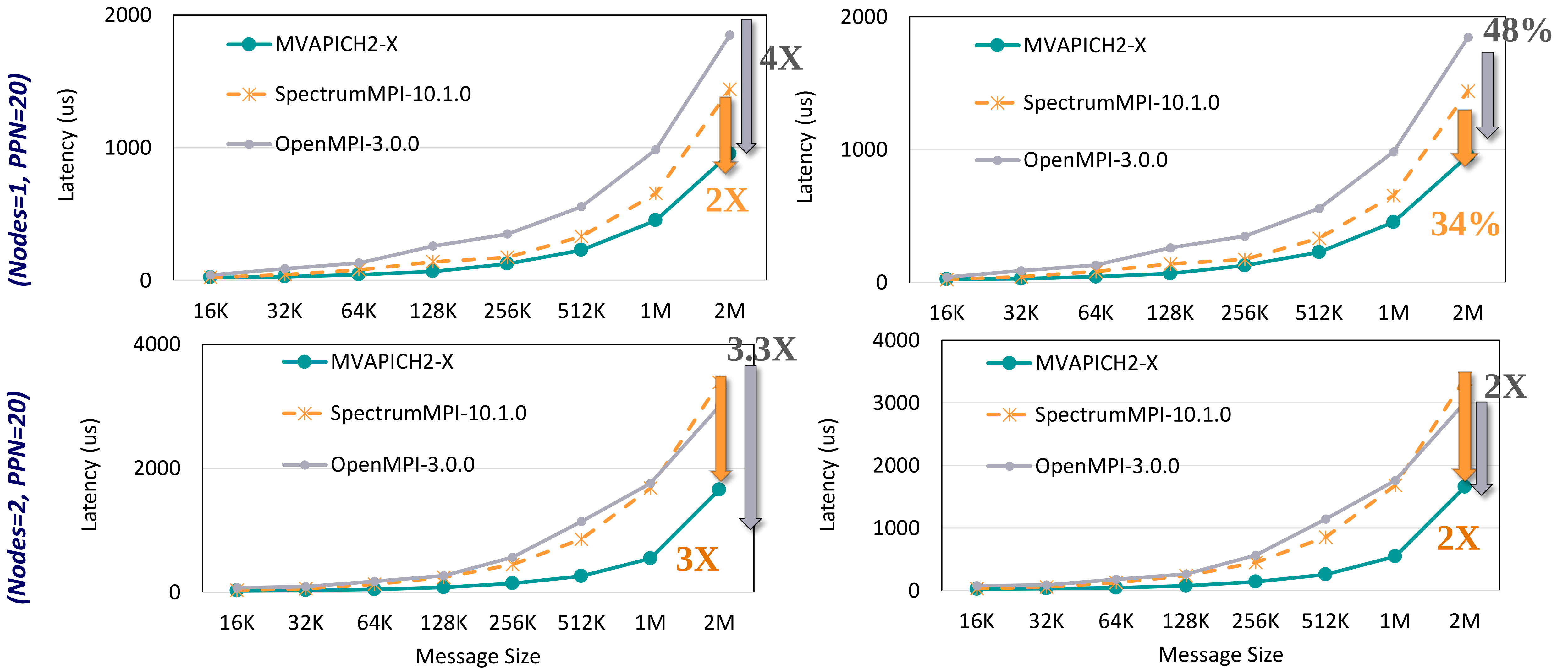




- "*Shared Address Space*"-based true *zero-copy* Reduction collective designs in MVAPICH2
- Offloaded computation/communication to peers ranks in reduction collective operation
- Up to **4X** improvement for 4MB Reduce and up to **1.8X** improvement for 4M AllReduce

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.

Optimized All-Reduce with XPMEM on OpenPOWER



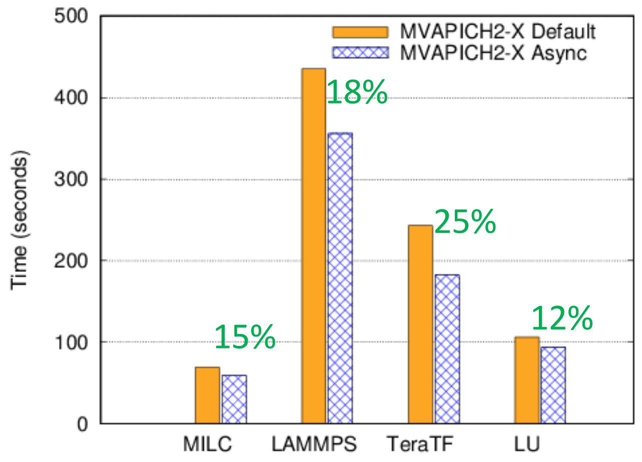
- **Optimized MPI All-Reduce Design in MVAPICH2**
  - *Up to 2X* performance improvement over Spectrum MPI and 4X over OpenMPI for intra-node

Optimized Runtime Parameters: MV2_CPU_BINDING_POLICY=hybrid MV2_HYBRID_BINDING_POLICY=bunch

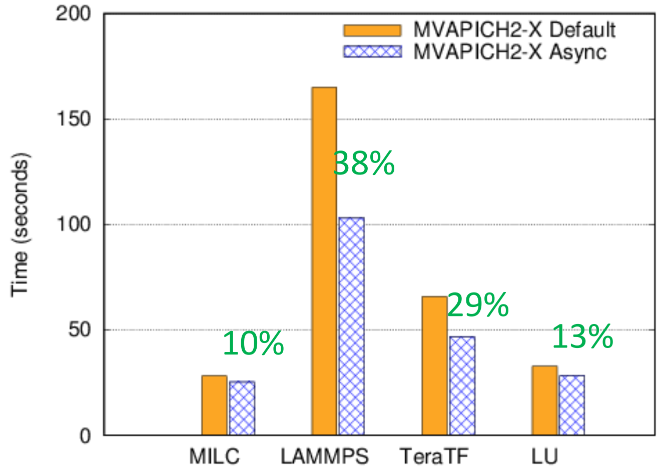
More Details in Poster: Designing Shared Address Space MPI libraries in the Many-core Era - Jahanzeb Hashmi, The Ohio State University

Enhanced Asynchronous Progress Communication

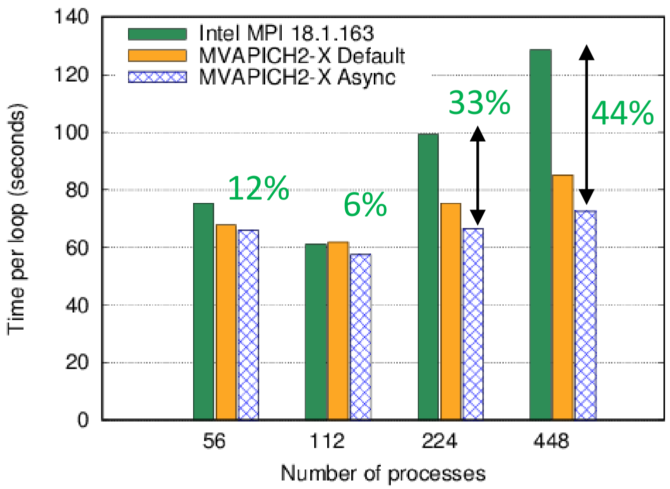
- Achieve overlaps without using specialized hardware or software resources
- Can work with MPI tasks in full-subscribed mode



SPEC MPI : KNL + Omni-Path



SPEC MPI : Skylake + Omni-Path



P3DFFT : Broadwell + InfiniBand

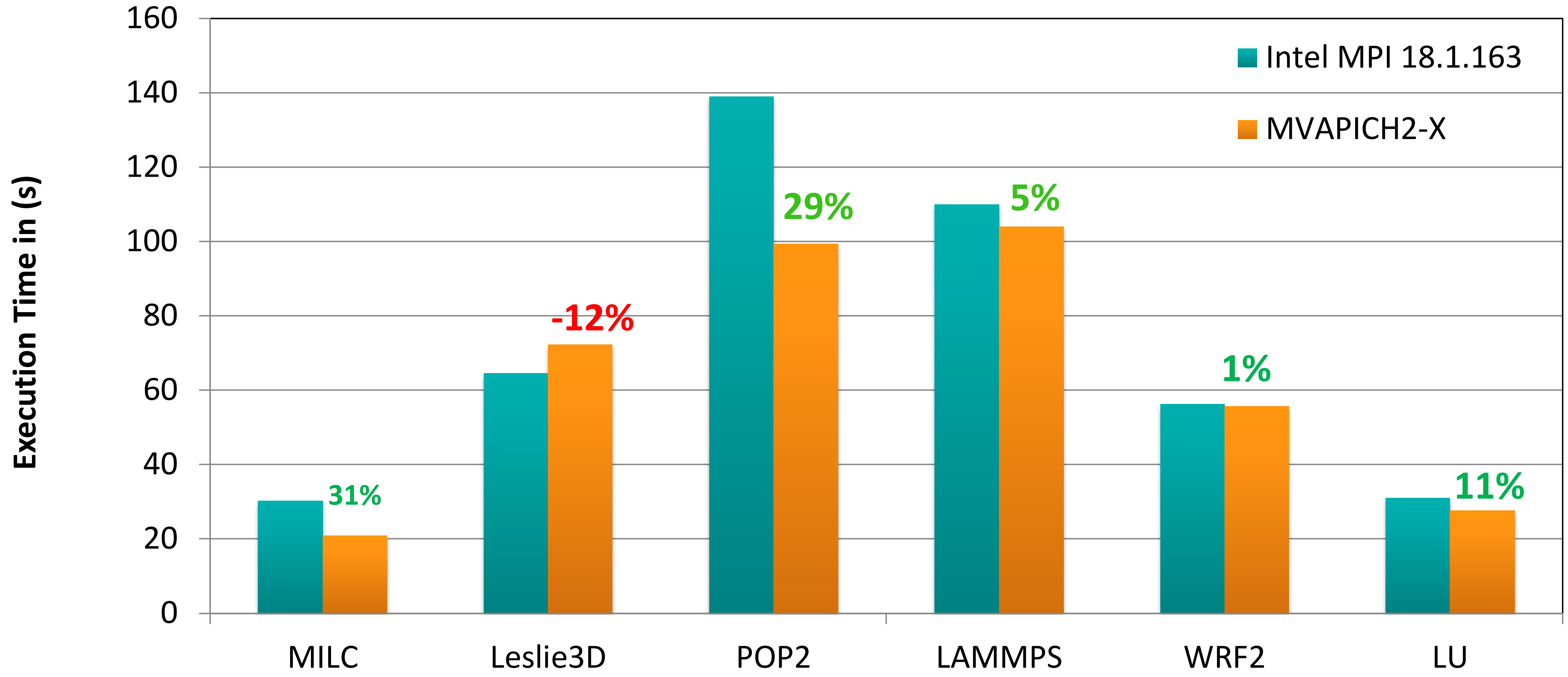
38% performance improvement for SPECMPI applications on 384 processes

44% performance improvement with the P3DFFT application on 448 processes.

A. Ruhela, H. Subramoni, S. Chakraborty, M. Bayatpour, P. Kousha, and D. K. Panda, Efficient Asynchronous Communication Progress for MPI without Dedicated Resources, EuroMPI 2018

Efficient Asynchronous Communication Progress for MPI without Dedicated Resources - Amit Ruhela, The Ohio State University

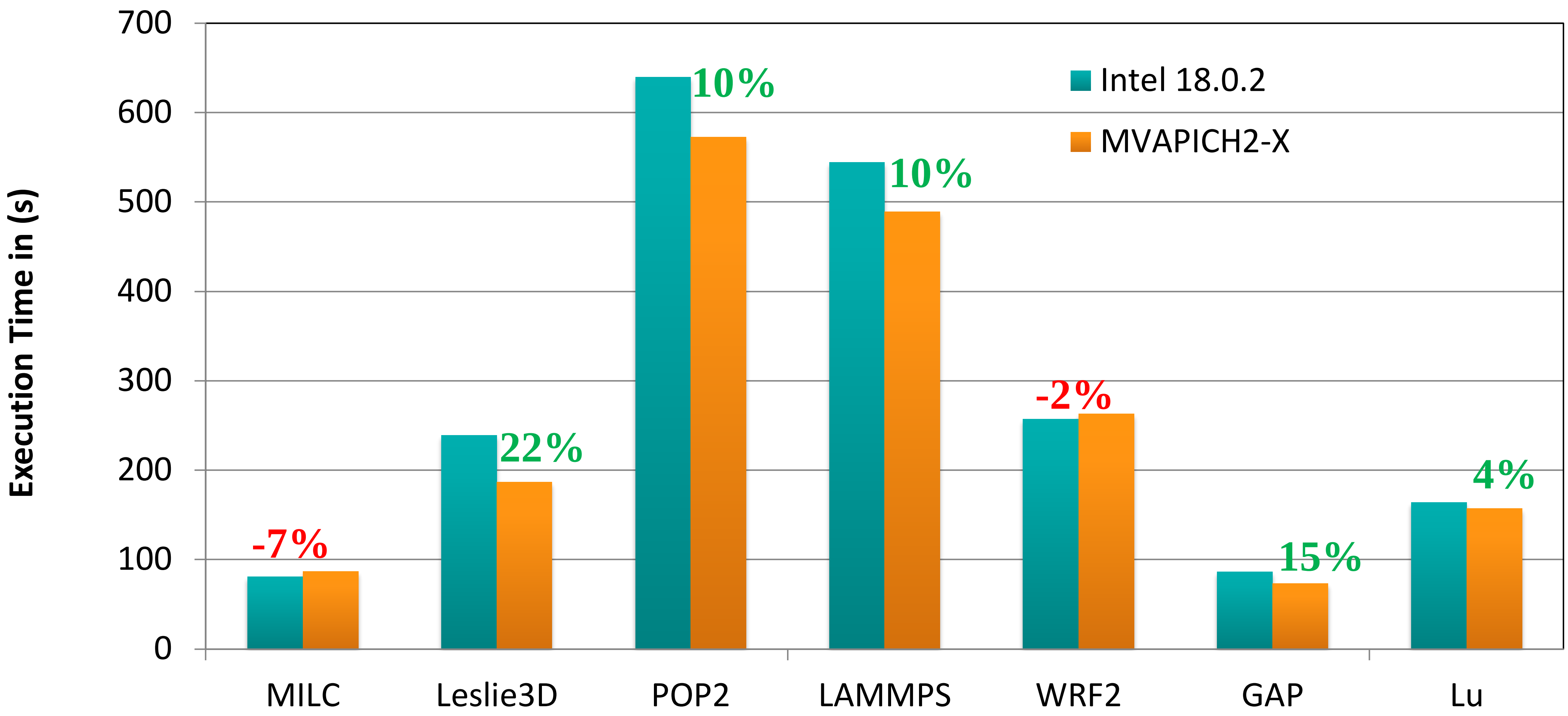
SPEC MPI 2007 Benchmarks: Broadwell + InfiniBand



MVAPICH2-X outperforms Intel MPI by up to 31%

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

SPEC MPI 2007 Benchmarks: KNL + Omni-Path

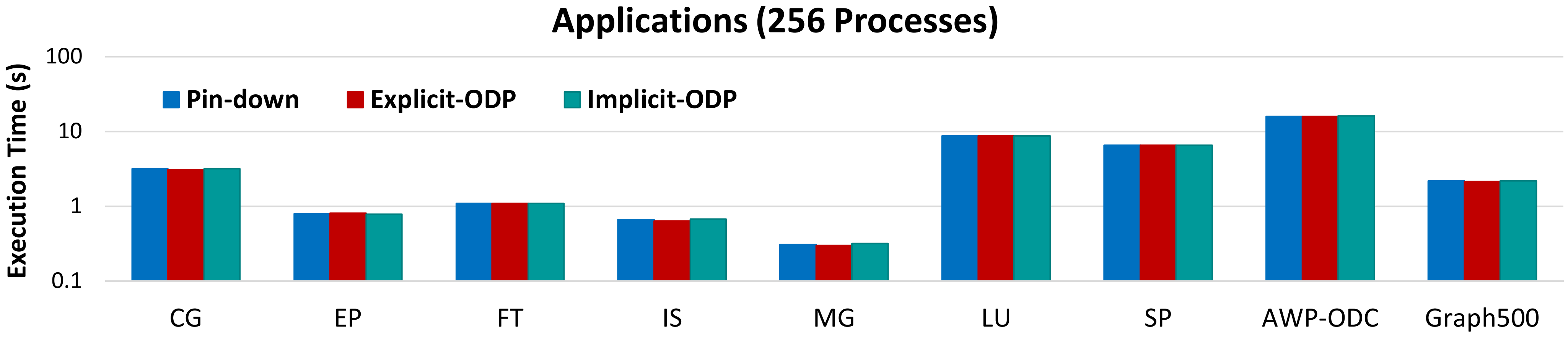


MVAPICH2-X outperforms Intel MPI by up to 22%

Configuration : 384 processes on 8 nodes of Intel Xeon Phi 7250(KNL) with 48 processes per node. KNL contains 68 cores on a single socket and interconnects with 100Gb/sec Intel Omni-Path network.

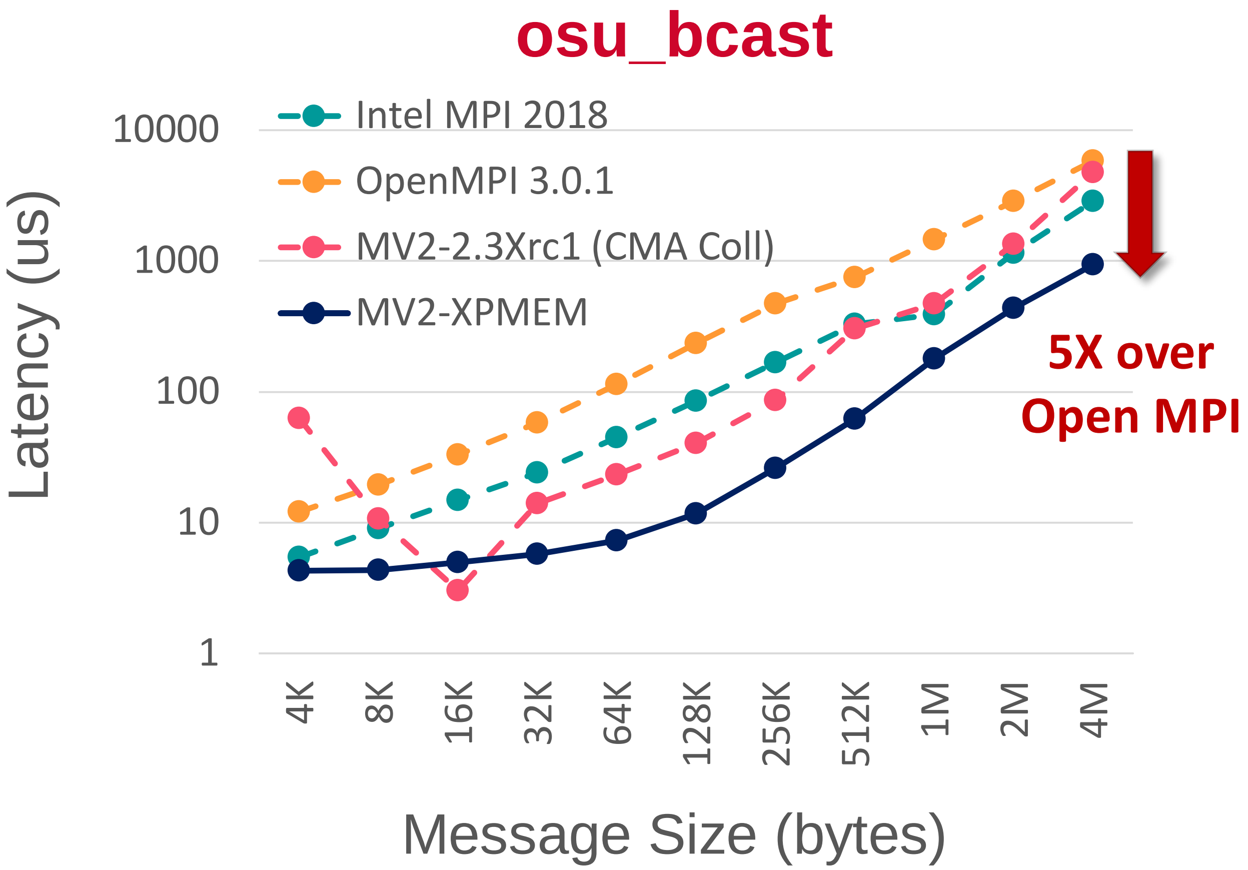
Implicit On-Demand Paging (ODP)

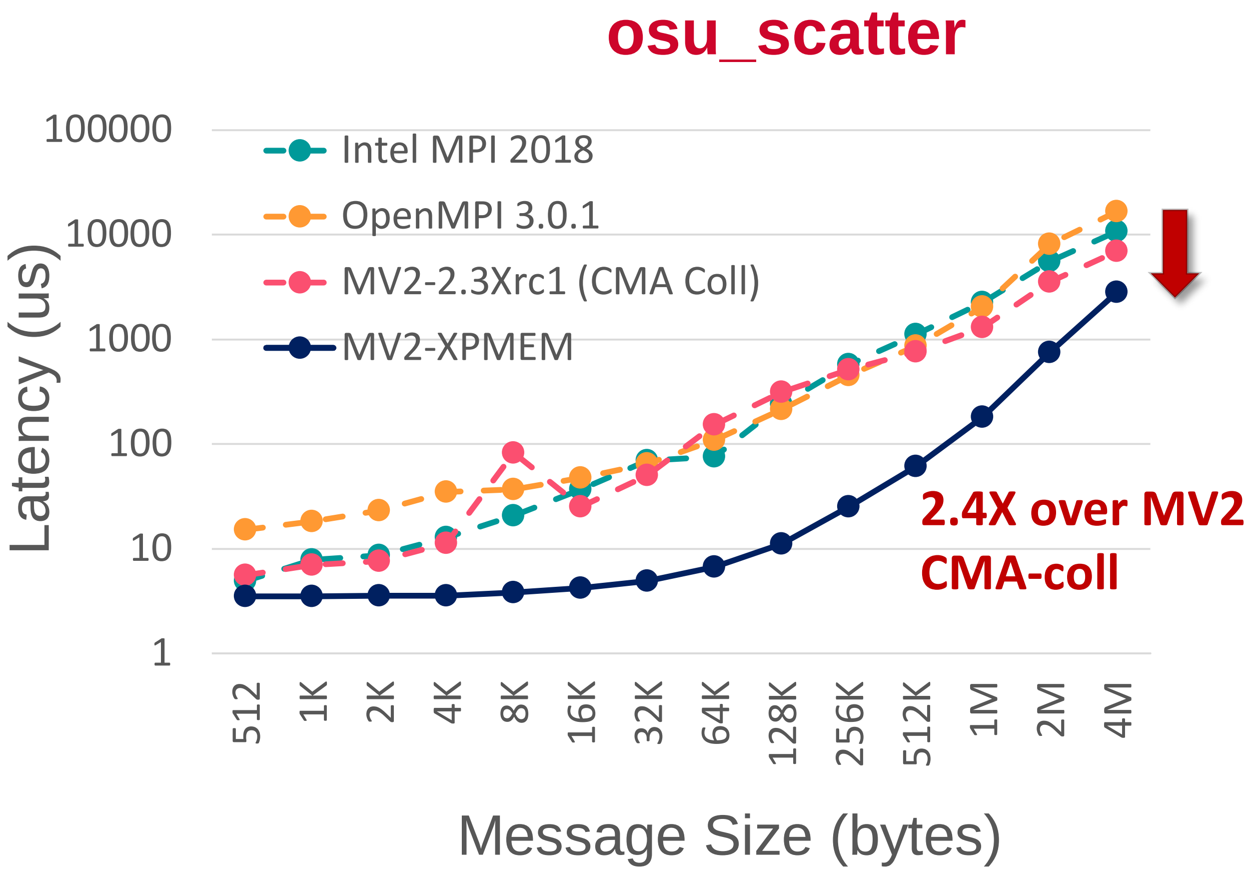
- Introduced by Mellanox to avoid pinning the pages of registered memory regions
- ODP-aware runtime could reduce the size of pin-down buffers while maintaining performance



