MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

High-Performance Deep Learning

High-Performance Big Data

Designing HPC, Big Data, Deep Learning, and Cloud Middleware for Exascale Systems: Challenges and Opportunities

Keynote Talk at SEA Symposium (April '18)

by

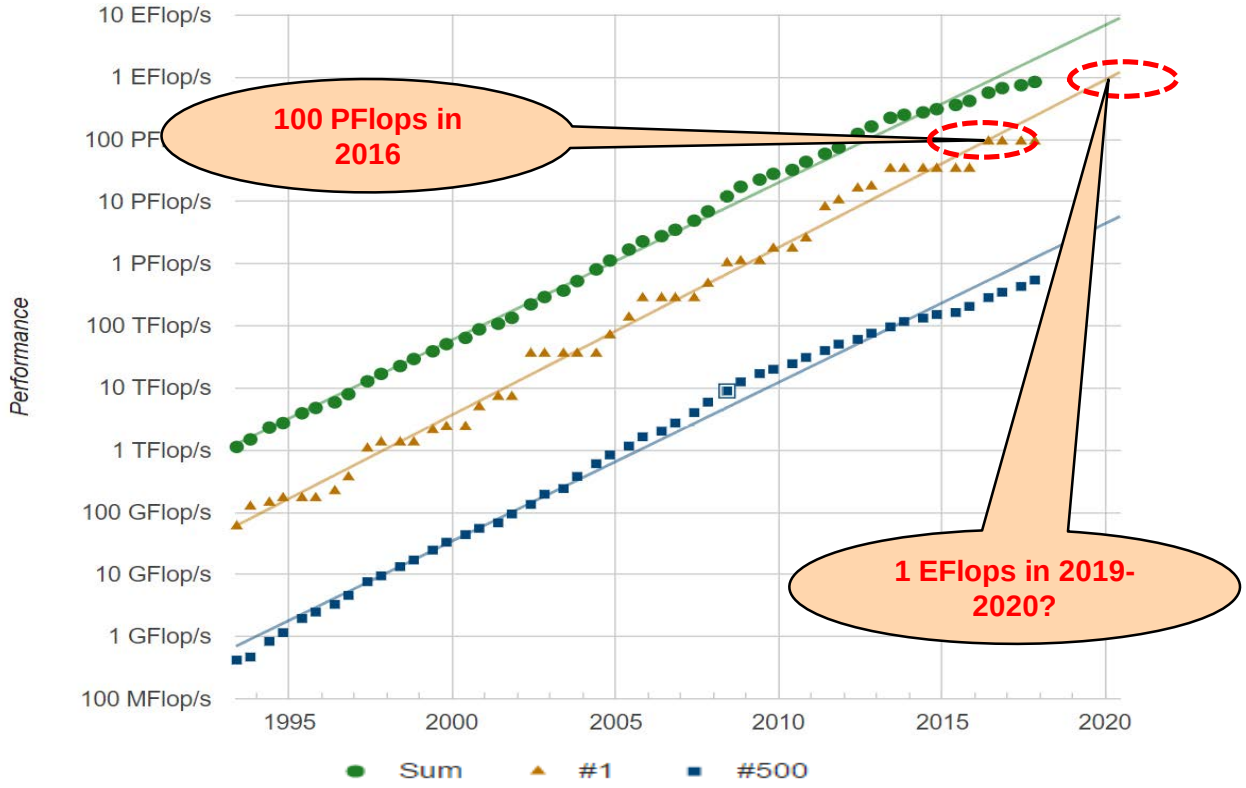
Dhabaleswar K. (DK) Panda

The Ohio State University

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

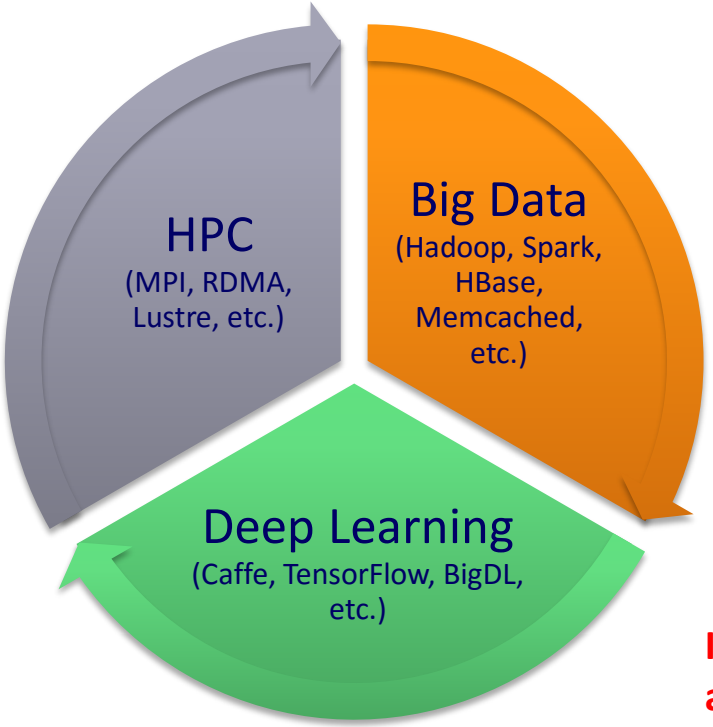
http://www.cse.ohio-state.edu/~panda

High-End Computing (HEC): Towards Exascale



Expected to have an ExaFlop system in 2019-2020!