M. Li, X. Lu, H. Subramoni, and D. K. Panda, "Designing Registration Caching Free High-Performance MPI Library with Implicit On-Demand Paging (ODP) of InfiniBand", HiPC '17

XPMEM Collectives: osu_bcast and osu_scatter

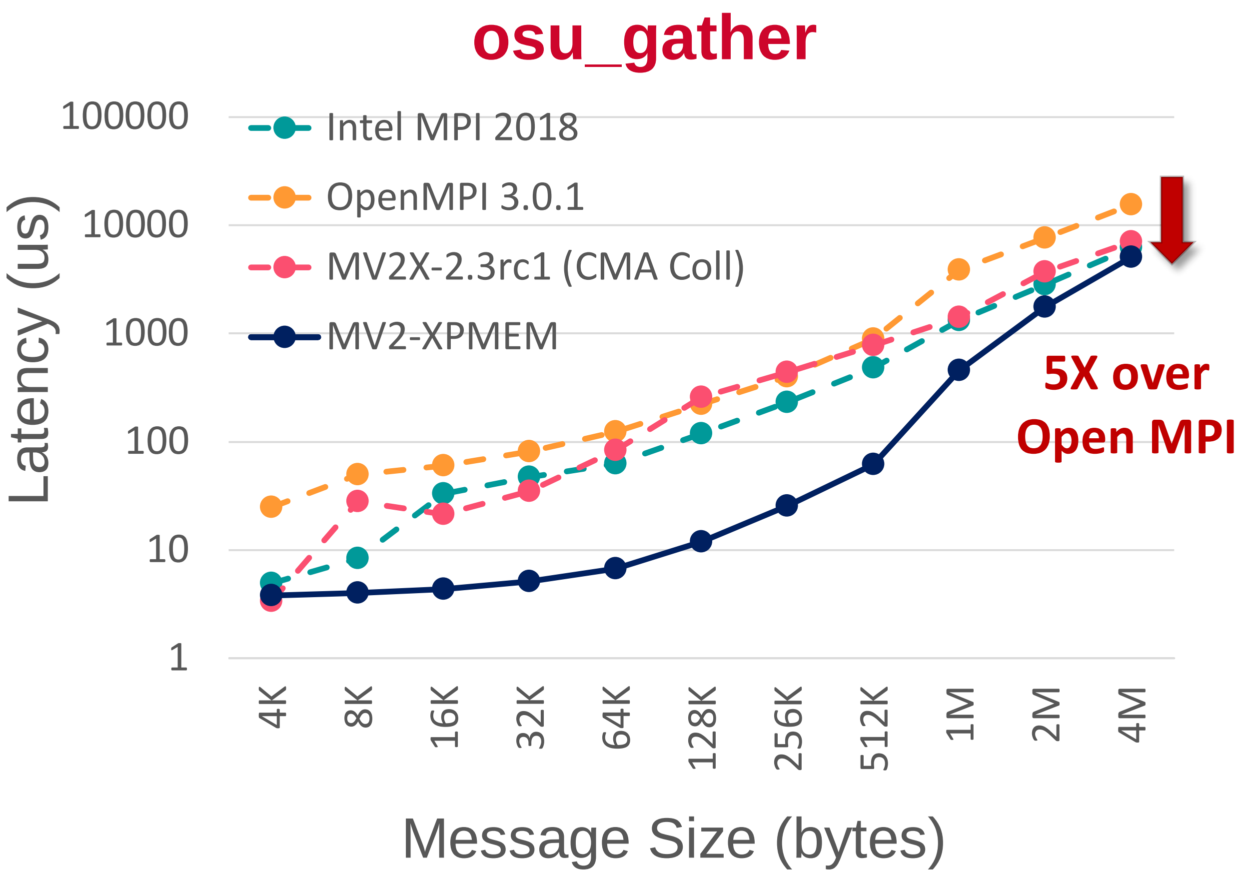


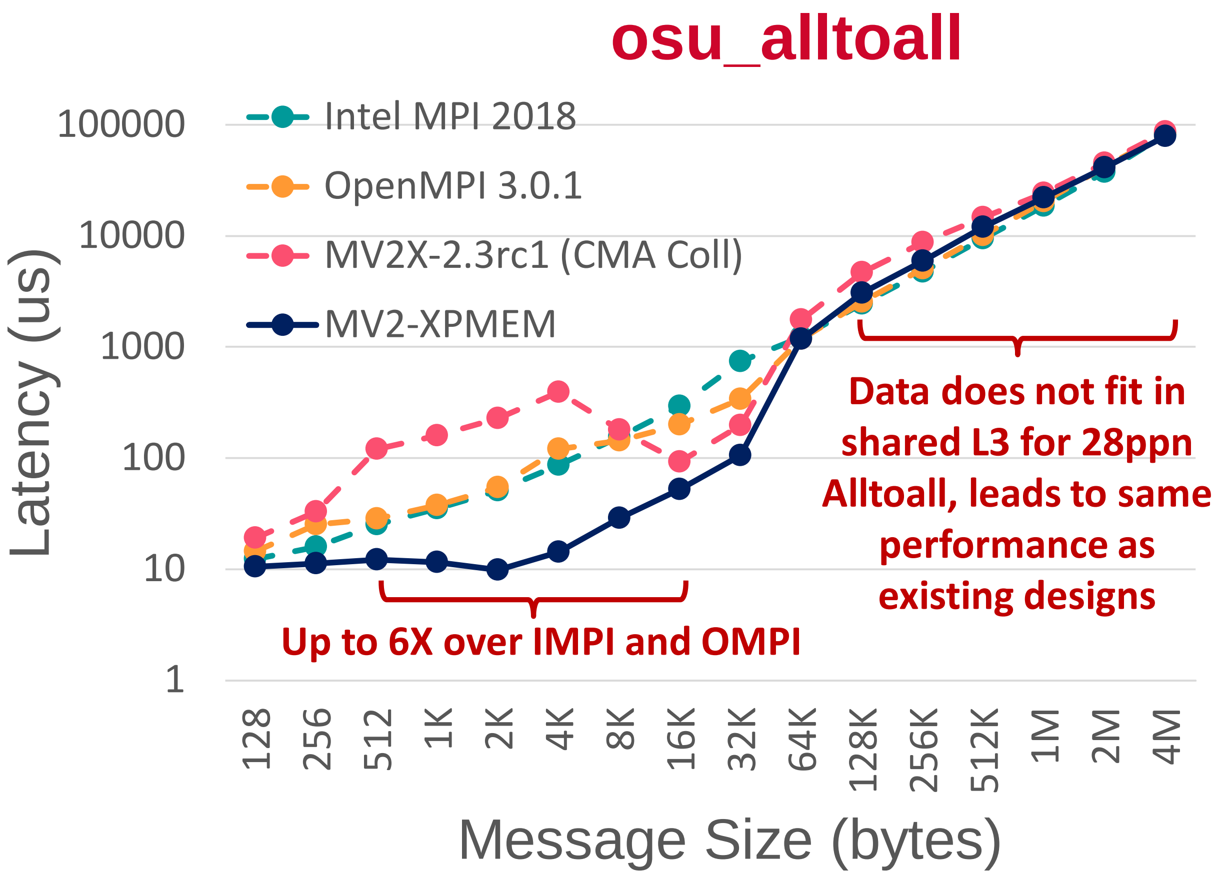


- **28 MPI Processes** on single dual-socket Broadwell E5-2680v4, 2x14 core processor

More Details in Poster: Designing Shared Address Space MPI libraries in the Many-core Era - Jahanzeb Hashmi, The Ohio State University

XPMEM Collectives: osu_gather and osu_alltoall





- **28 MPI Processes** on single dual-socket Broadwell E5-2680v4, 2x14 core processor

More Details in Poster: Designing Shared Address Space MPI libraries in the Many-core Era - Jahanzeb Hashmi, The Ohio State University

MVAPICH2 Software Family

Requirements	Library
MPI with IB, iWARP, Omni-Path, and RoCE	MVAPICH2
Advanced MPI Features/Support, OSU INAM, PGAS and MPI+PGAS with IB, Omni-Path, and RoCE	MVAPICH2-X
MPI with IB, RoCE & GPU and Support for Deep Learning	MVAPICH2-GDR
HPC Cloud with MPI & IB	MVAPICH2-Virt
Energy-aware MPI with IB, iWARP and RoCE	MVAPICH2-EA
MPI Energy Monitoring Tool	OEMT
InfiniBand Network Analysis and Monitoring	OSU INAM
Microbenchmarks for Measuring MPI and PGAS Performance	OMB

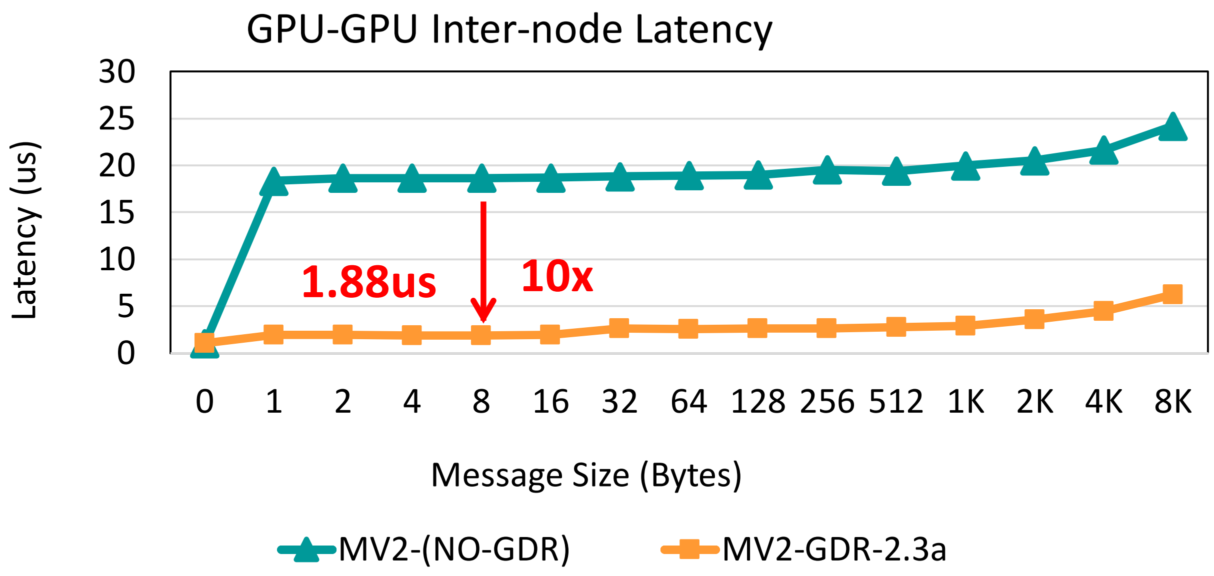
CUDA-Aware MPI: MVAPICH2-GDR 1.8-2.3 Releases

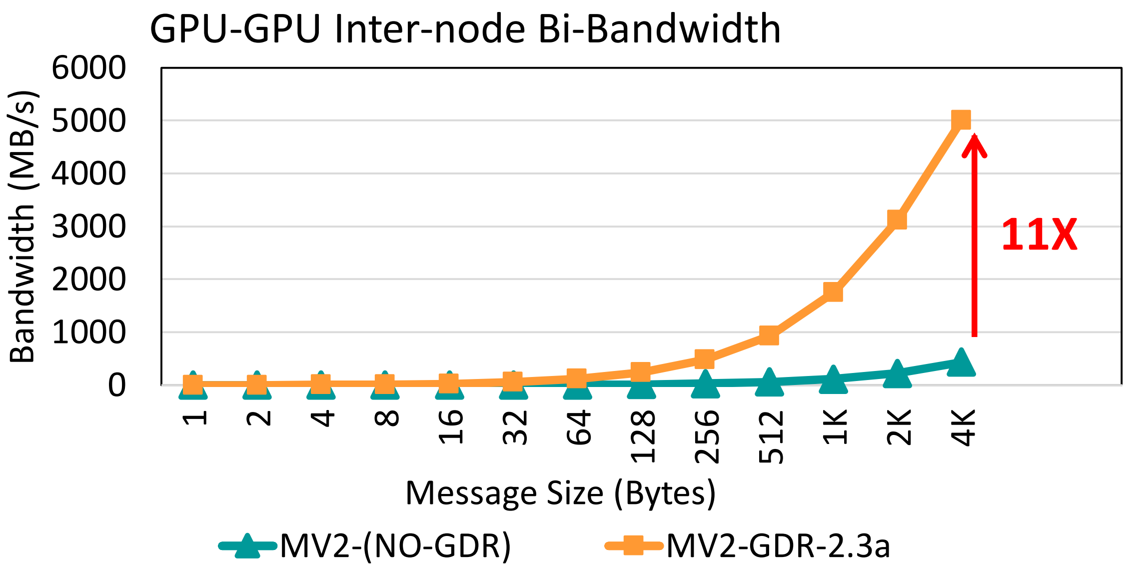
- Support for MPI communication from NVIDIA GPU device memory
- High performance RDMA-based inter-node point-to-point communication (GPU-GPU, GPU-Host and Host-GPU)
- High performance intra-node point-to-point communication for multi-GPU adapters/node (GPU-GPU, GPU-Host and Host-GPU)
- Taking advantage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

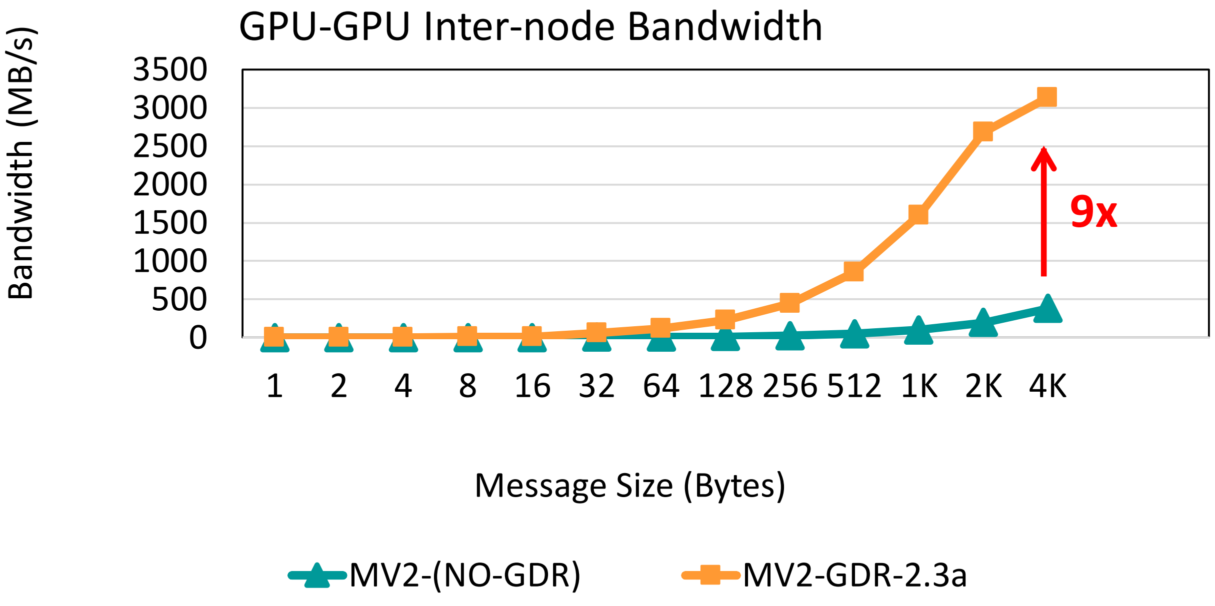
MVAPICH2-GDR 2.3a

- Released on 11/09/2017
- Major Features and Enhancements
  - **Based on MVAPICH2 2.2**
  - **Support for CUDA 9.0**
  - **Add support for Volta (V100) GPU**
  - **Support for OpenPOWER with NVLink**
  - **Efficient Multiple CUDA stream-based IPC communication for multi-GPU systems with and without NVLink**
  - **Enhanced performance of GPU-based point-to-point communication**
  - **Leverage Linux Cross Memory Attach (CMA) feature for enhanced host-based communication**
  - **Enhanced performance of MPI_Allreduce for GPU-resident data**
  - **InfiniBand Multicast (IB-MCAST) based designs for GPU-based broadcast and streaming applications**
    - **Basic support for IB-MCAST designs with GPUDirect RDMA**
    - **Advanced support for zero-copy IB-MCAST designs with GPUDirect RDMA**
    - **Advanced reliability support for IB-MCAST designs**
  - **Efficient broadcast designs for Deep Learning applications**
  - **Enhanced collective tuning on Xeon, OpenPOWER, and NVIDIA DGX-1 systems**

Optimized MVAPICH2-GDR Design



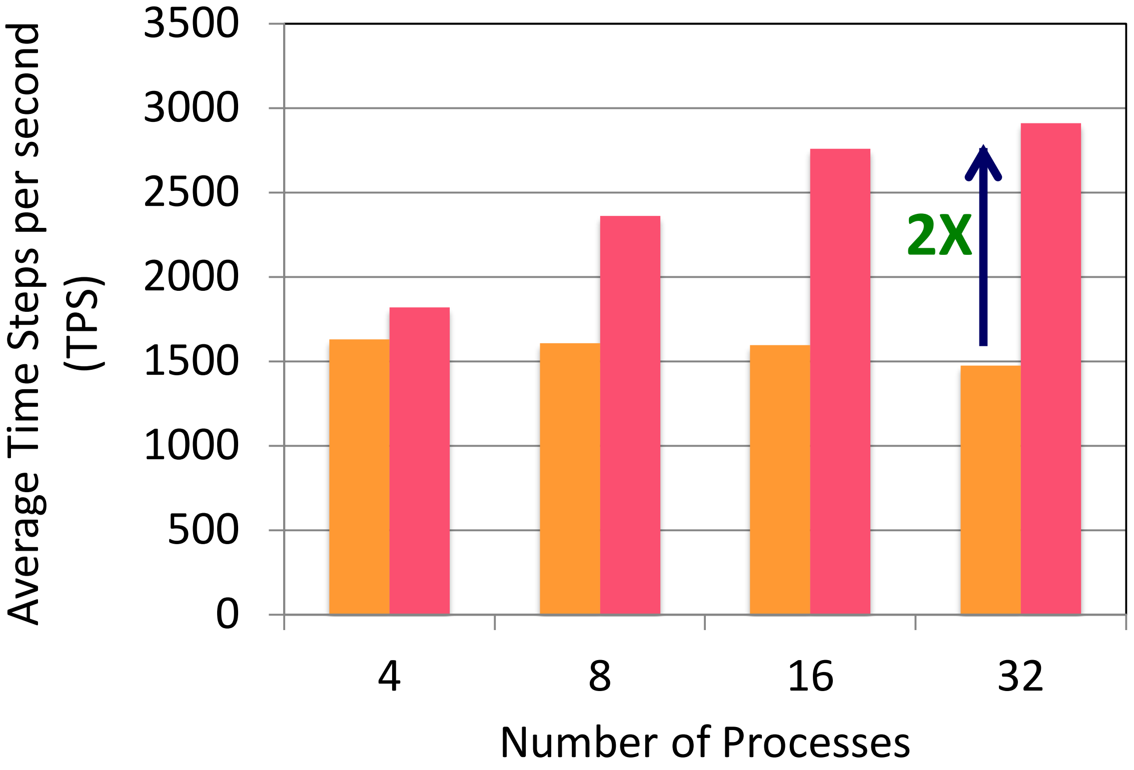




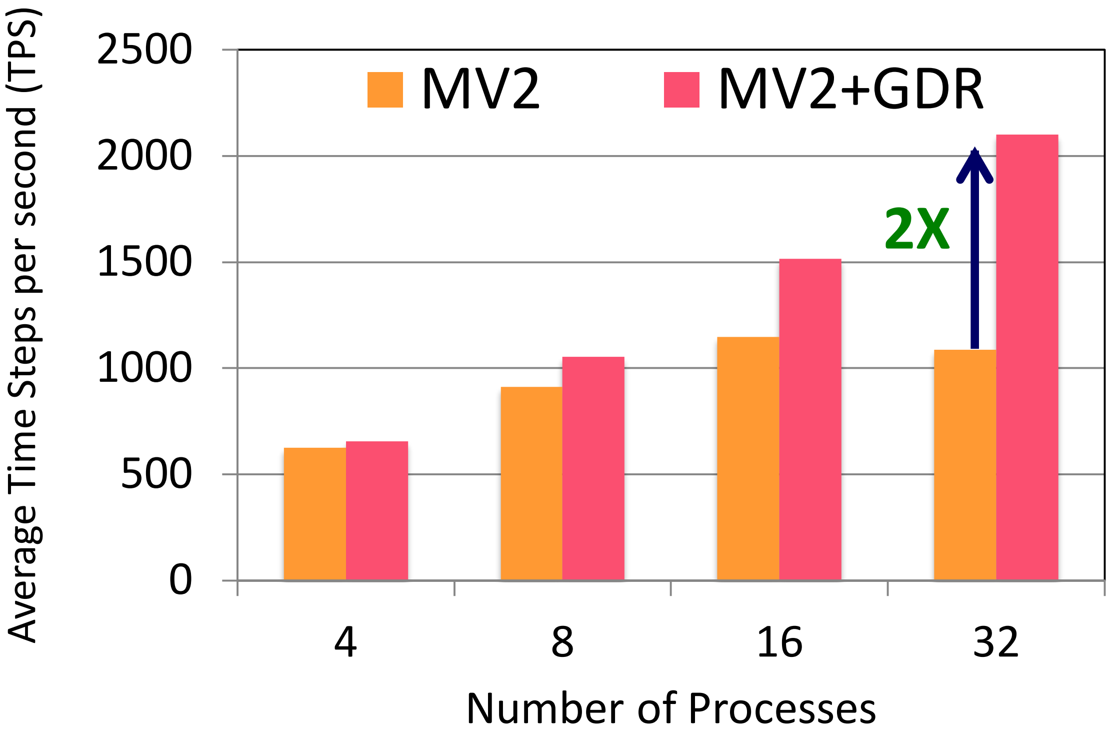
MVAPICH2-GDR-2.3a
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

Application-Level Evaluation (HOOMD-blue)

64K Particles

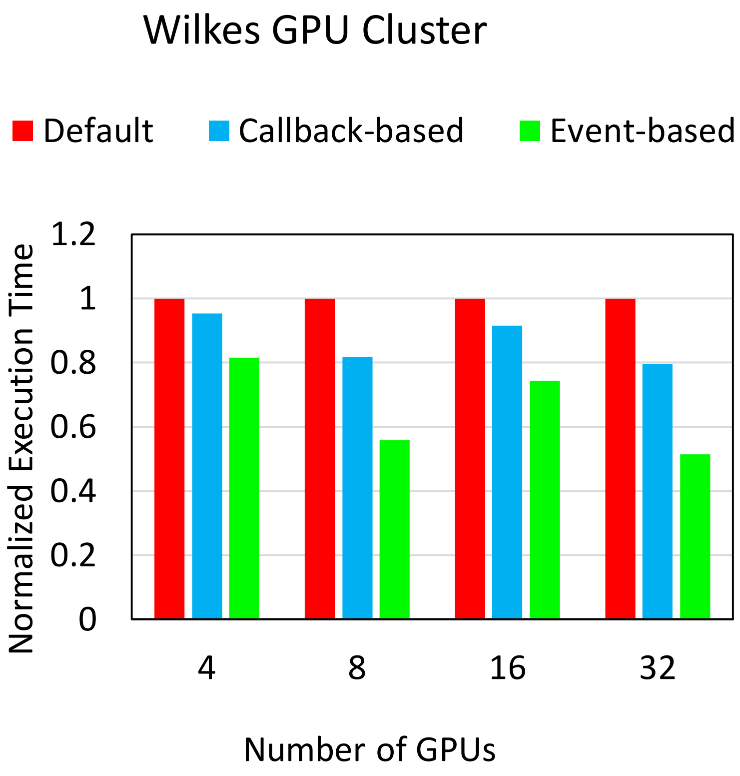


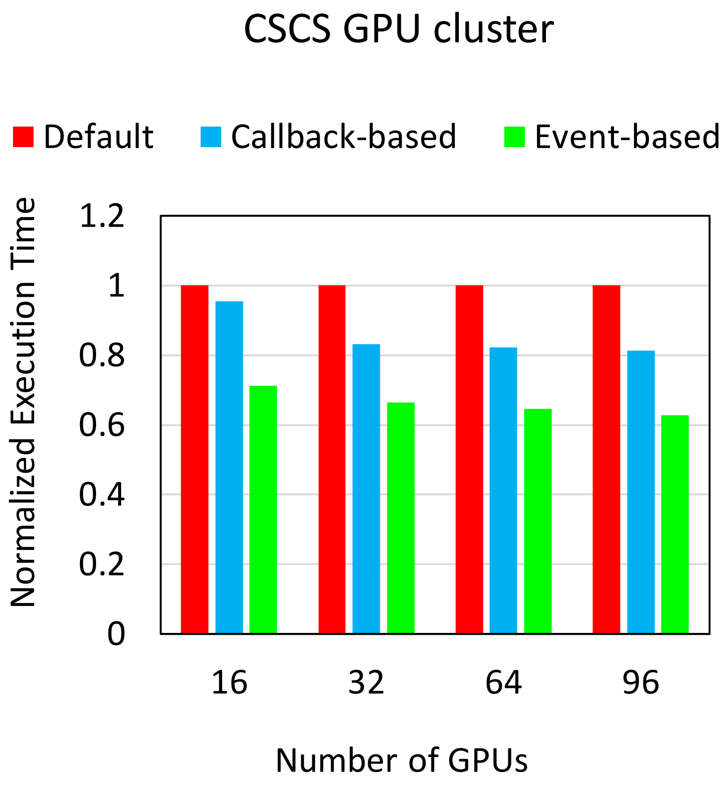
256K Particles

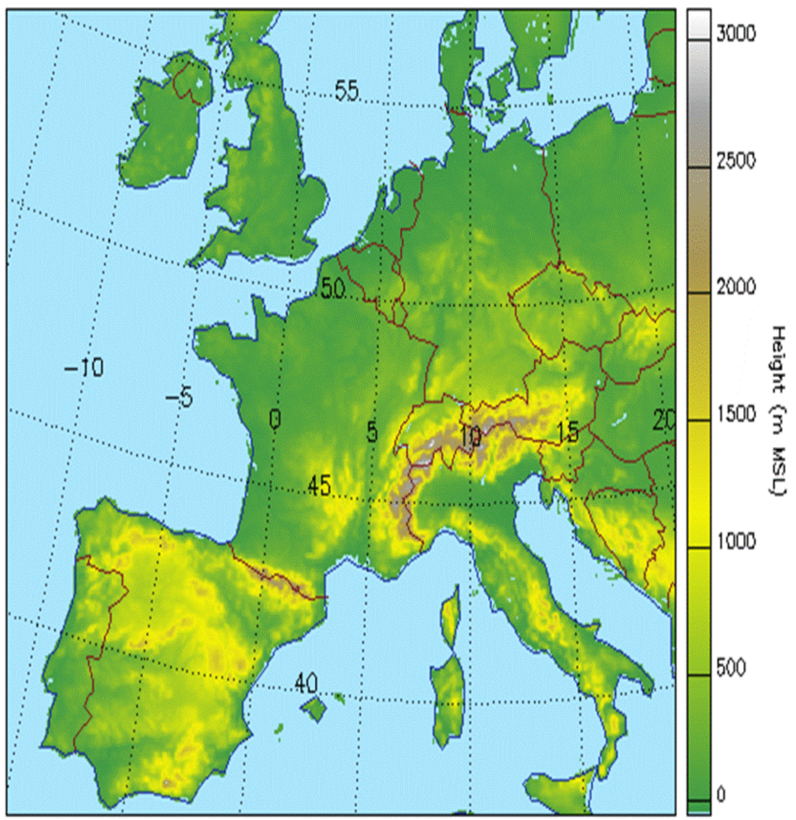


- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- HoomdBlue Version 1.0.5
  - GDRCOPY enabled: MV2_USE_CUDA=1 MV2_IBA_HCA=mlx5_0 MV2_IBA_EAGER_THRESHOLD=32768 MV2_VBUF_TOTAL_SIZE=32768 MV2_USE_GPUDIRECT_LOOPBACK_LIMIT=32768 MV2_USE_GPUDIRECT_GDRCOPY=1 MV2_USE_GPUDIRECT_GDRCOPY_LIMIT=16384

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







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

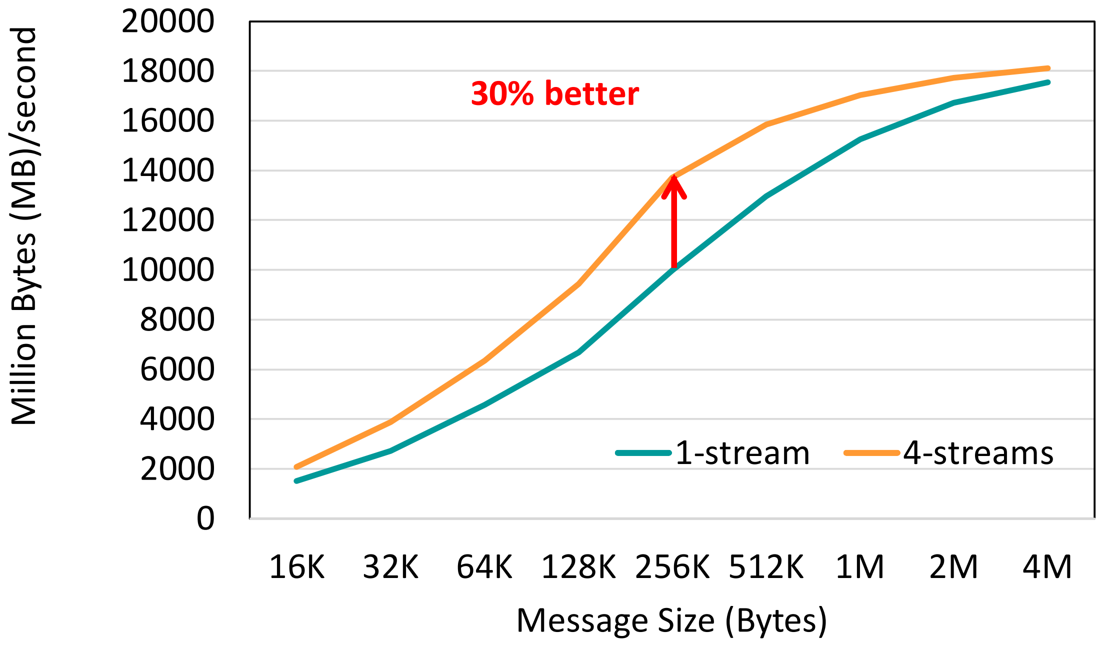
- 2X improvement on 32 GPUs nodes
- 30% improvement on 96 GPU nodes (8 GPUs/node)

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

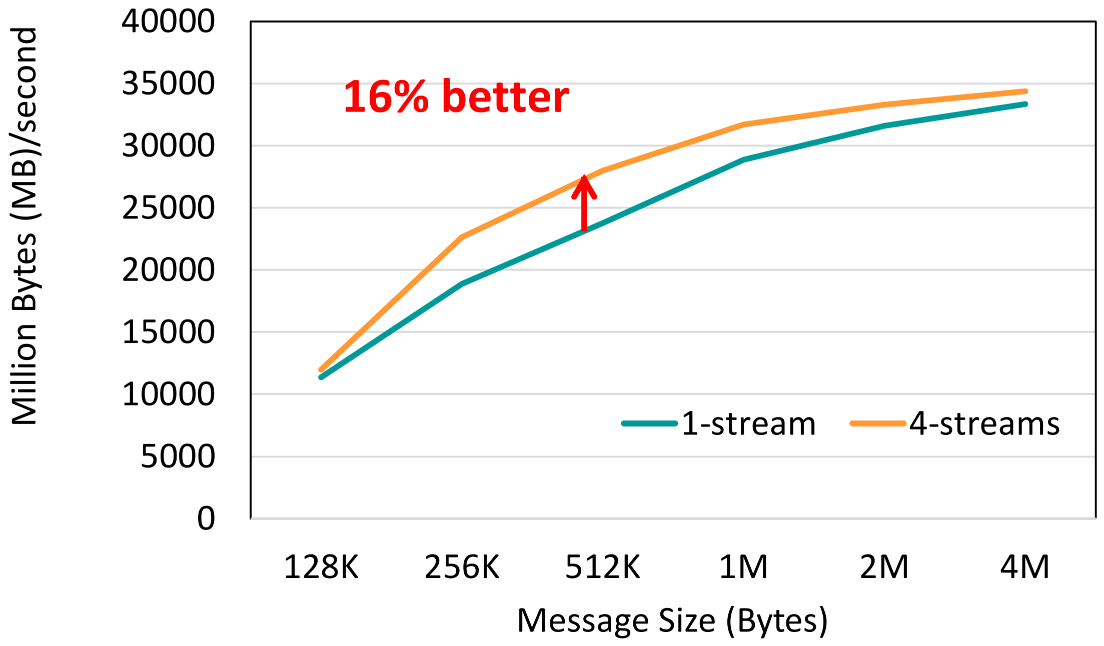
C. Chu, K. Hamidouche, A. Venkatesh, D. Banerjee , H. Subramoni, and D. K. Panda, Exploiting Maximal Overlap for Non-Contiguous Data Movement Processing on Modern GPU-enabled Systems, IPDPS'16

Multi-stream Communication using CUDA IPC on OpenPOWER and DGX-1

- Up to **16% higher** Device to Device (D2D) bandwidth on OpenPOWER + NVLink inter-connect
- Up to **30% higher** D2D bandwidth on DGX-1 with NVLink
- Pt-to-pt (D-D) Bandwidth: Benefits of Multi-stream CUDA IPC Design



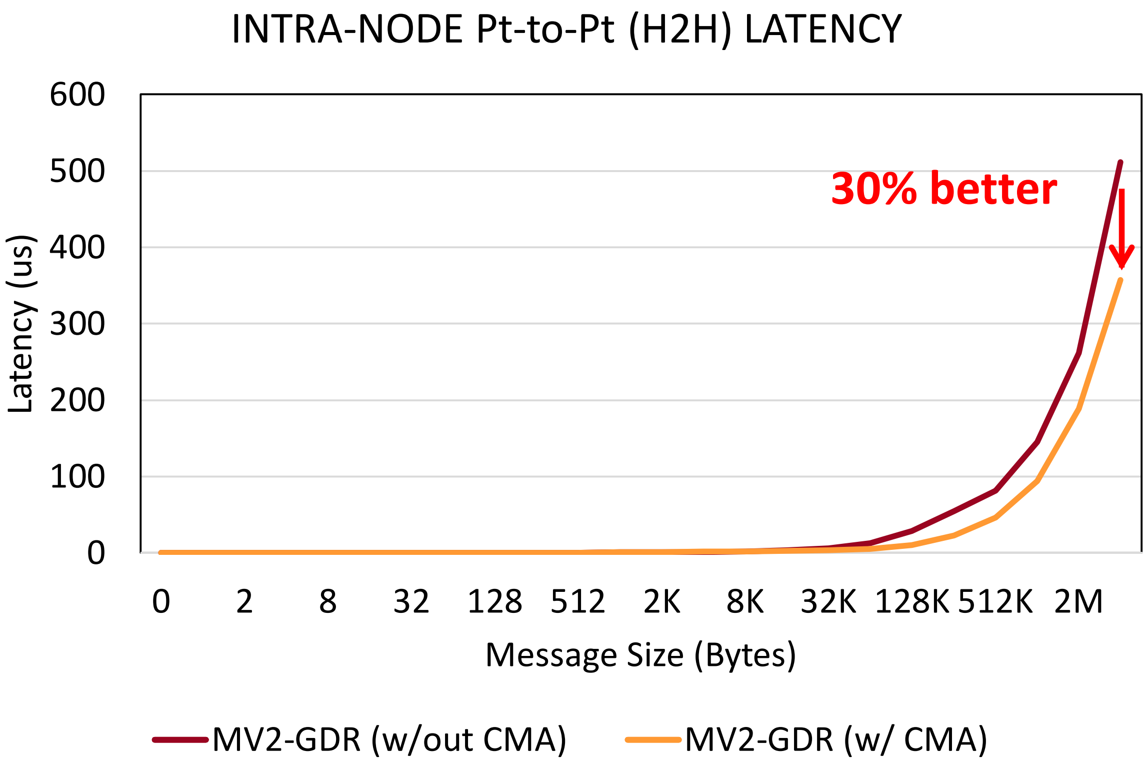
Pt-to-pt (D-D) Bandwidth: Benefits of Multi-stream CUDA IPC Design

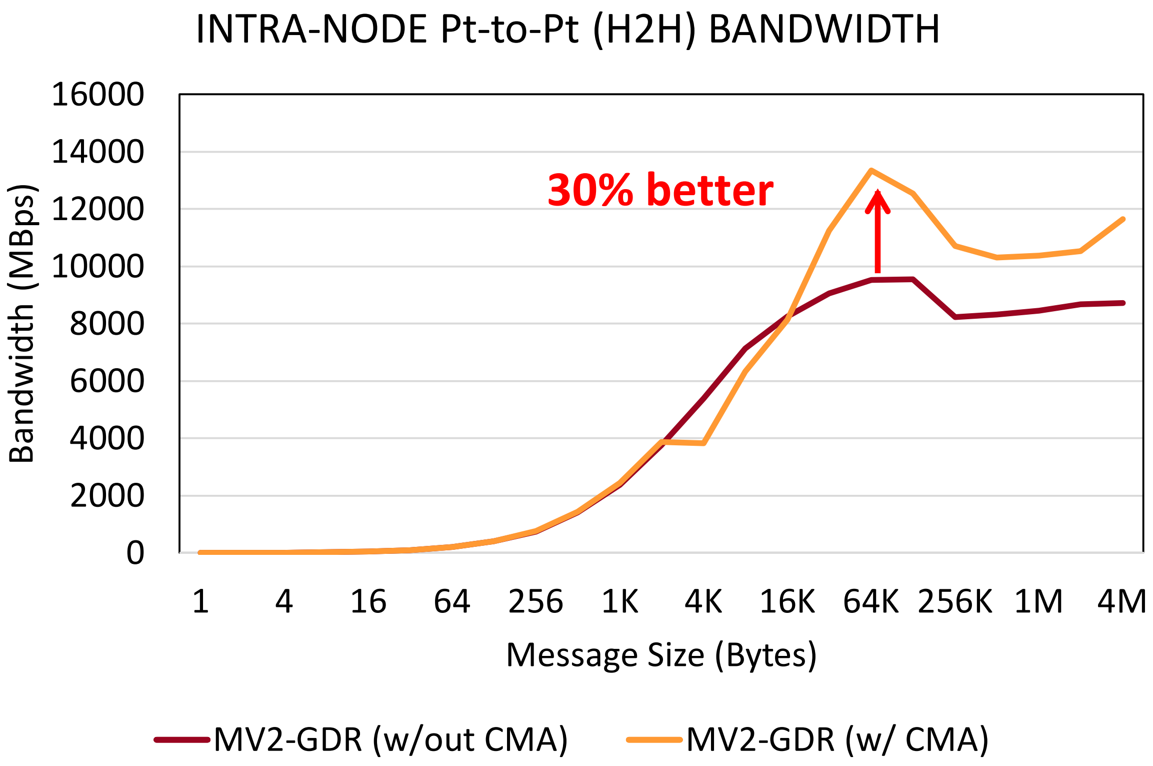


Available since MVAPICH2-GDR-2.3a

CMA-based Intra-node Communication Support

- Up to **30% lower** Host-to-Host (H2H) latency and **30% higher** H2H Bandwidth

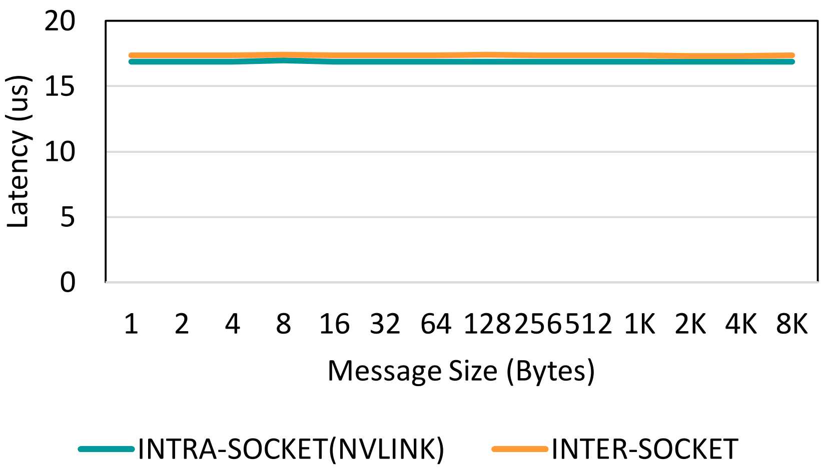




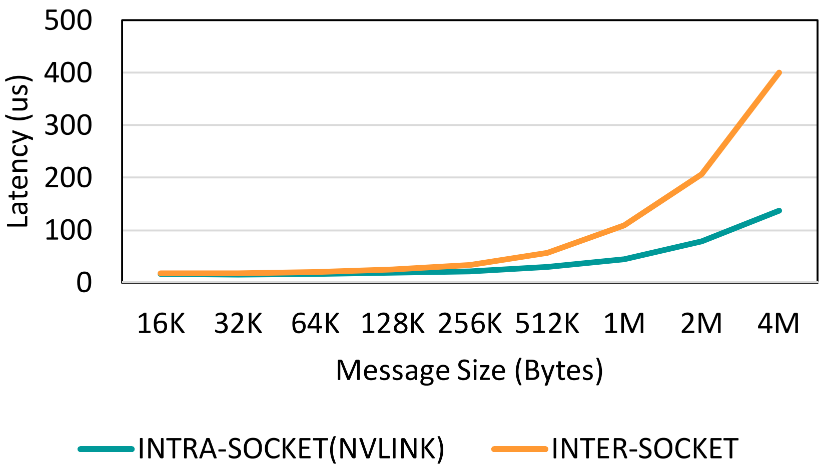
MVAPICH2-GDR-2.3a
Intel Broadwell (E5-2680 v4 @ 3240 GHz) node – 28 cores
NVIDIA Tesla K-80 GPU, and Mellanox Connect-X4 EDR HCA
CUDA 8.0, Mellanox OFED 4.0 with GPU-Direct-RDMA

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

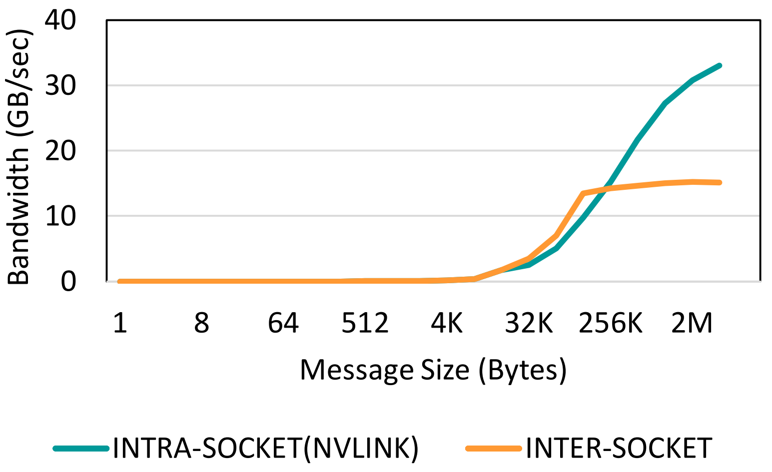
INTRA-NODE LATENCY (SMALL)



INTRA-NODE LATENCY (LARGE)



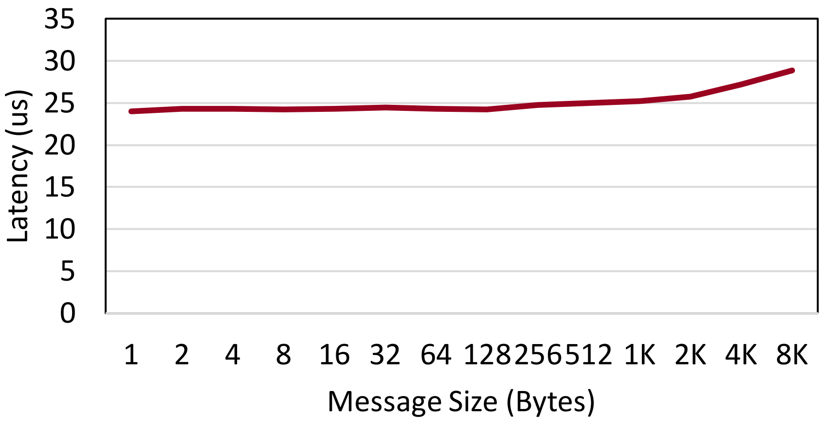
INTRA-NODE BANDWIDTH



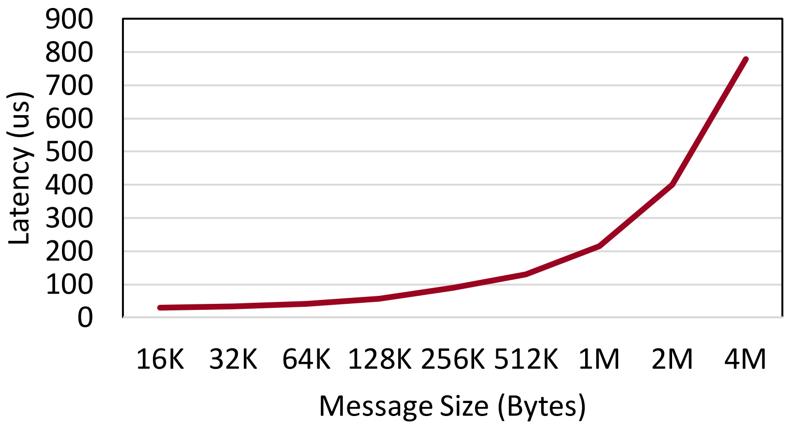
Intra-node Latency: 16.8 us (without GPUDirectRDMA)

Intra-node Bandwidth: 32 GB/sec (NVLINK)

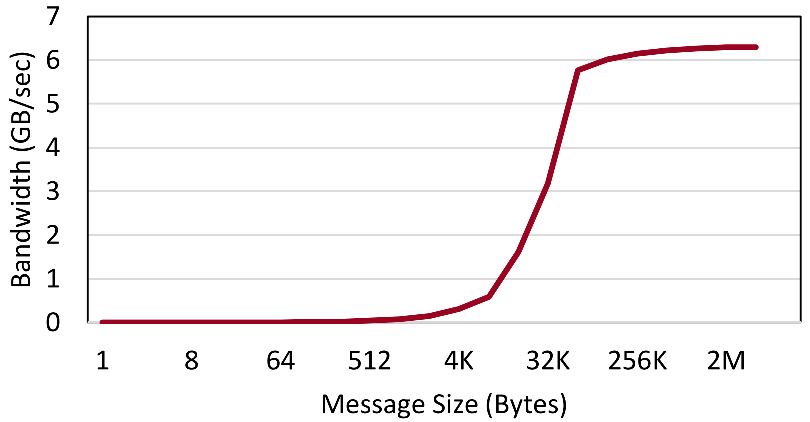
INTER-NODE LATENCY (SMALL)



INTER-NODE LATENCY (LARGE)



INTER-NODE BANDWIDTH



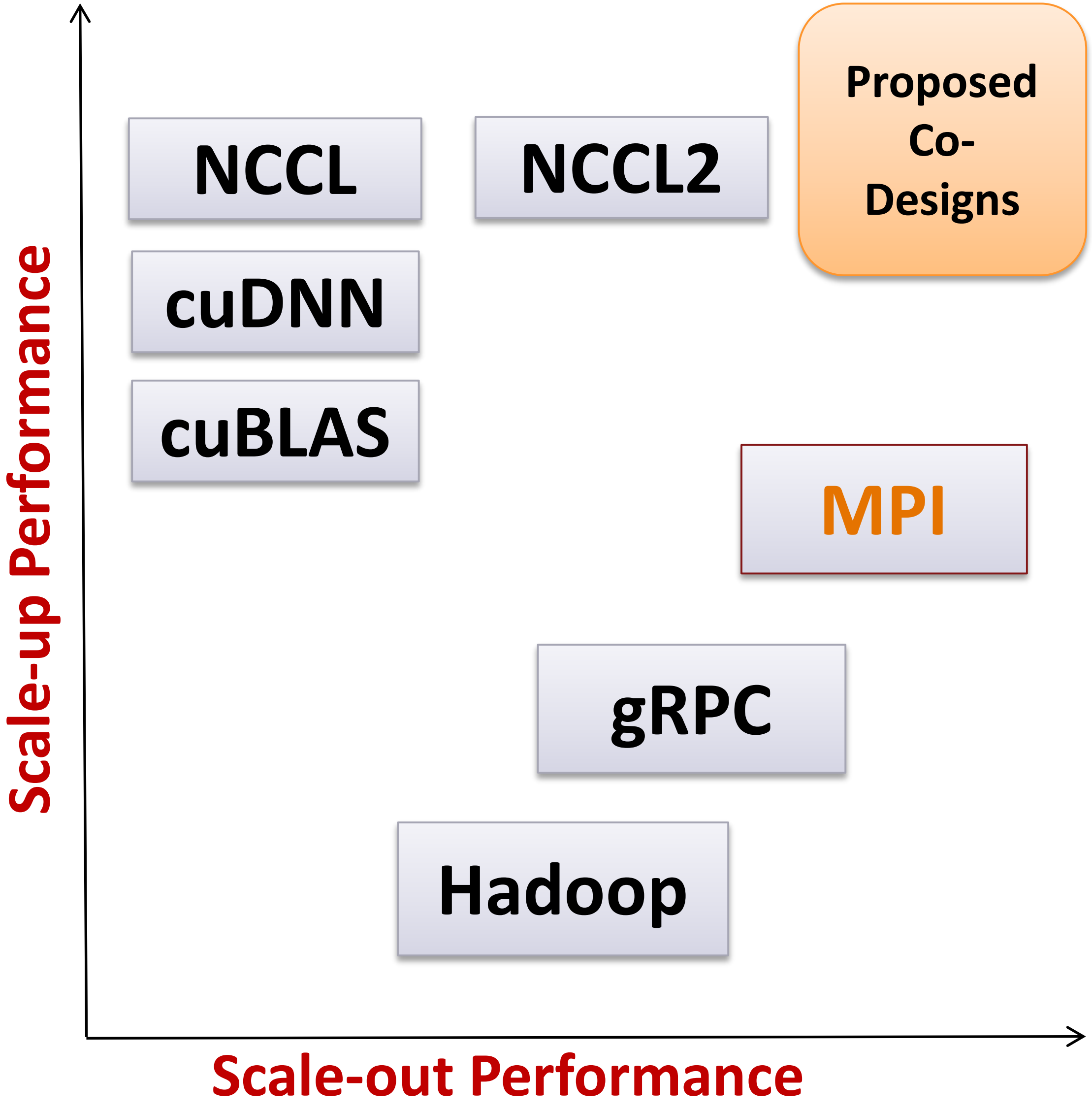
Inter-node Latency: 22 us (without GPUDirectRDMA)

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

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

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**?

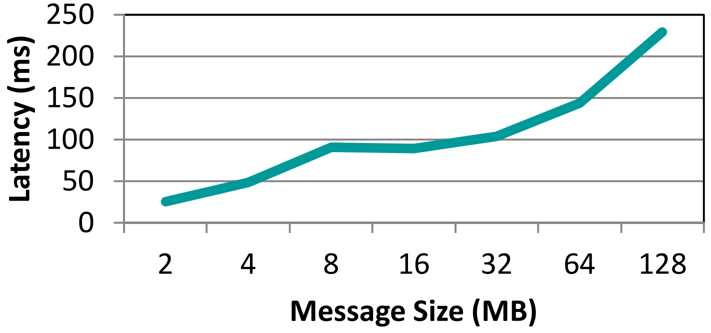


A. A. Awan, K. Hamidouche, J. M. Hashmi, and D. K. Panda, S-Caffe: Co-designing MPI Runtimes and Caffe for Scalable Deep Learning on Modern GPU Clusters. In *Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (PPoPP '17)

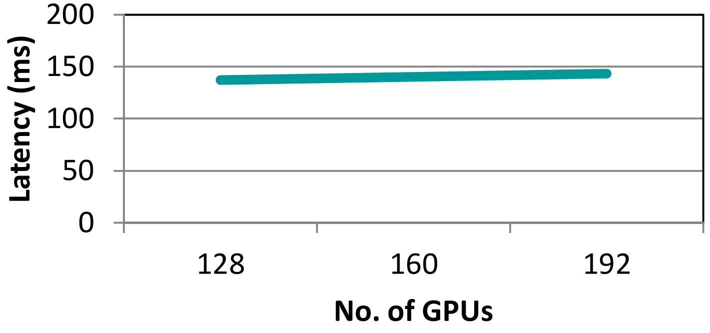
Large Message Optimized Collectives for Deep Learning

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

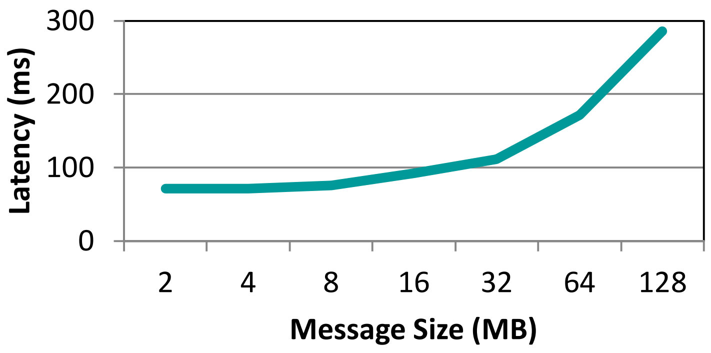
Reduce – 192 GPUs



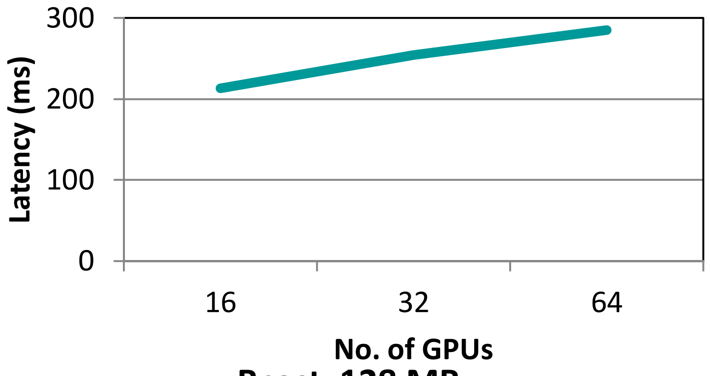
Reduce – 64 MB



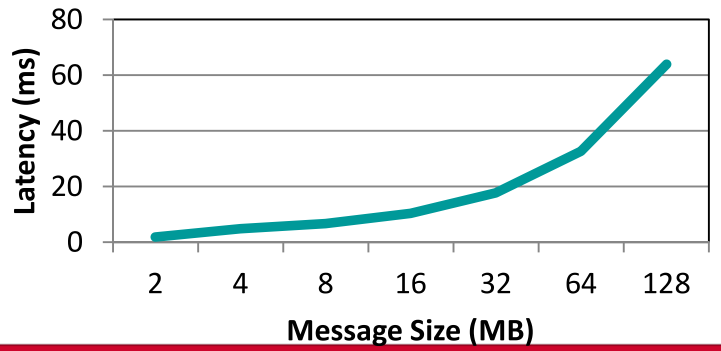
Allreduce – 64 GPUs



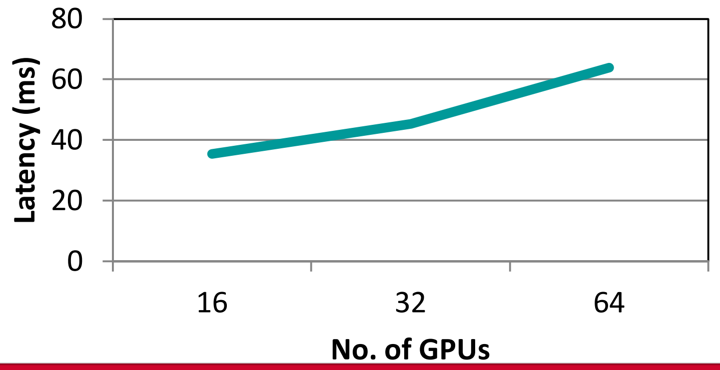
Allreduce - 128 MB



Bcast – 64 GPUs

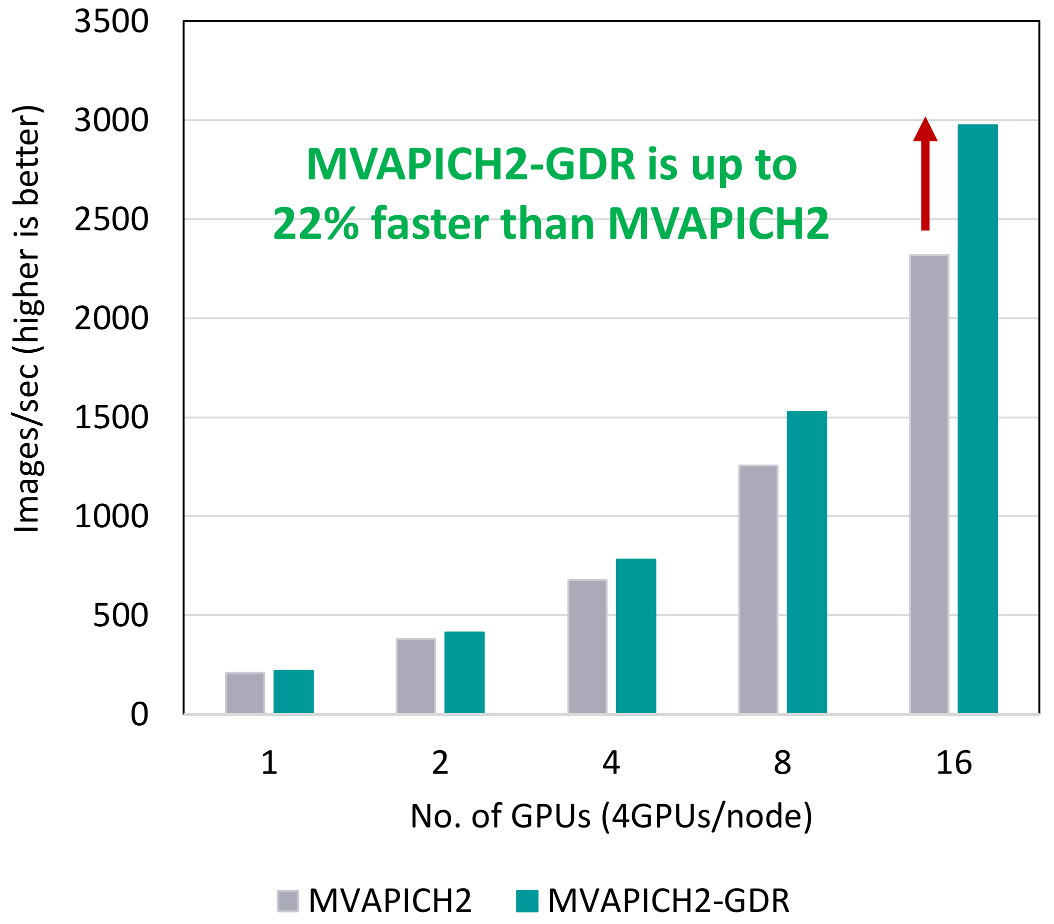


Bcast 128 MB

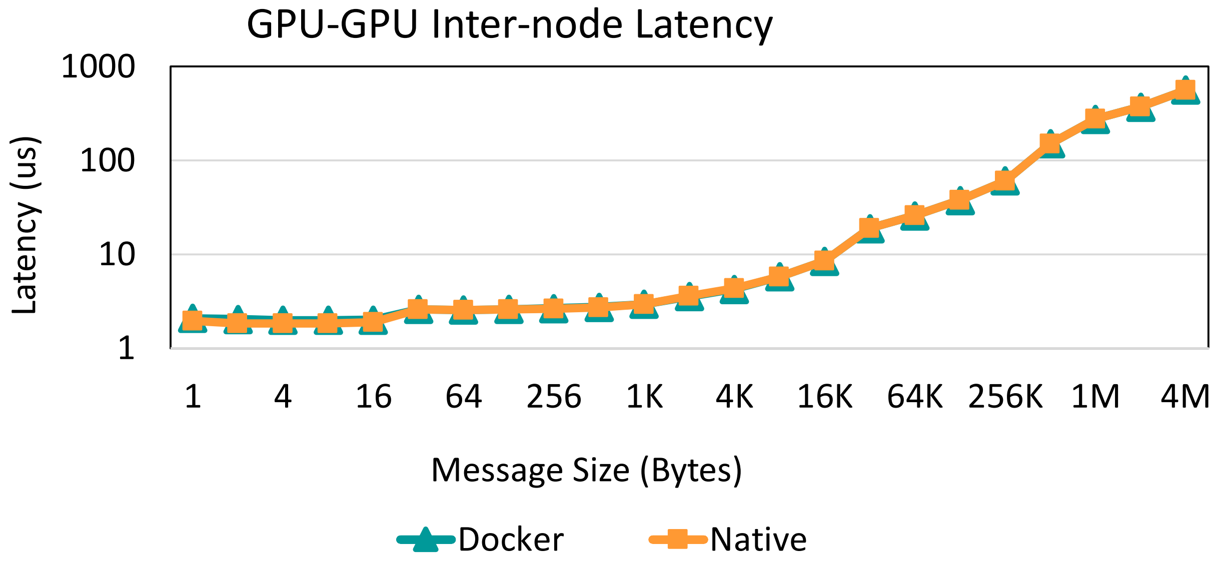


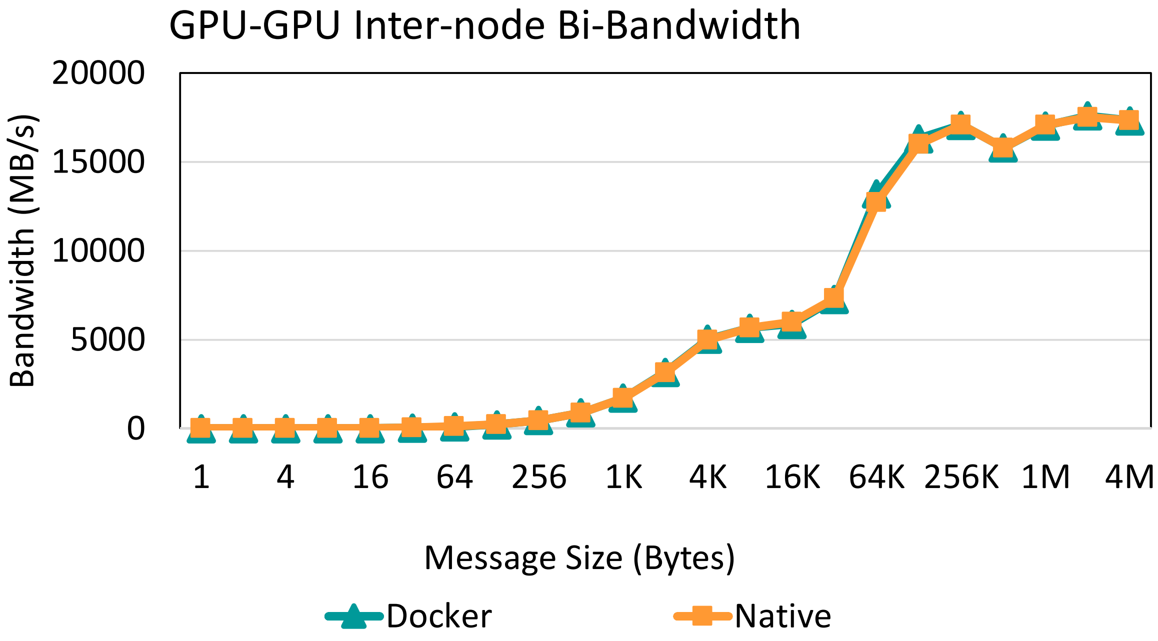
Exploiting CUDA-Aware MPI for TensorFlow (Horovod)

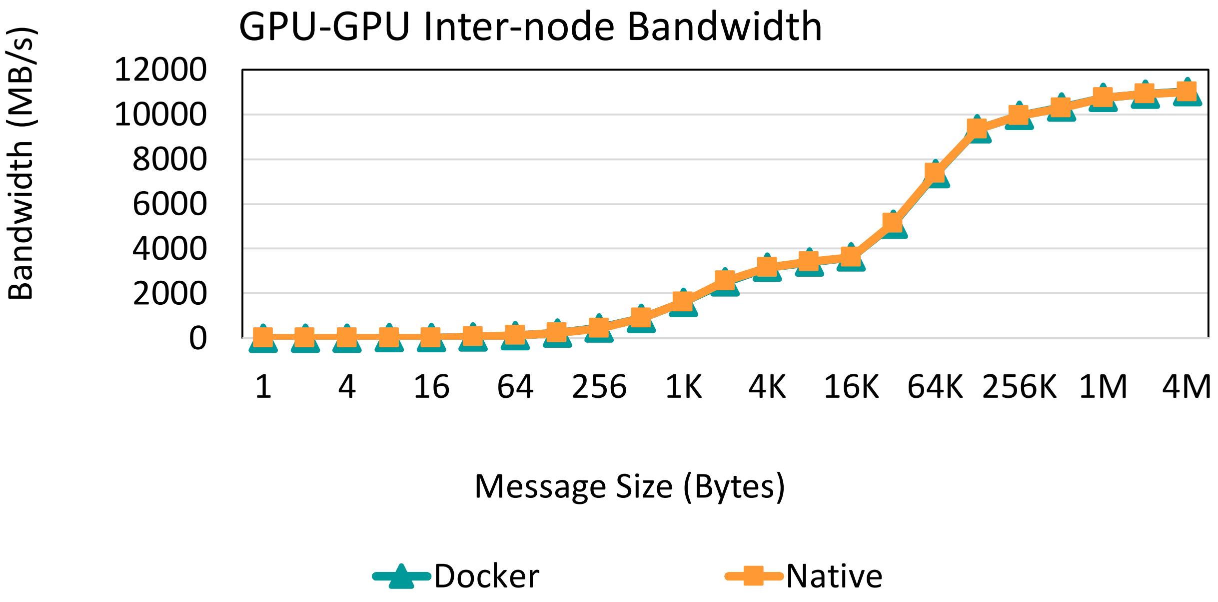
- MVAPICH2-GDR offers excellent performance via advanced designs for MPI_Allreduce.
- Up to **22% better** performance on Wilkes2 cluster (16 GPUs)



MVAPICH2-GDR on Container with Negligible Overhead







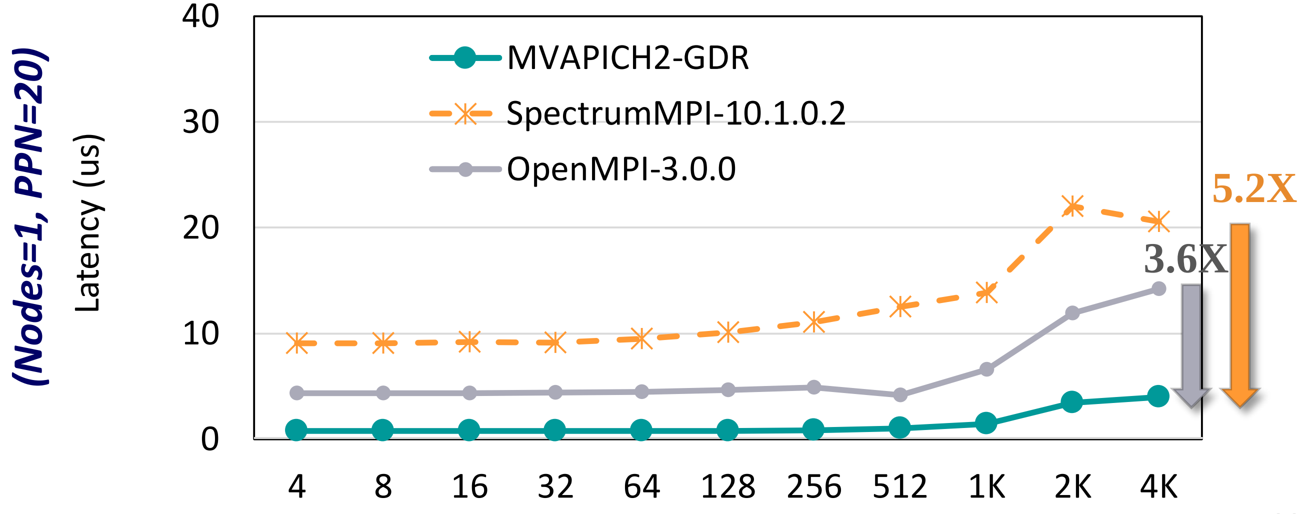
MVAPICH2-GDR-2.3a
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

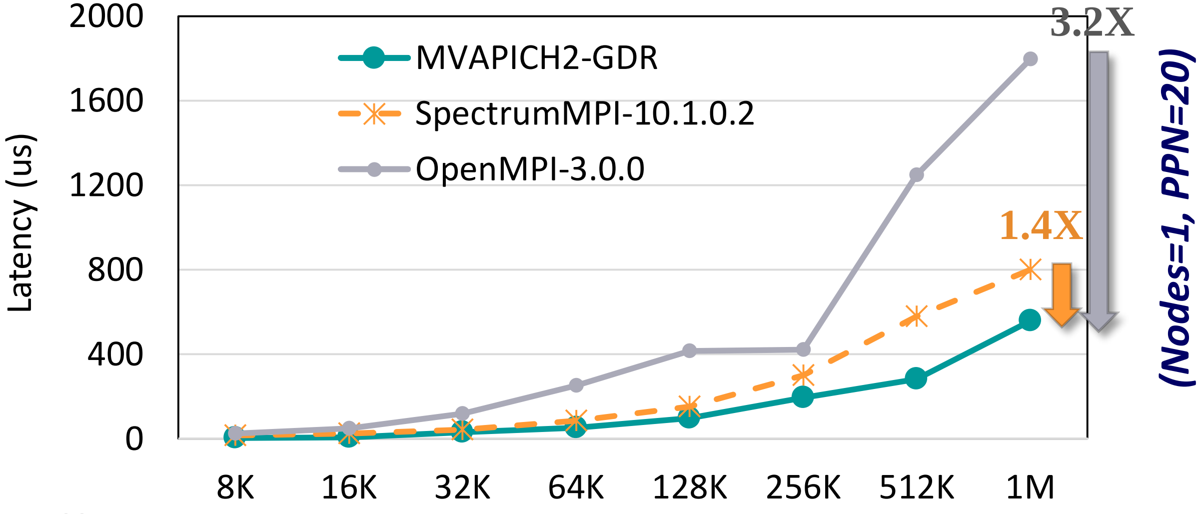
MVAPICH2-GDR Upcoming Features

- MVAPICH2-GDR 2.3b will be released soon
  - Scalable Host-based Collectives
  - Optimized Support for Deep Learning (Caffe and CNTK)
  - Support for Streaming Applications
- Optimized Collectives for Multi-Rails (Sierra)
- Integrated Collective Support with SHArP

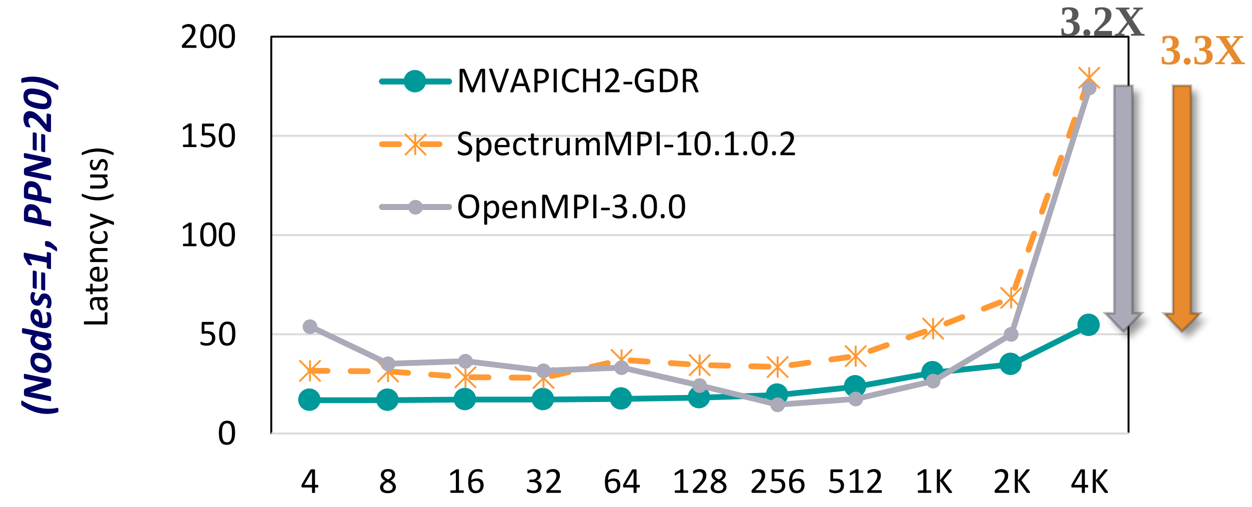
Scalable Host-based Collectives on OpenPOWER (Intra-node Reduce & AlltoAll)

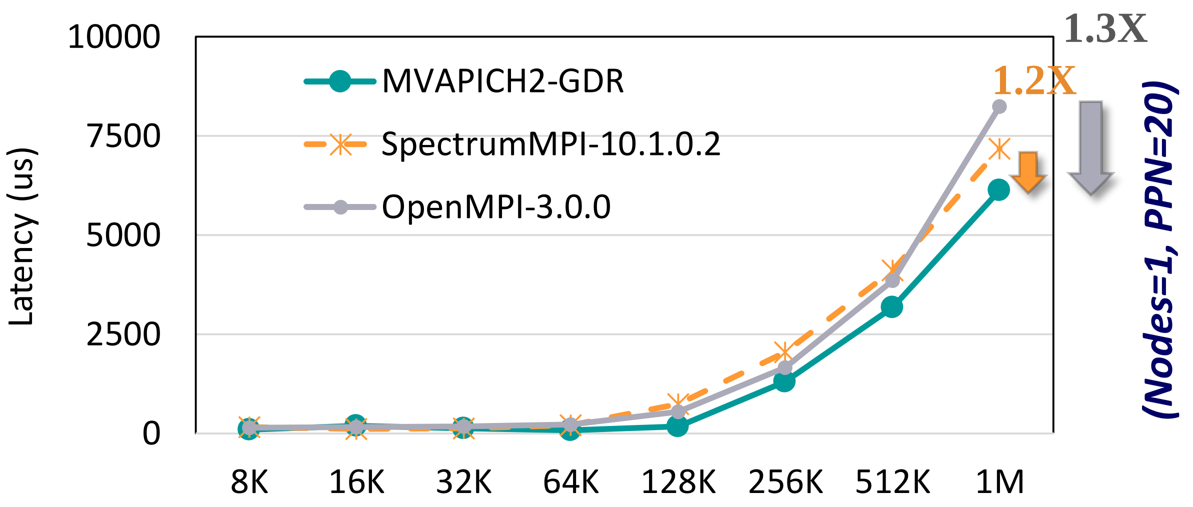
Reduce





Alltoall

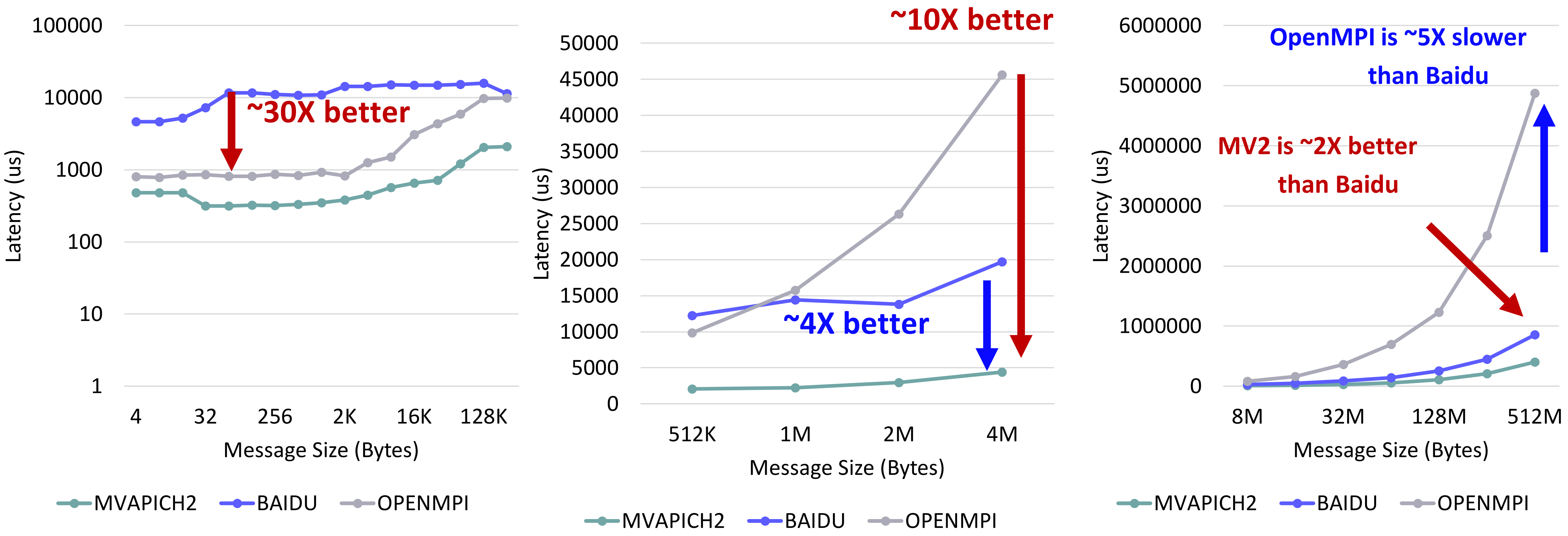




Up to 5X and 3x performance improvement by MVAPICH2 for small and large messages respectively

MVAPICH2-GDR: Allreduce Comparison with Baidu and OpenMPI

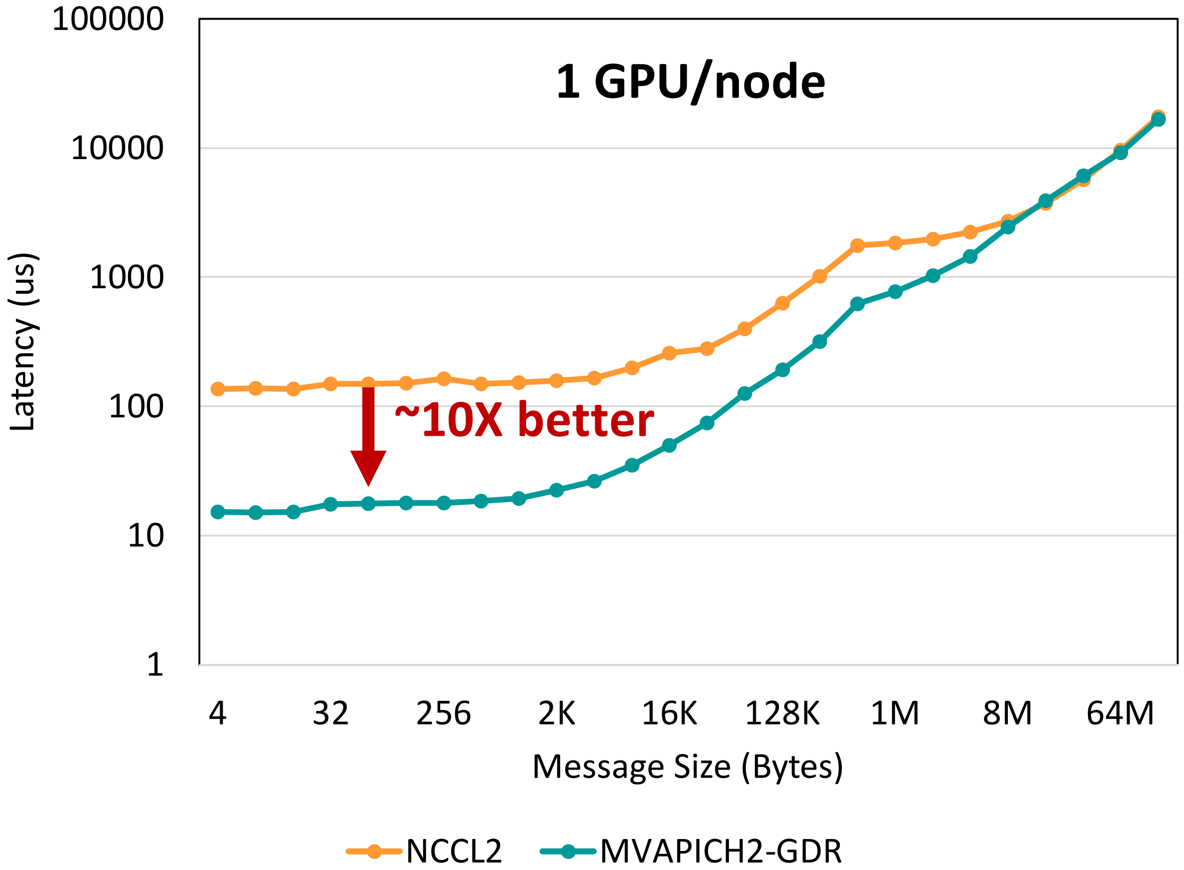
- 16 GPUs (4 nodes) MVAPICH2-GDR vs. Baidu-Allreduce and OpenMPI 3.0

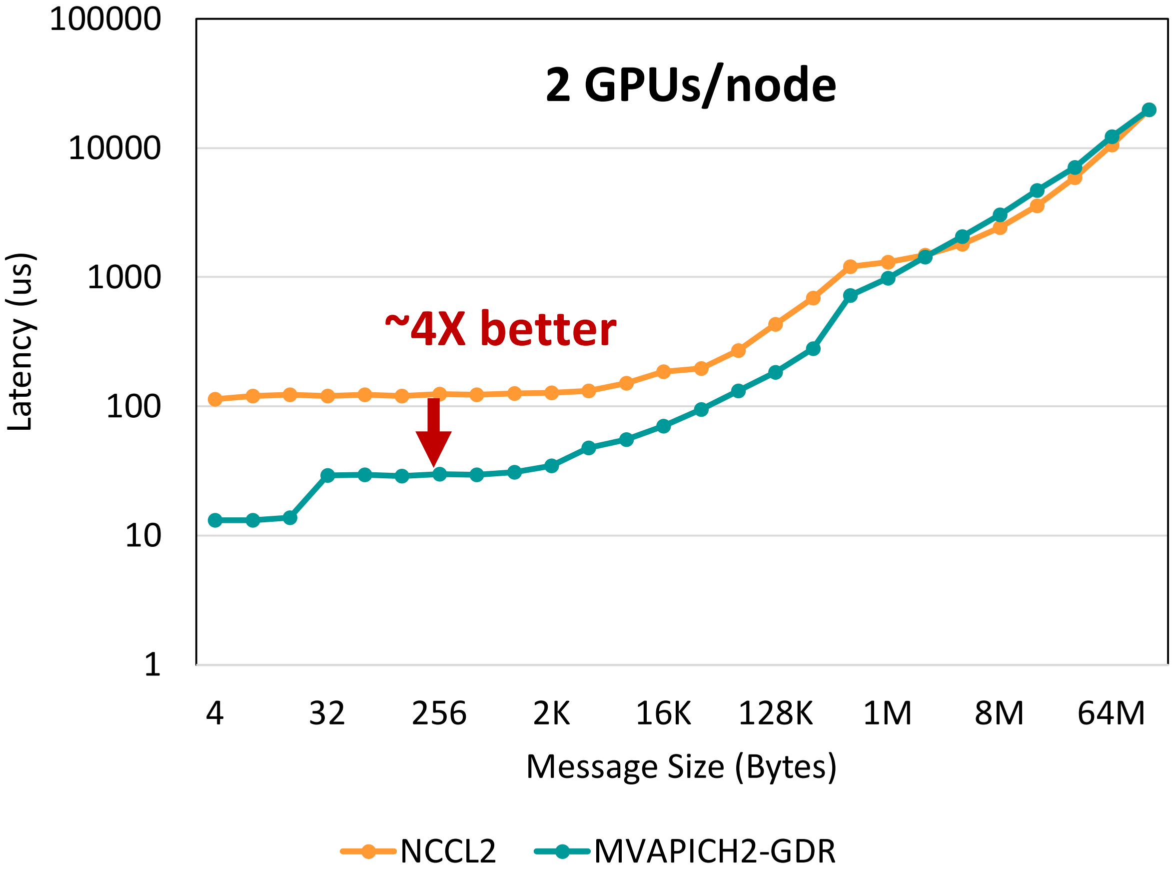


**Available since MVAPICH2-GDR 2.3a*

MVAPICH2-GDR vs. NCCL2 – Broadcast Operation

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



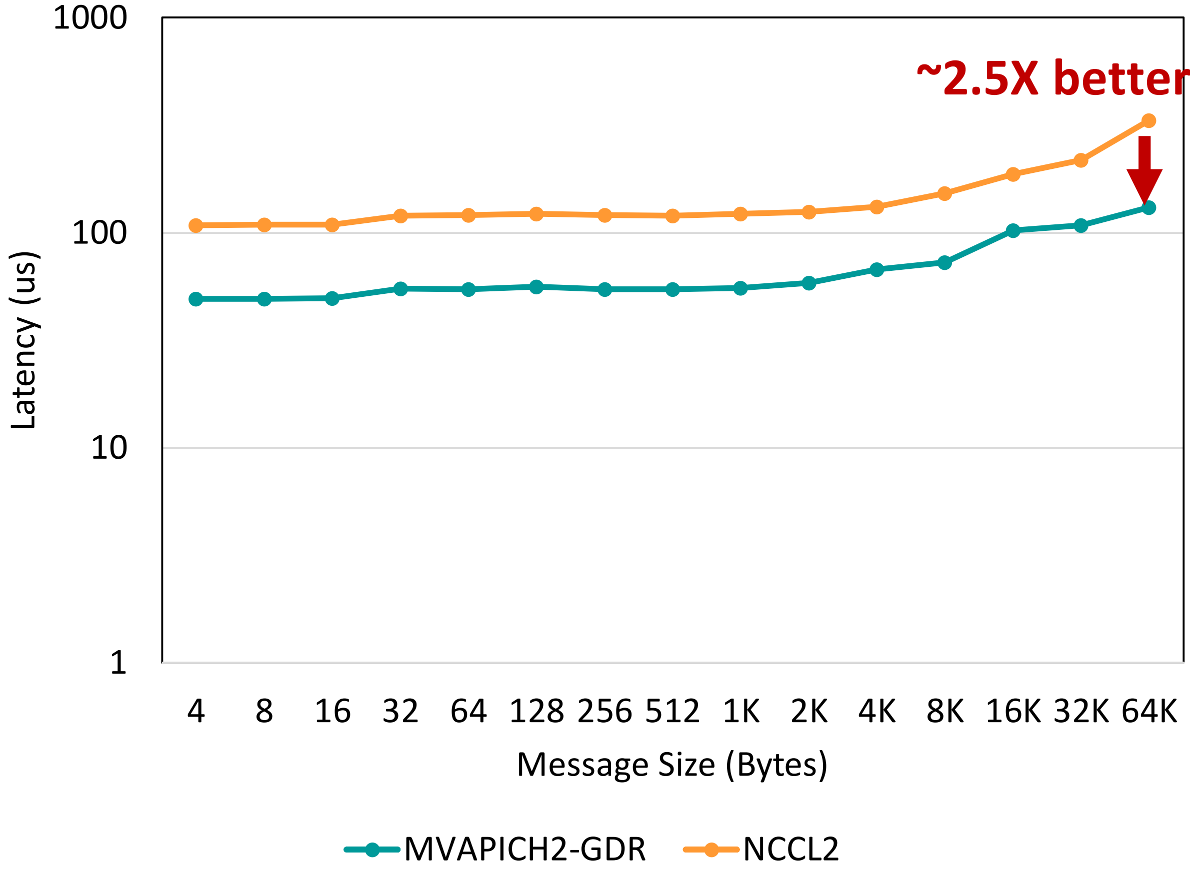


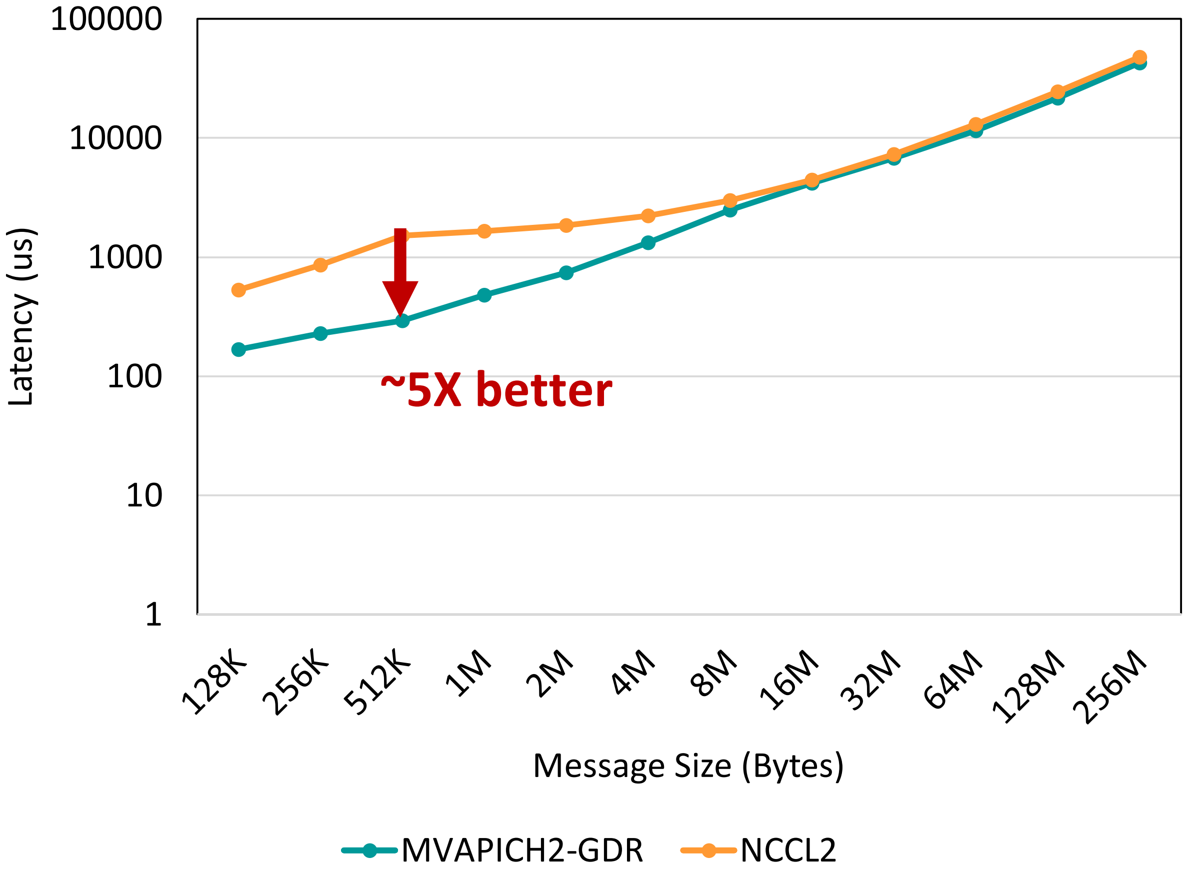
***Will be available with upcoming MVAPICH2-GDR 2.3b**

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

MVAPICH2-GDR vs. NCCL2 – Reduce Operation

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



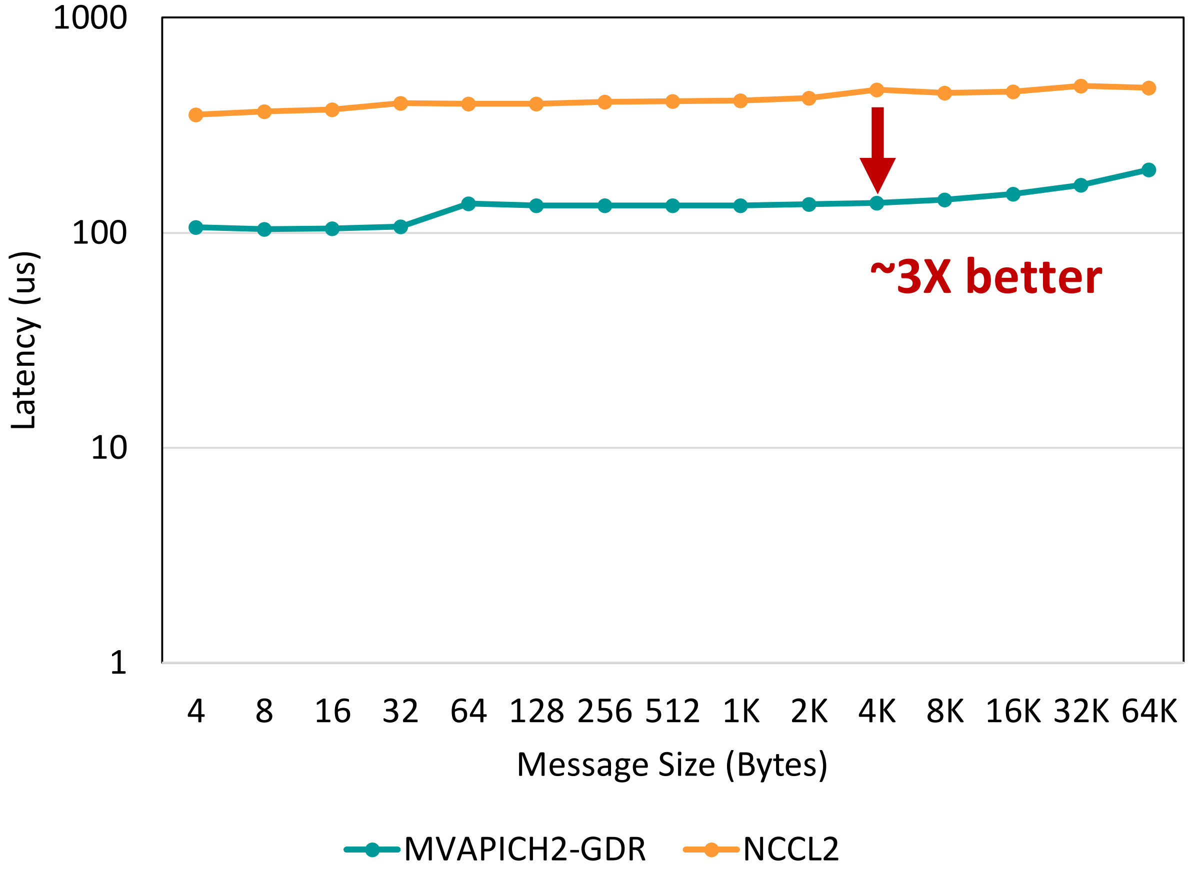


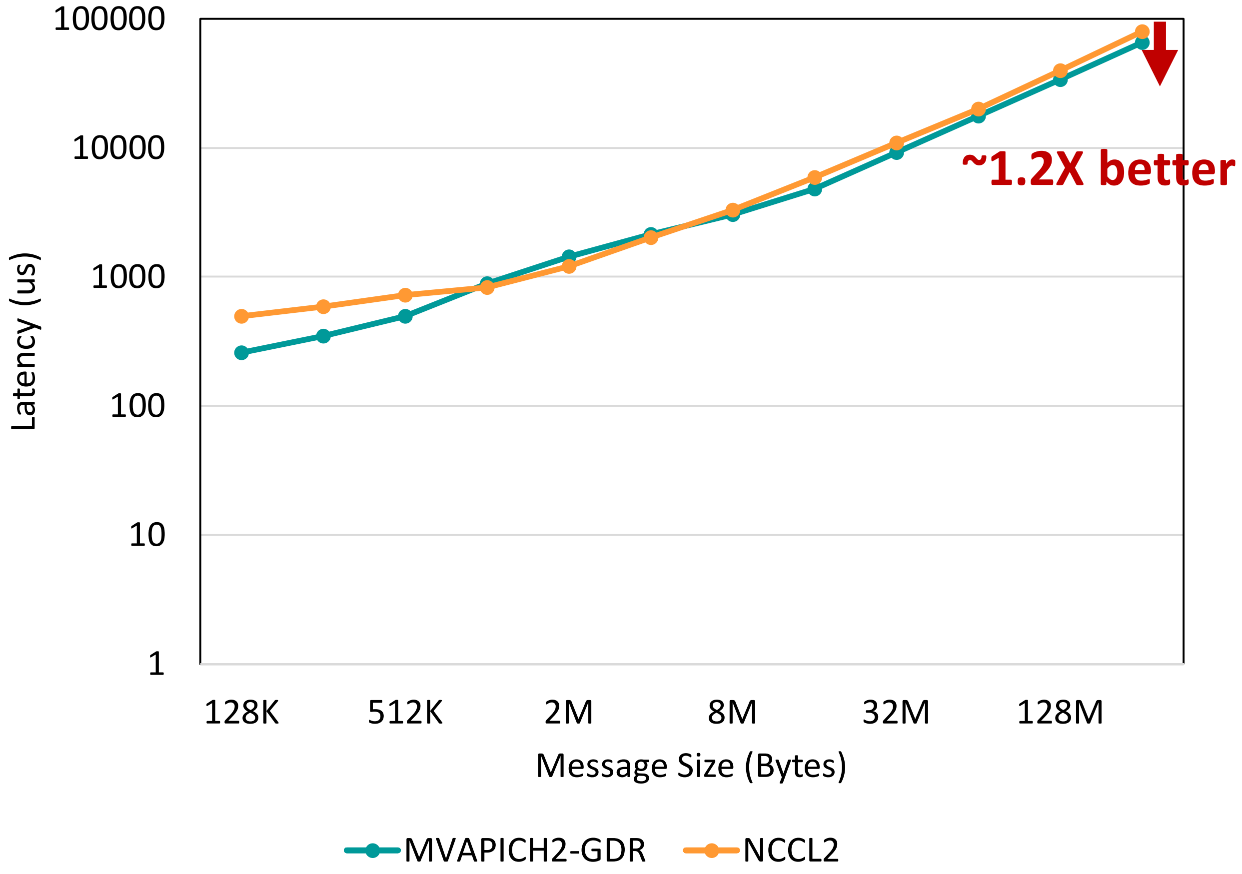
***Will be available with upcoming MVAPICH2-GDR 2.3b**

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

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



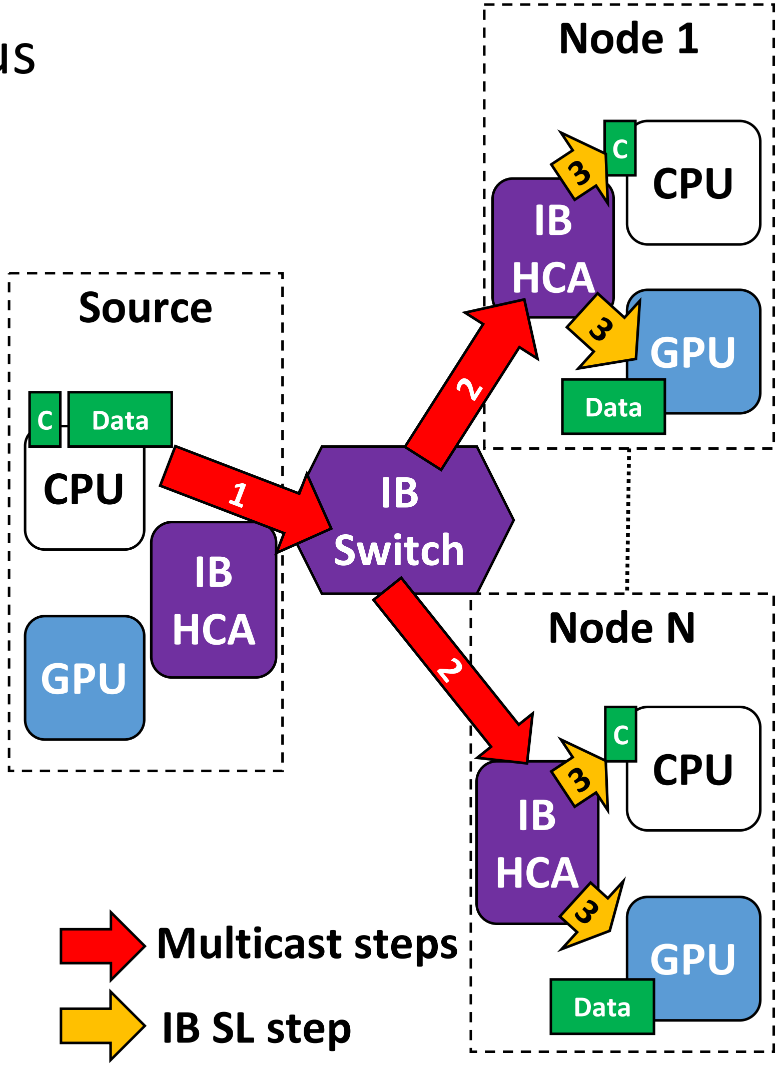


***Will be available with upcoming MVAPICH2-GDR 2.3b**

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

Streaming Support (Combining GDR and IB-Mcast)

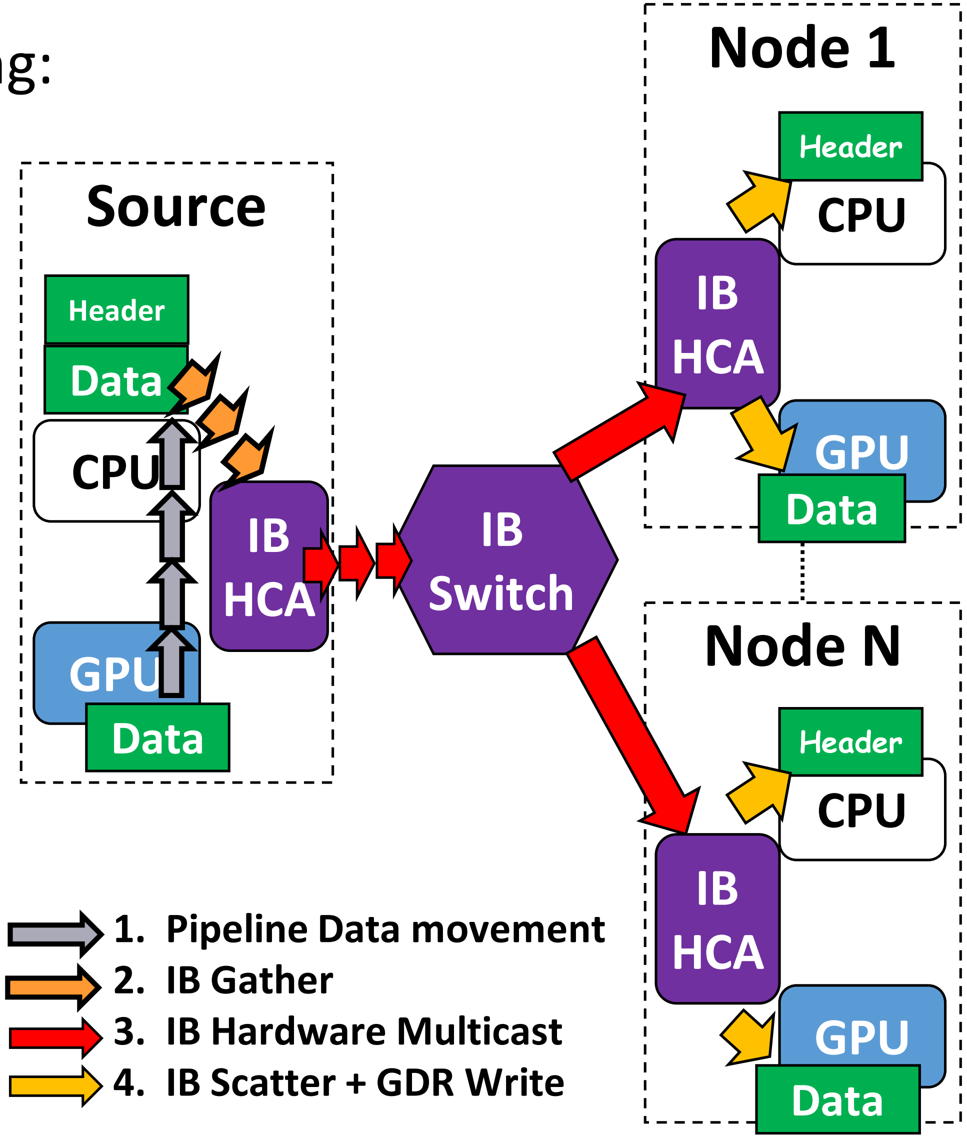
- Combining MCAST+GDR hardware features for heterogeneous configurations:
  - Source on the Host and destination on Device
  - SL design: Scatter at destination
    - Source: Data and Control on Host
    - Destinations: Data on Device and Control on Host
  - Combines IB MCAST and GDR features at receivers
  - CUDA IPC-based topology-aware intra-node broadcast
  - Minimize use of PCIe resources (Maximizing availability of PCIe Host-Device Resources)



C.-H. Chu, K. Hamidouche, H. Subramoni, A. Venkatesh, B. Elton, and D. K. Panda.
"Designing High Performance Heterogeneous Broadcast for Streaming Applications on GPU Clusters, " SBAC-PAD'16, Oct 2016.

Exploiting GDR+IB-Mcast Design for Deep Learning Applications

- Optimizing MCAST+GDR Broadcast for deep learning:
  - Source and destination buffers are on GPU Device
    - Typically very large messages (>1MB)
  - Pipelining data from Device to Host
    - Avoid GDR read limit
    - Leverage high-performance SL design
  - Combines IB MCAST and GDR features
  - Minimize use of PCIe resources on the receiver side
    - Maximizing availability of PCIe Host-Device Resources

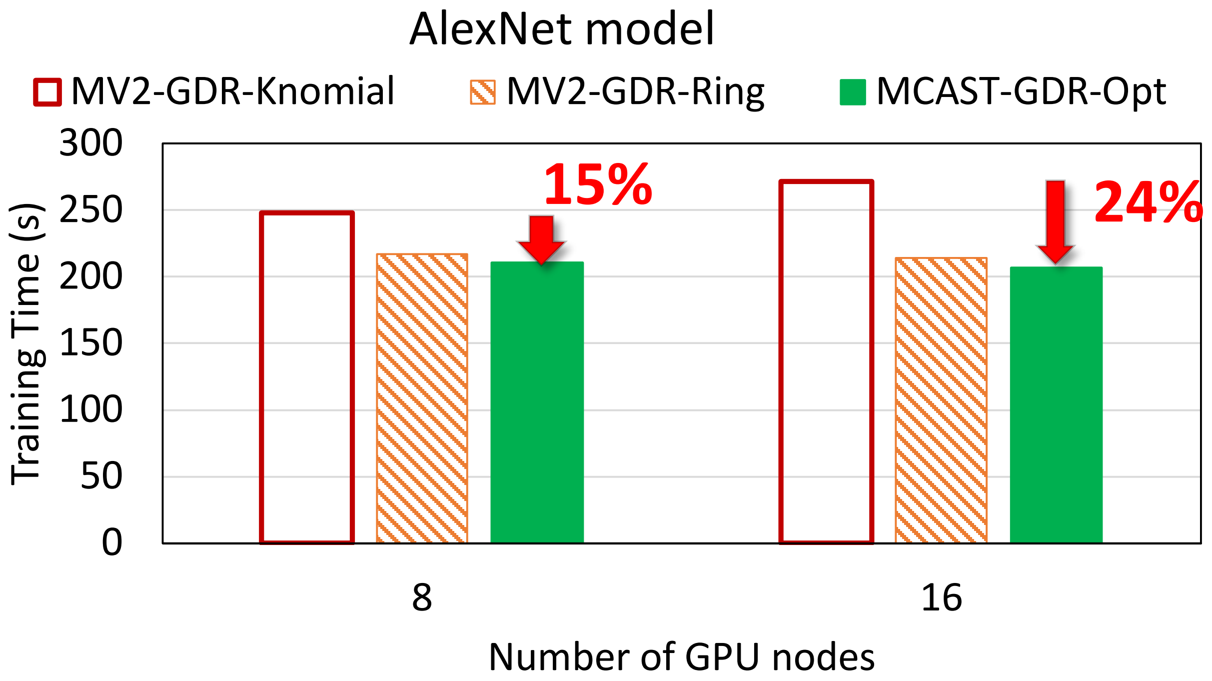


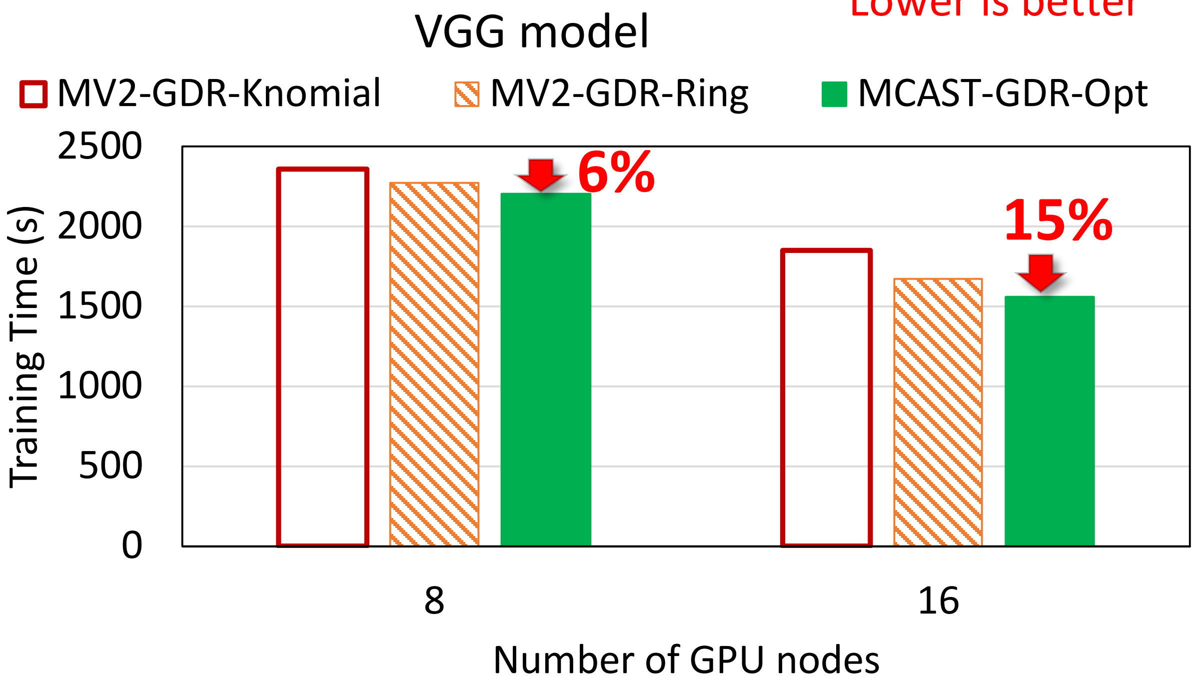
Ching-Hsiang Chu, Xiaoyi Lu, Ammar A. Awan, Hari Subramoni, Jahanzeb Hashmi, Bracy Elton, and Dhabaleswar K. Panda, "Efficient and Scalable Multi-Source Streaming Broadcast on GPU Clusters for Deep Learning , " ICPP'17.

Application Evaluation: Deep Learning Frameworks

- @ RI2 cluster, 16 GPUs, 1 GPU/node
  - Microsoft Cognitive Toolkit (CNTK) *[https://github.com/Microsoft/CNTK]*

Lower is better





- **Reduces up to 24% and 15% of latency for AlexNet and VGG models**
- **Higher improvement can be observed for larger system sizes**

Ching-Hsiang Chu, Xiaoyi Lu, Ammar A. Awan, Hari Subramoni, Jahanzeb Hashmi, Bracy Elton, and Dhabaleswar K. Panda, "Efficient and Scalable Multi-Source Streaming Broadcast on GPU Clusters for Deep Learning , " ICPP'17.

MVAPICH2 Software Family

Requirements	Library
MPI with IB, iWARP, Omni-Path, and RoCE	MVAPICH2
Advanced MPI Features/Support, OSU INAM, PGAS and MPI+PGAS with IB, Omni-Path, and RoCE	MVAPICH2-X
MPI with IB, RoCE & GPU and Support for Deep Learning	MVAPICH2-GDR
HPC Cloud with MPI & IB	MVAPICH2-Virt
Energy-aware MPI with IB, iWARP and RoCE	MVAPICH2-EA
MPI Energy Monitoring Tool	OEMT
InfiniBand Network Analysis and Monitoring	OSU INAM
Microbenchmarks for Measuring MPI and PGAS Performance	OMB

Can HPC and Virtualization be Combined?

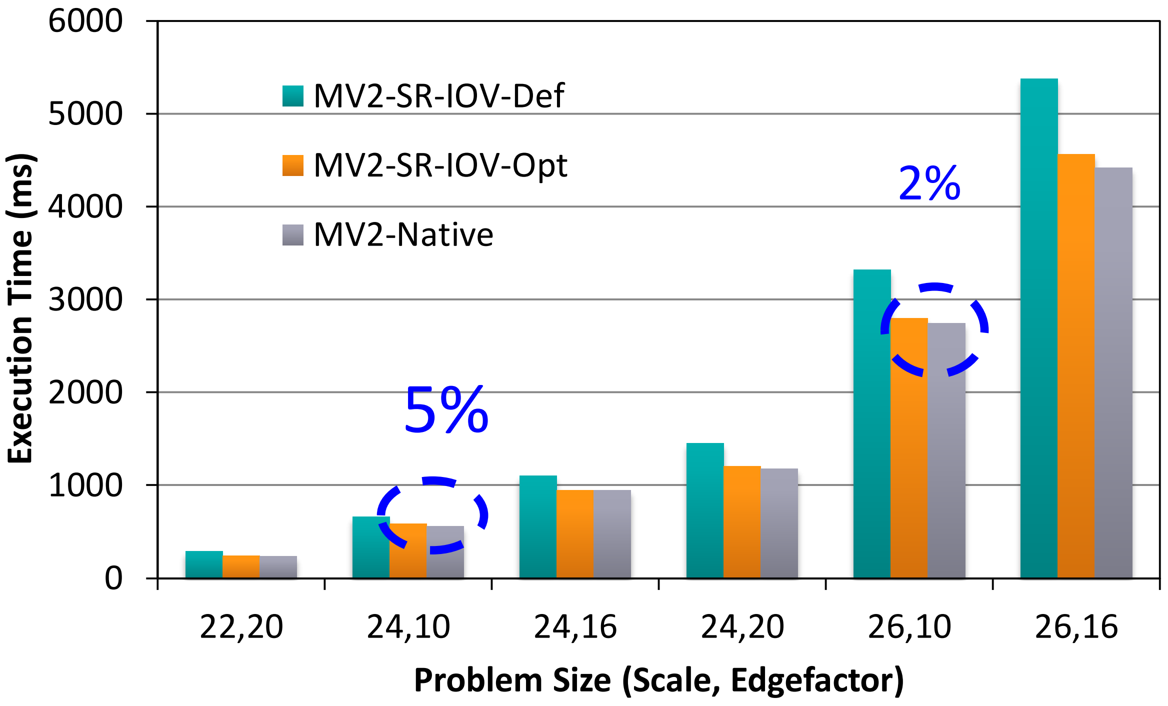
- Virtualization has many benefits
  - Fault-tolerance
  - Job migration
  - Compaction
- Have not been very popular in HPC due to overhead associated with Virtualization
- New SR-IOV (Single Root – IO Virtualization) support available with Mellanox InfiniBand adapters changes the field
- Enhanced MVAPICH2 support for SR-IOV
- MVAPICH2-Virt 2.2 supports:
  - Efficient MPI communication over SR-IOV enabled InfiniBand networks
  - High-performance and locality-aware MPI communication for VMs and containers
  - Automatic communication channel selection for VMs (SR-IOV, IVSHMEM, and CMA/LiMIC2) and containers (IPC-SHM, CMA, and HCA)
  - OpenStack, Docker, and Singularity

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

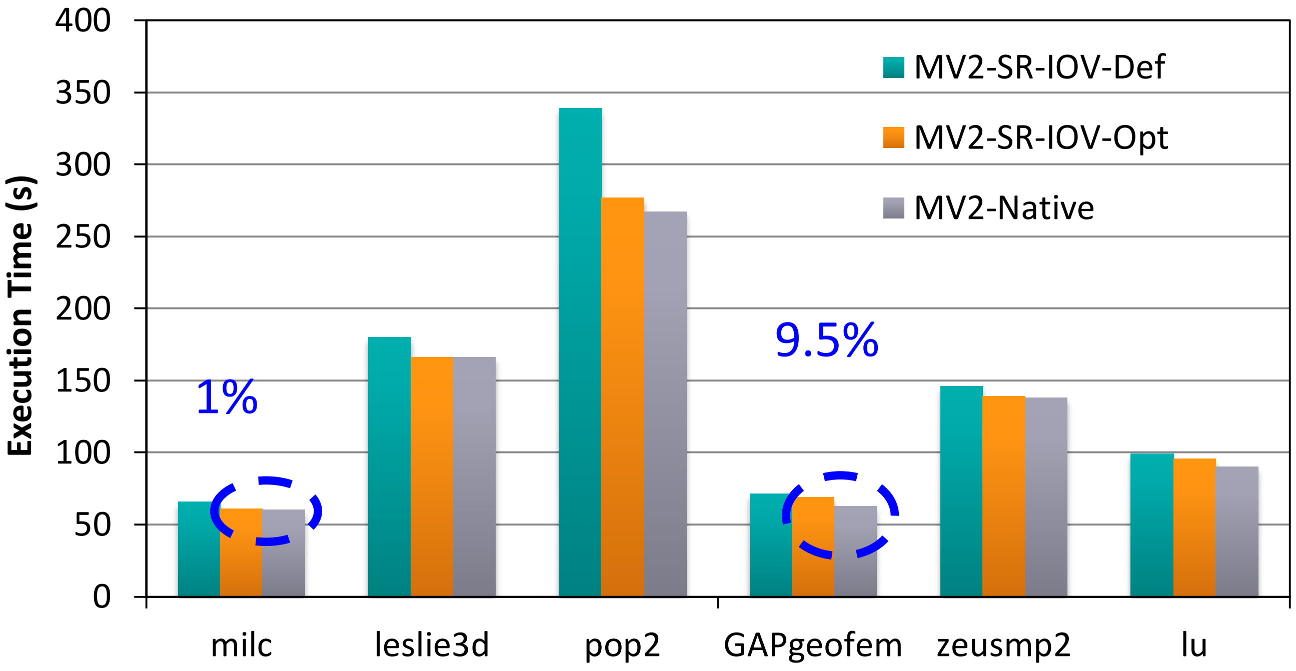
J. Zhang, X. Lu, J. Jose, M. Li, R. Shi and D.K. Panda, High Performance MPI Libray over SR-IOV enabled InfiniBand Clusters, HiPC'14

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

Application-Level Performance on Chameleon (SR-IOV Support)



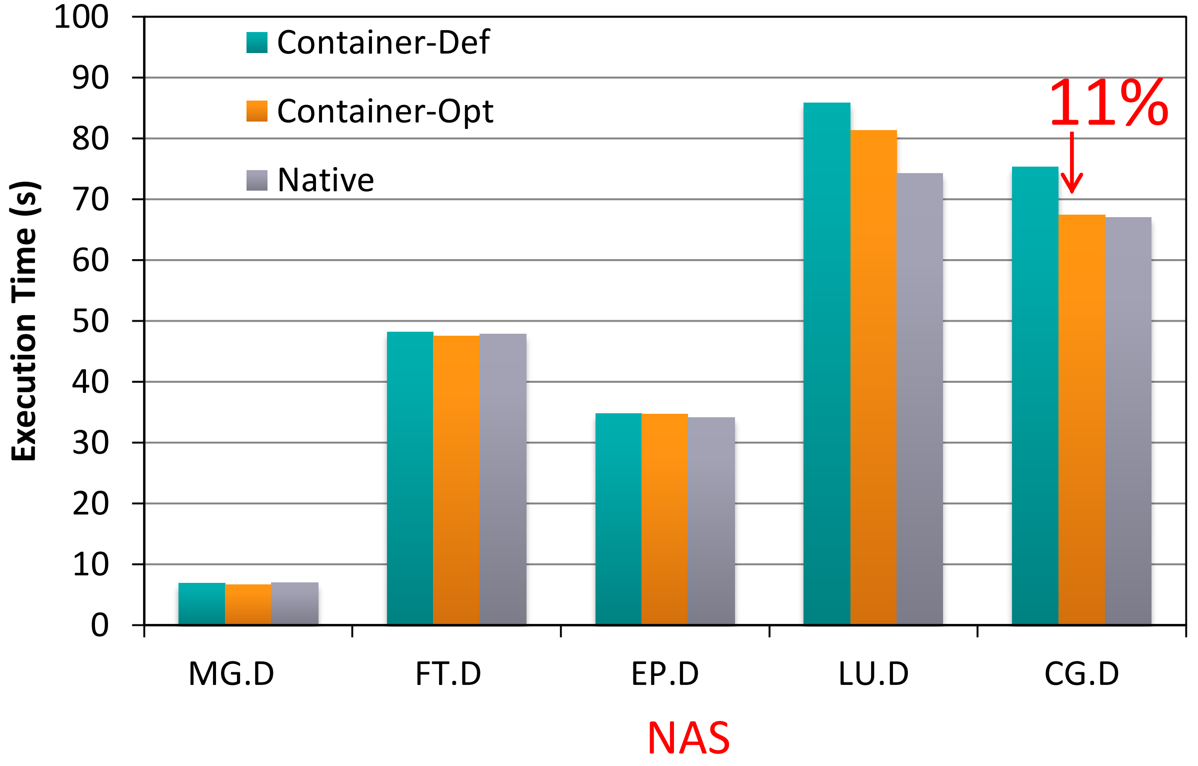
Graph500

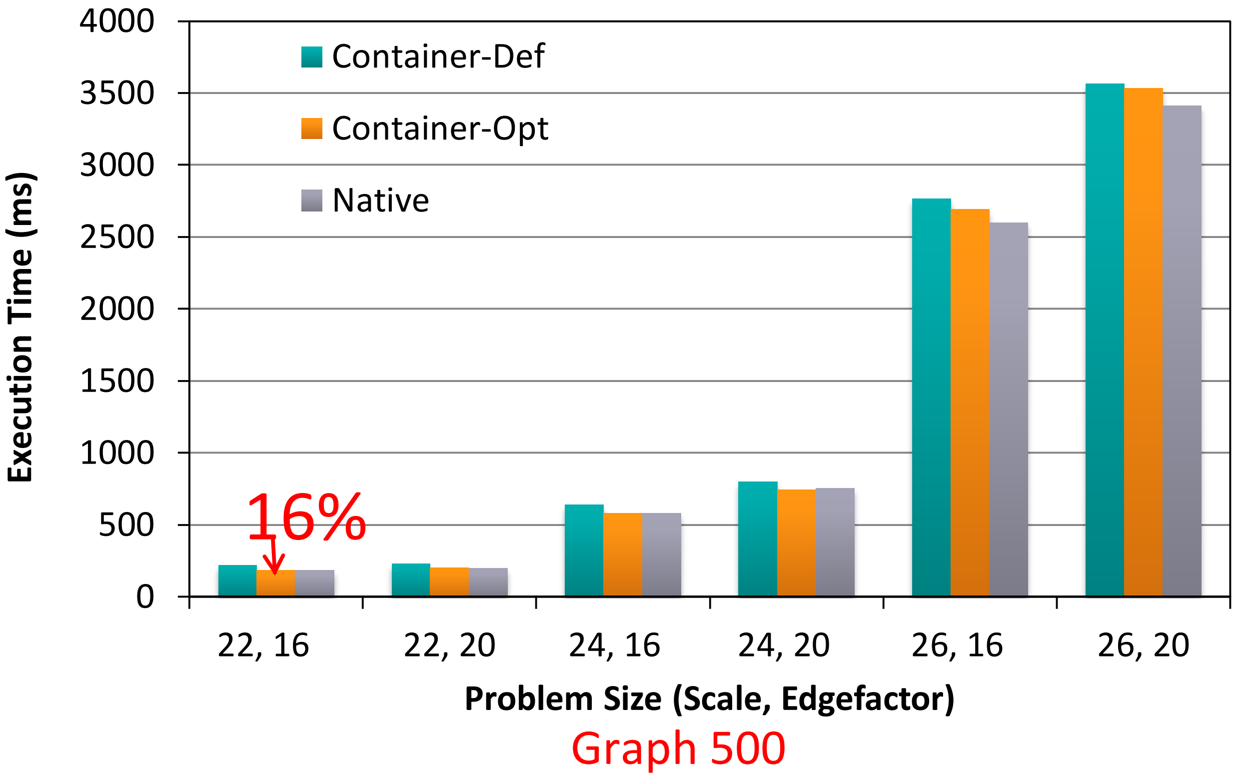


SPEC MPI2007

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

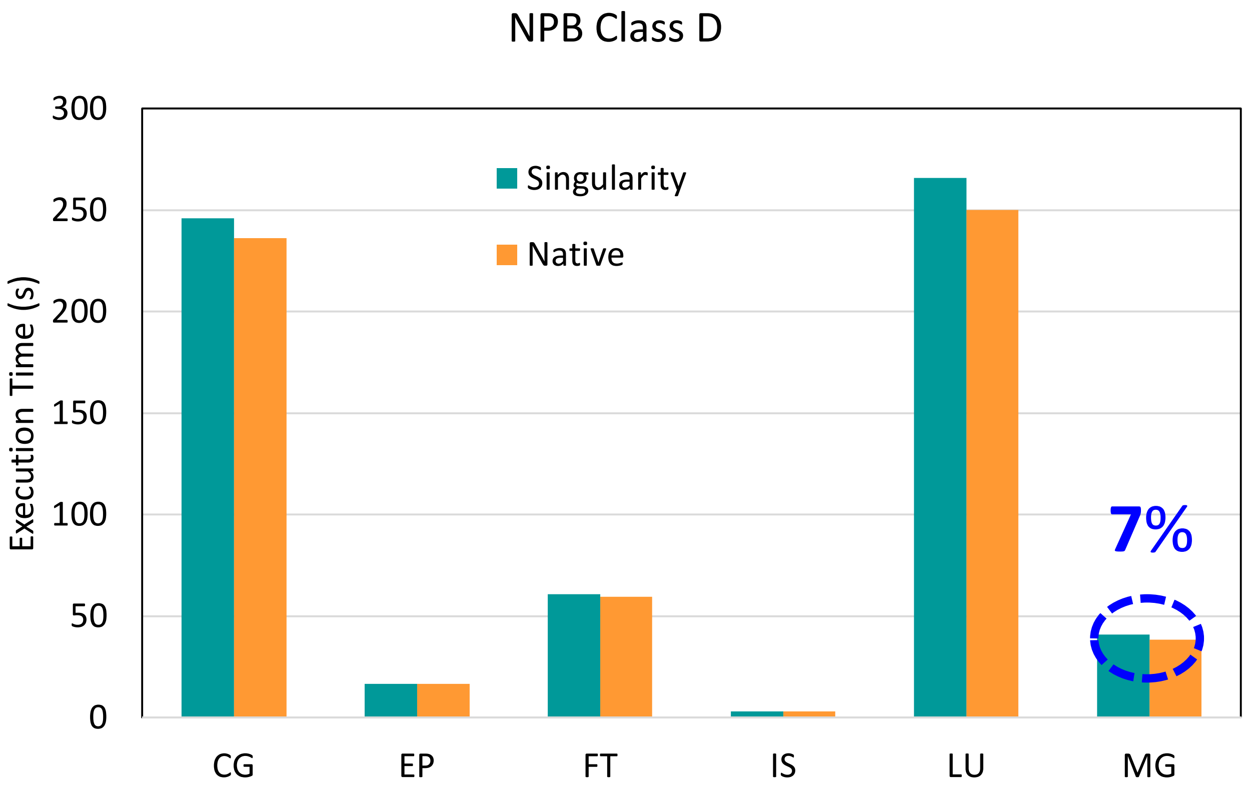
Application-Level Performance on Chameleon (Containers Support)

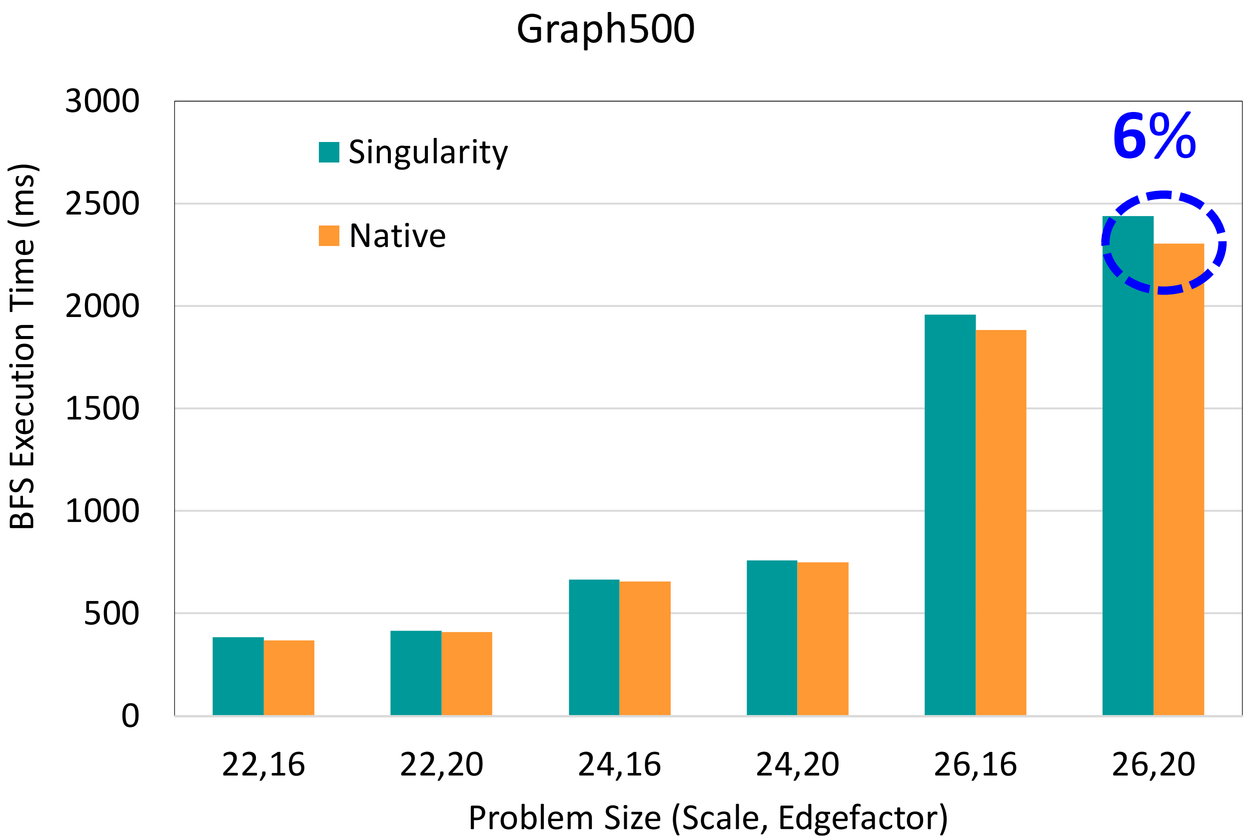




- 64 Containers across 16 nodes, pining 4 Cores per Container
- Compared to Container-Def, up to 11% and 16% of execution time reduction for NAS and Graph 500
- Compared to Native, less than 9 % and 4% overhead for NAS and Graph 500

Application-Level Performance on Singularity with MVAPICH2





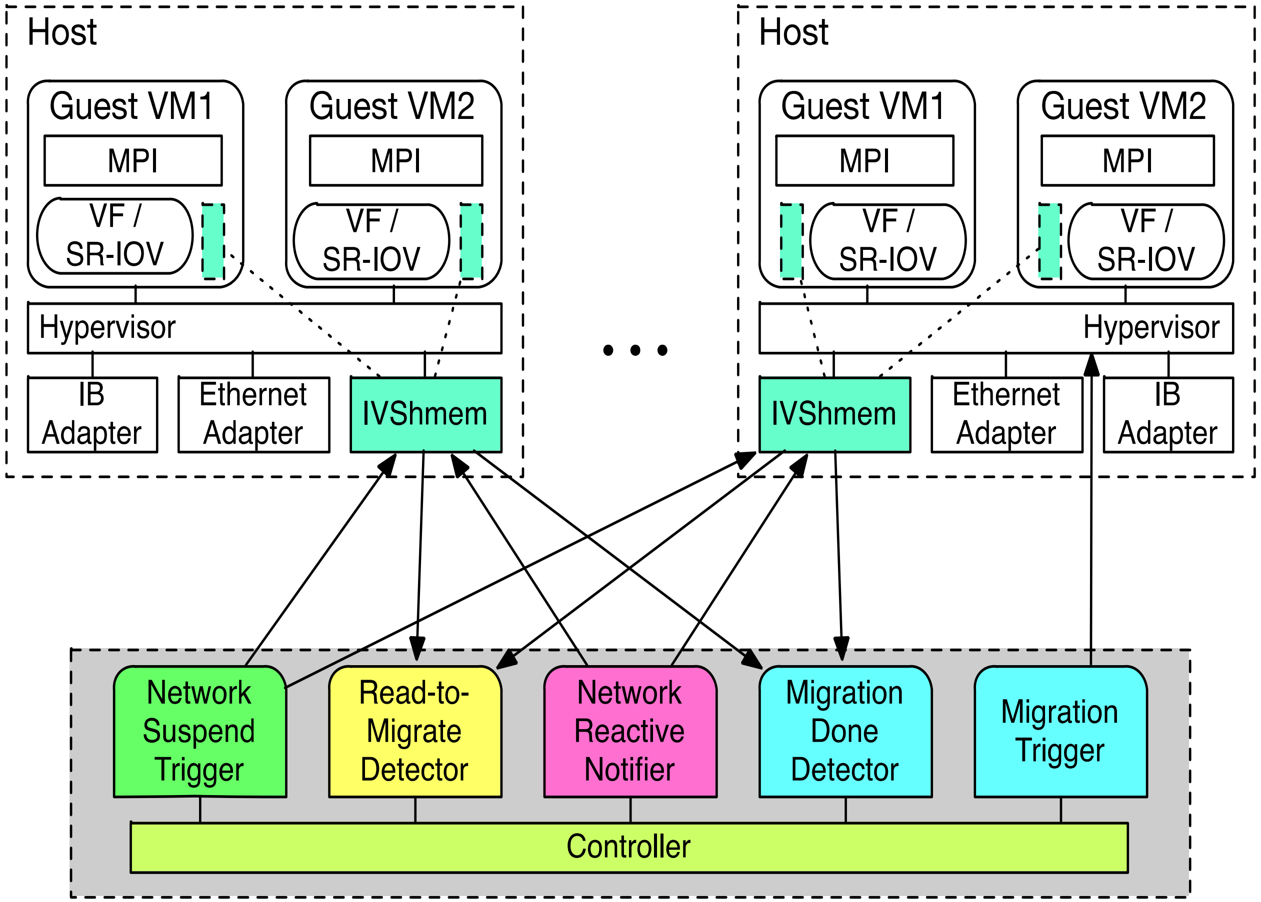
- 512 Processes across 32 nodes
- Less than 7% and 6% overhead for NPB and Graph500, respectively

More Details were presented in yesterday's Tutorial Session

MVAPICH2-Virt Upcoming Features

- Support for SR-IOV Enabled VM Migration
- SR-IOV and IVSHMEM Support in SLURM
- Support for Microsoft Azure Platform

High Performance SR-IOV enabled VM Migration Support in MVAPICH2



- Migration with SR-IOV device has to handle the challenges of detachment/re-attachment of virtualized IB device and IB connection
- Consist of SR-IOV enabled IB Cluster and External Migration Controller
- Multiple parallel libraries to notify MPI applications during migration (detach/reattach SR-IOV/IVShmem, migrate VMs, migration status)
- Handle the IB connection suspending and reactivating
- Propose Progress engine (PE) and migration thread based (MT) design to optimize VM migration and MPI application performance

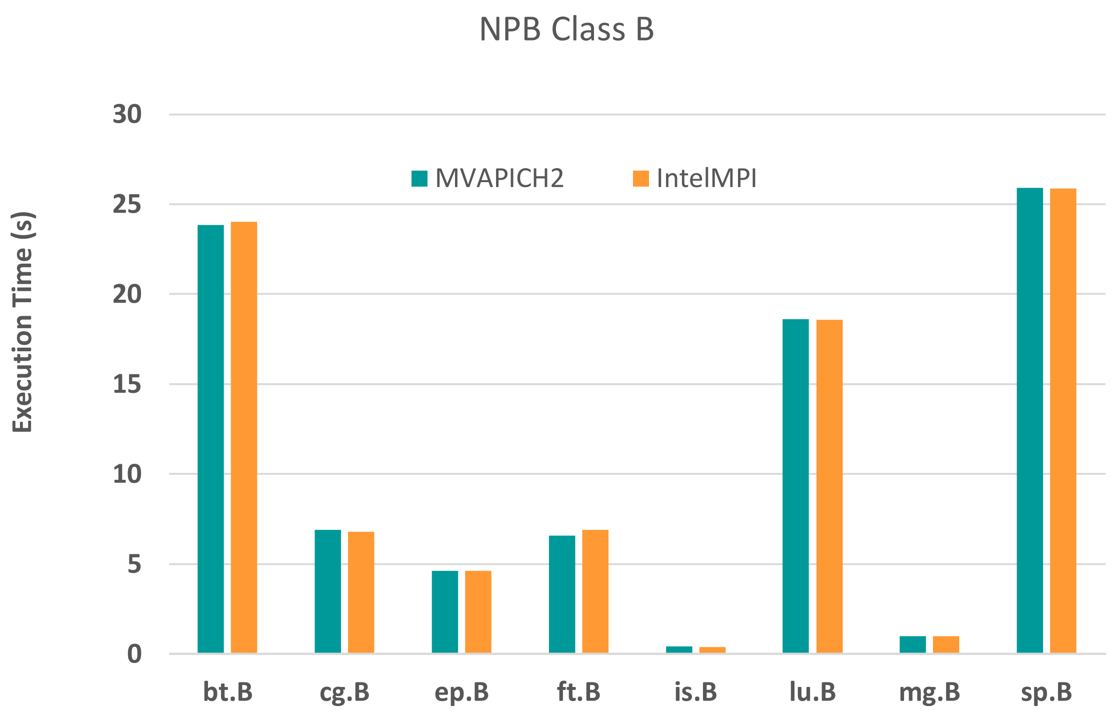
J. Zhang, X. Lu, D. K. Panda. High-Performance Virtual Machine Migration Framework for MPI Applications on SR-IOV enabled InfiniBand Clusters. IPDPS, 2017

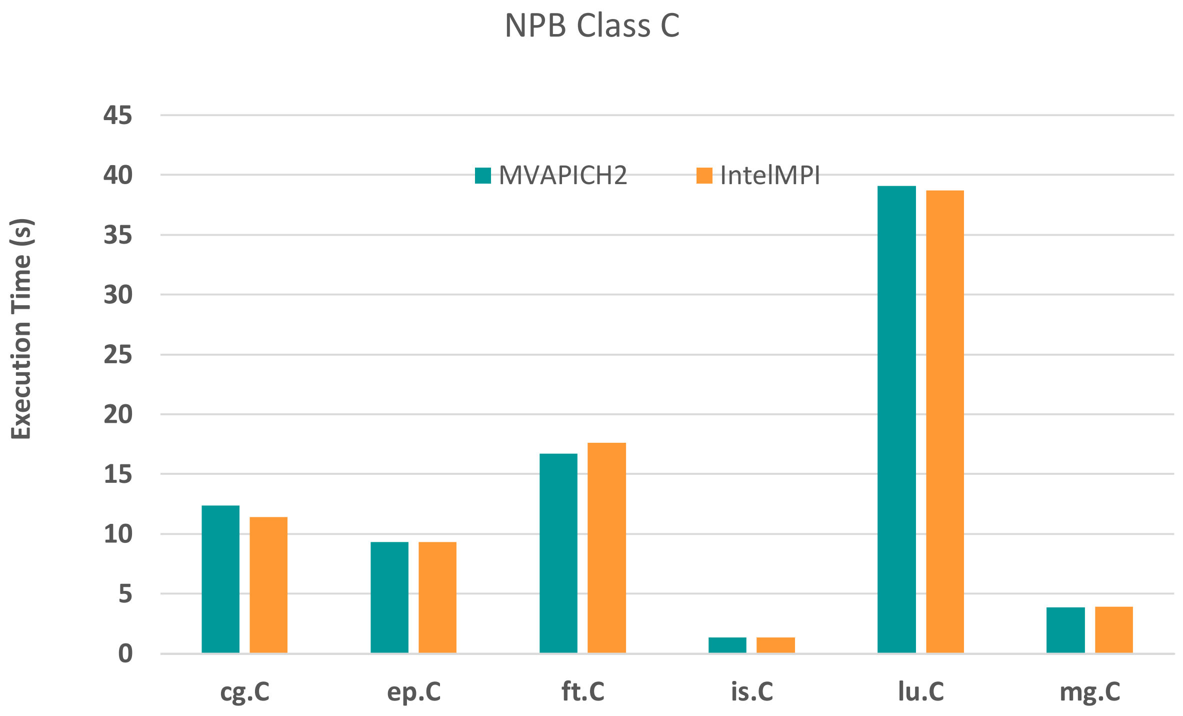
SR-IOV and IVSHMEM in SLURM

- Requirement of managing and isolating virtualized resources of SR-IOV and IVSHMEM
- Such kind of management and isolation is hard to be achieved by MPI library alone, but much easier with SLURM
- Efficient running MPI applications on HPC Clouds needs SLURM to support managing SR-IOV and IVSHMEM
  - Can critical HPC resources be efficiently shared among users by extending SLURM with support for SR-IOV and IVSHMEM based virtualization?
  - Can SR-IOV and IVSHMEM enabled SLURM and MPI library provide bare-metal performance for end applications on HPC Clouds?

J. Zhang, X. Lu, S. Chakraborty, and D. K. Panda. SLURM-V: Extending SLURM for Building Efficient HPC Cloud with SR-IOV and IVShmem. The 22nd International European Conference on Parallel Processing (Euro-Par), 2016.

Application-Level Performance on Azure





- NPB Class B with 16 Processes on 2 Azure A8 instances
- NPB Class C with 32 Processes on 4 Azure A8 instances
- Comparable performance between MVAPICH2 and Intel MPI

Will be available publicly for Azure soon

MVAPICH2 Software Family

Requirements	Library
MPI with IB, iWARP, Omni-Path, and RoCE	MVAPICH2
Advanced MPI Features/Support, OSU INAM, PGAS and MPI+PGAS with IB, Omni-Path, and RoCE	MVAPICH2-X
MPI with IB, RoCE & GPU and Support for Deep Learning	MVAPICH2-GDR
HPC Cloud with MPI & IB	MVAPICH2-Virt
Energy-aware MPI with IB, iWARP and RoCE	MVAPICH2-EA
MPI Energy Monitoring Tool	OEMT
InfiniBand Network Analysis and Monitoring	OSU INAM
Microbenchmarks for Measuring MPI and PGAS Performance	OMB

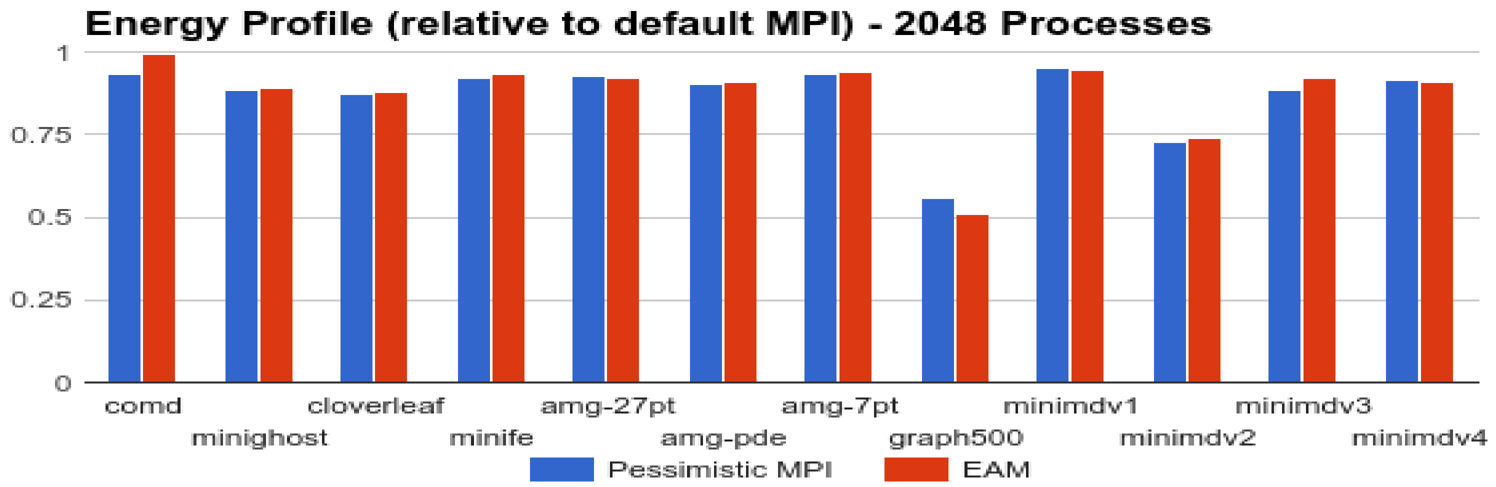
Energy-Aware MVAPICH2 & OSU Energy Management Tool (OEMT)

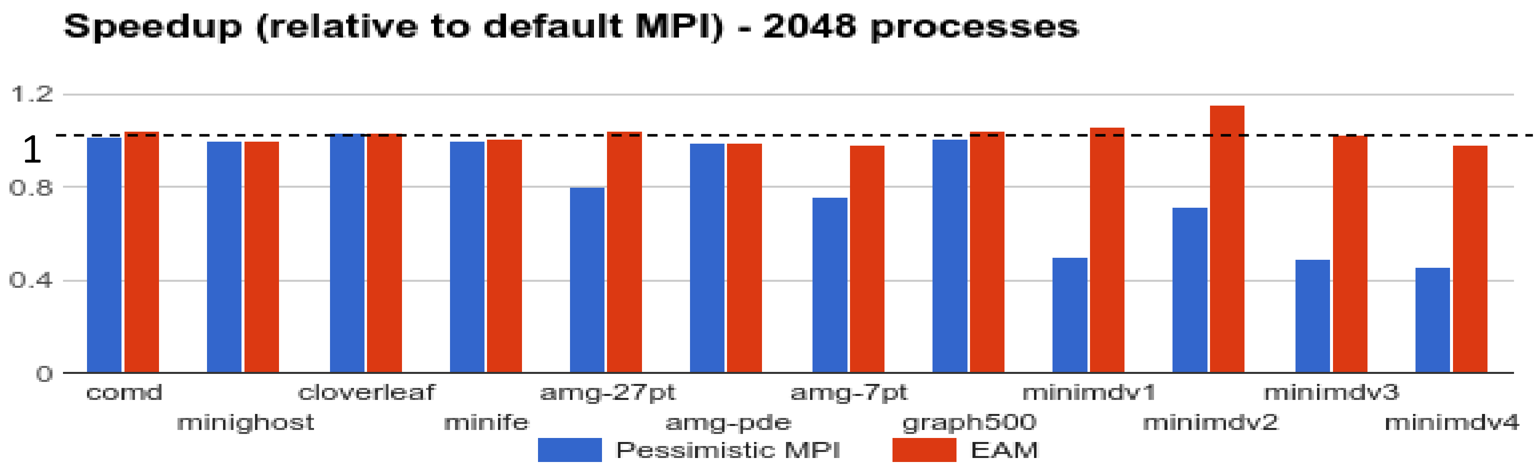
- MVAPICH2-EA 2.1 (Energy-Aware)
  - A white-box approach
  - New Energy-Efficient communication protocols for pt-pt and collective operations
  - Intelligently apply the appropriate Energy saving techniques
  - Application oblivious energy saving

- OEMT
  - A library utility to measure energy consumption for MPI applications
  - Works with all MPI runtimes
  - PRELOAD option for precompiled applications
  - Does not require ROOT permission:
    - A safe kernel module to read only a subset of MSRs

MVAPICH2-EA: Application Oblivious Energy-Aware-MPI (EAM)

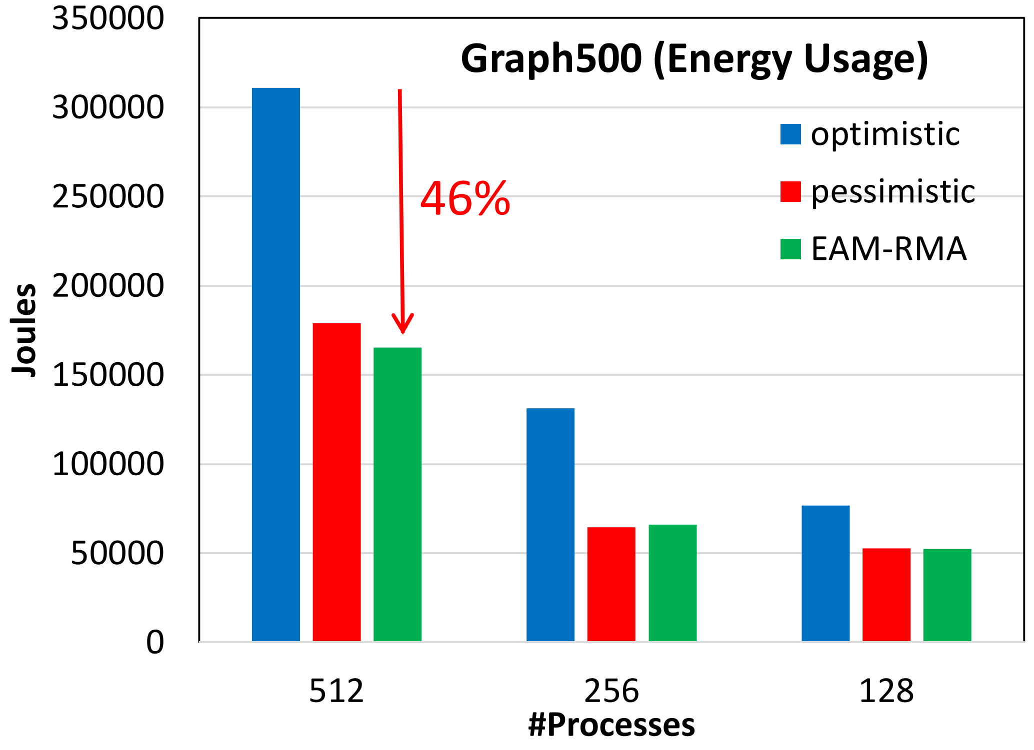
- An energy efficient runtime that provides energy savings without application knowledge
- Uses automatically and transparently the best energy lever
- Provides guarantees on maximum degradation with 5-41% savings at <= 5% degradation
- Pessimistic MPI applies energy reduction lever to each MPI call

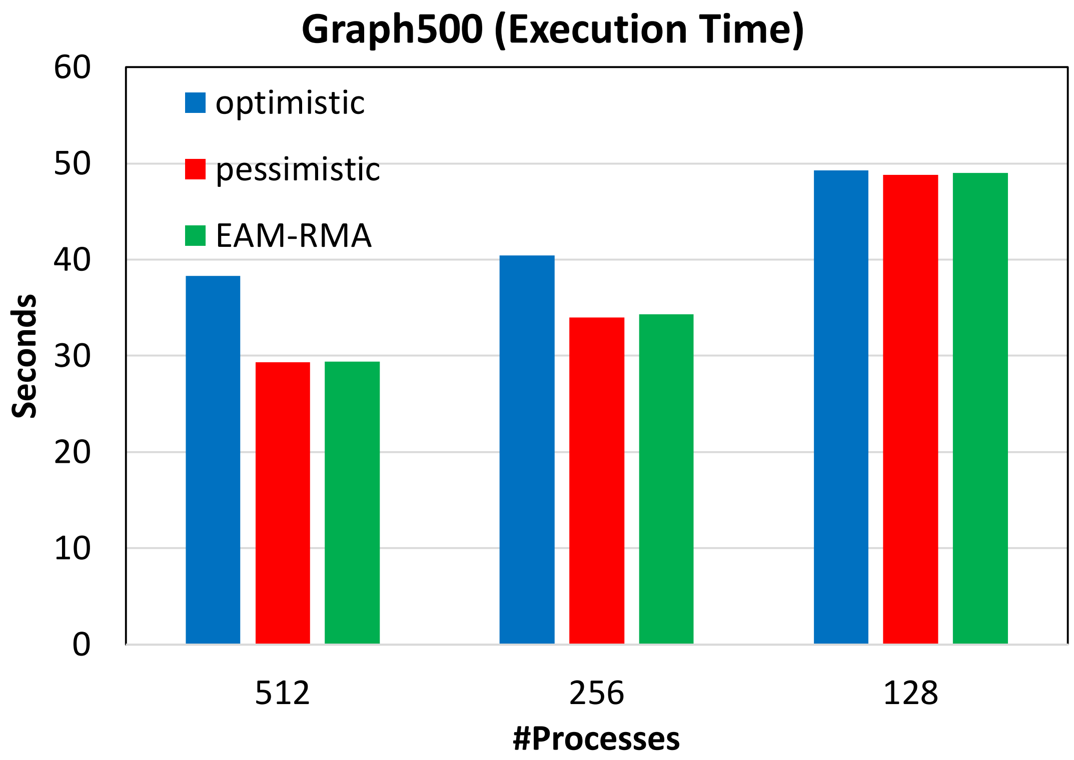




A Case for Application-Oblivious Energy-Efficient MPI Runtime A. Venkatesh, A. Vishnu, K. Hamidouche, N. Tallent, D. K. Panda, D. Kerbyson, and A. Hoise, Supercomputing '15, Nov 2015 ***[Best Student Paper Finalist]***

MPI-3 RMA Energy Savings with Proxy-Applications





- MPI_Win_fence dominates application execution time in graph500
- Between 128 and 512 processes, EAM-RMA yields between 31% and 46% savings with no degradation in execution time in comparison with the default optimistic MPI runtime
- MPI-3 RMA Energy-efficient support will be available in upcoming MVAPICH2-EA release

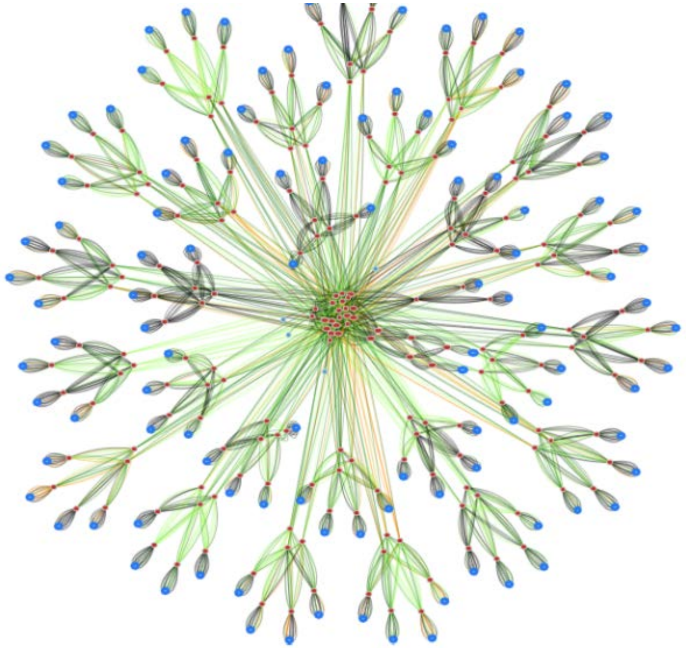
MVAPICH2 Software Family

Requirements	Library
MPI with IB, iWARP, Omni-Path, and RoCE	MVAPICH2
Advanced MPI Features/Support, OSU INAM, PGAS and MPI+PGAS with IB, Omni-Path, and RoCE	MVAPICH2-X
MPI with IB, RoCE & GPU and Support for Deep Learning	MVAPICH2-GDR
HPC Cloud with MPI & IB	MVAPICH2-Virt
Energy-aware MPI with IB, iWARP and RoCE	MVAPICH2-EA
MPI Energy Monitoring Tool	OEMT
InfiniBand Network Analysis and Monitoring	OSU INAM
Microbenchmarks for Measuring MPI and PGAS Performance	OMB

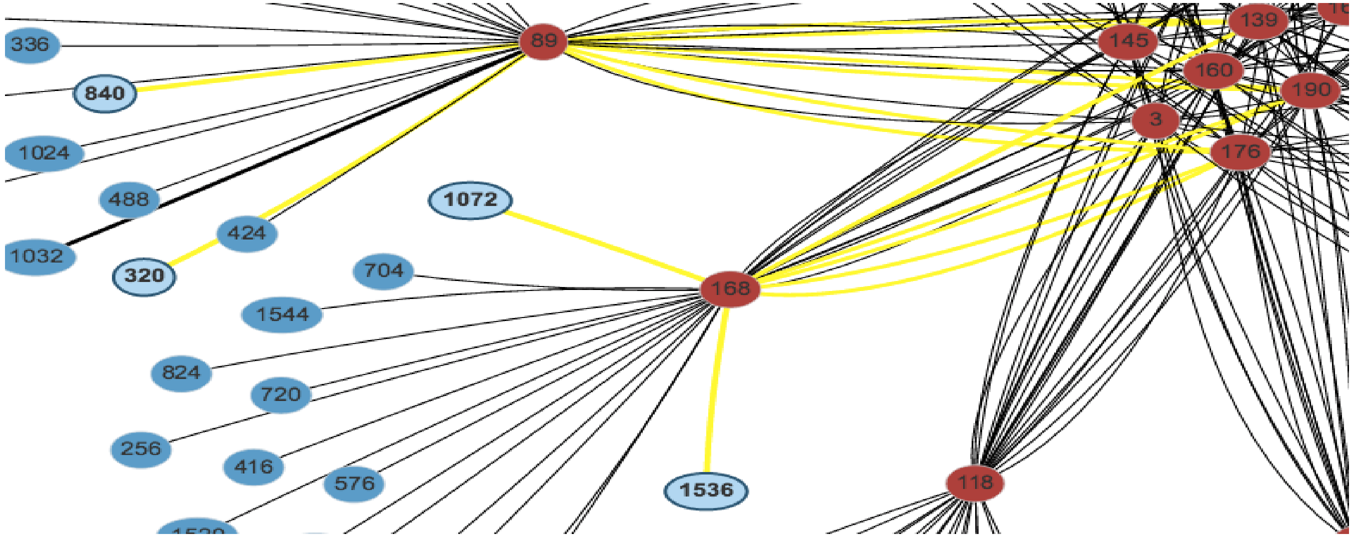
Overview of OSU INAM

- 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 v0.9.3 released on 03/16/2018
  – Enhance INAMD to query end nodes based on command line option
  – Add a web page to display size of the database in real-time
  – Enhance interaction between the web application and SLURM job launcher for increased portability
  – Improve packaging of web application and daemon to ease installation

OSU INAM Features



Comet@SDSC --- Clustered View

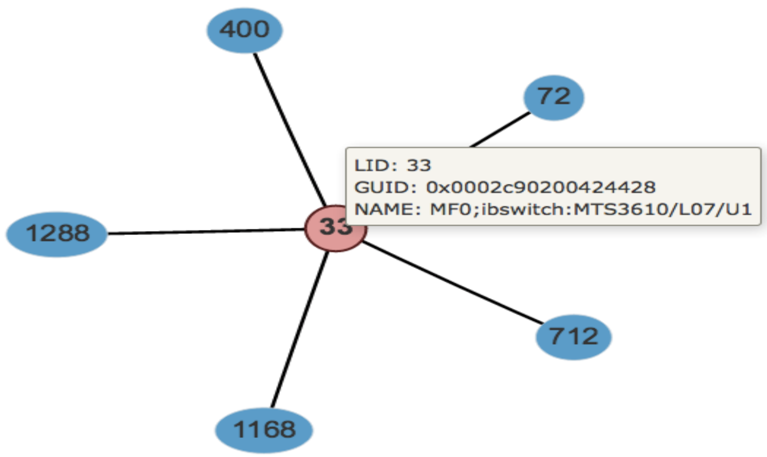


Finding Routes Between Nodes

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

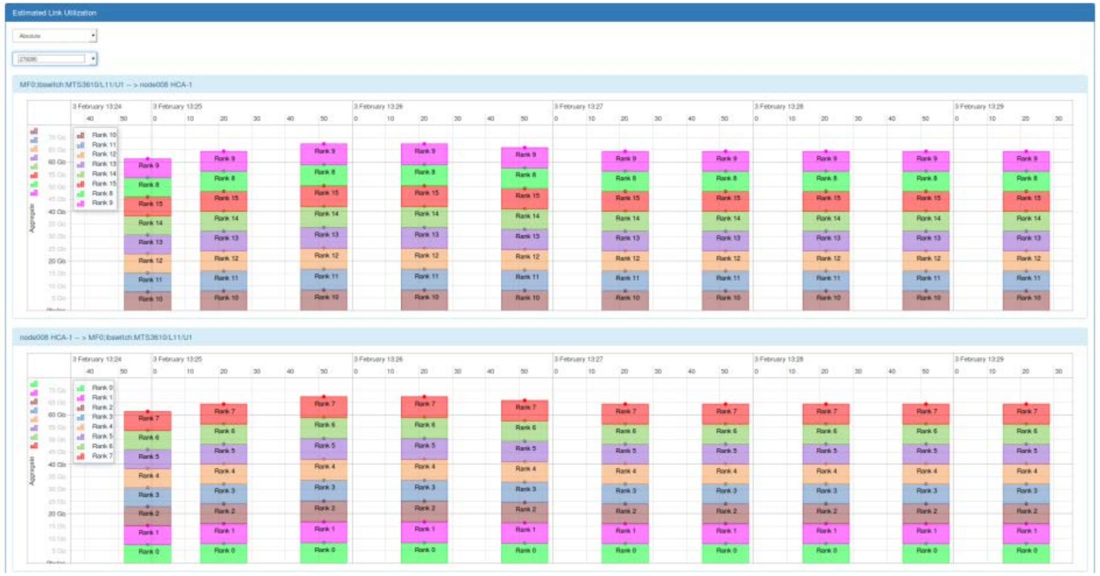
- Show network topology of large clusters
- Visualize traffic pattern on different links
- Quickly identify congested links/links in error state
- See the history unfold – play back historical state of the network

OSU INAM Features (Cont.)



Visualizing a Job (5 Nodes)

- 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 Process Level Link Utilization

- 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 Poster: High-Performance and Scalable Fabric Analysis, Monitoring and Introspection Infrastructure for HPC - Hari Subramoni

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
- Soliciting additional contributions, send your results to mvapich-help at cse.ohio-state.edu.
- We will link these results with credits to you.

MVAPICH Team Part of New NSF Tier-1 System

- TACC and MVAPICH partner to win new NSF Tier-1 System
  - https://www.hpcwire.com/2018/07/30/tacc-wins-next-nsf-funded-major-supercomputer/
- The MVAPICH team will be an integral part of the effort
- An overview of the recently awarded NSF Tier 1 System at TACC will be presented by Dan Stanzione
  - The presentation will also include discussion on MVAPICH collaboration on past systems and this upcoming system at TACC

Commercial Support for MVAPICH2 Libraries

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

MVAPICH2 – Plans for Exascale

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

Thank You!

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

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