Increasing Usage of HPC, Big Data and Deep Learning



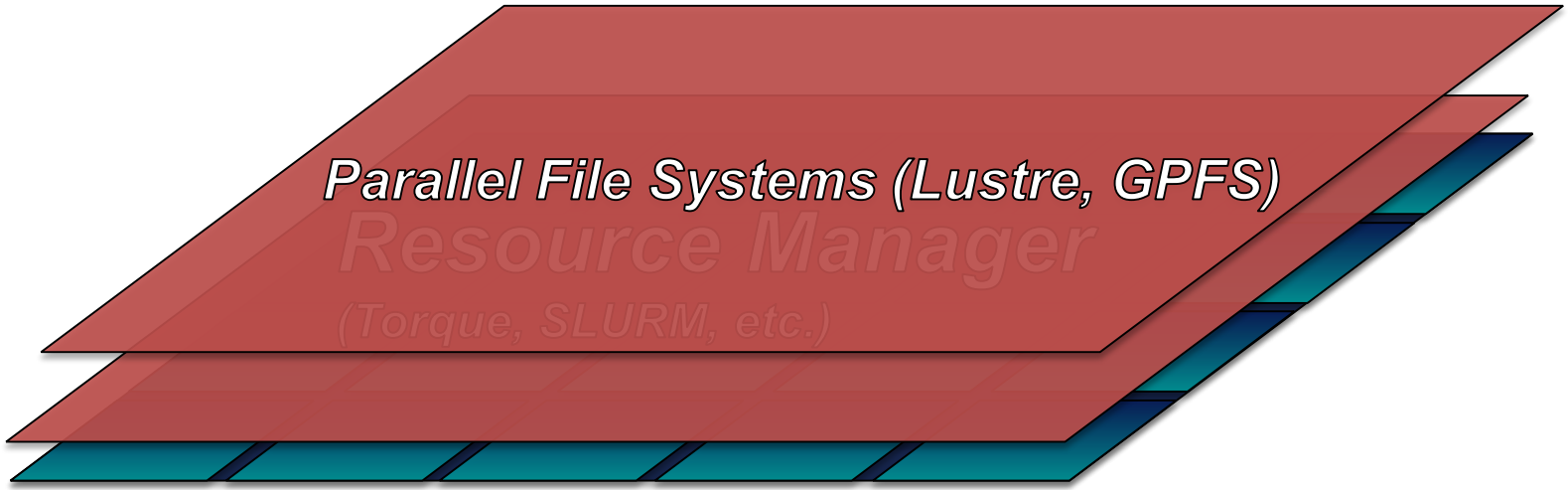
Convergence of HPC, Big Data, and Deep Learning!

Increasing Need to Run these applications on the Cloud!!

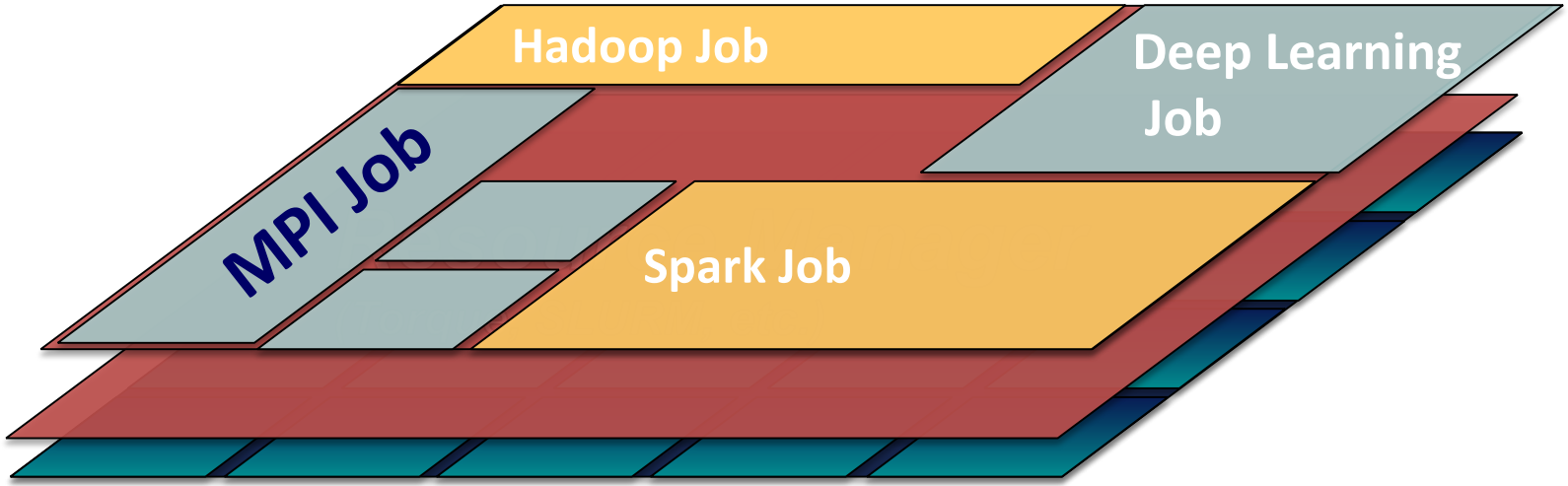
Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?

Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?

Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?



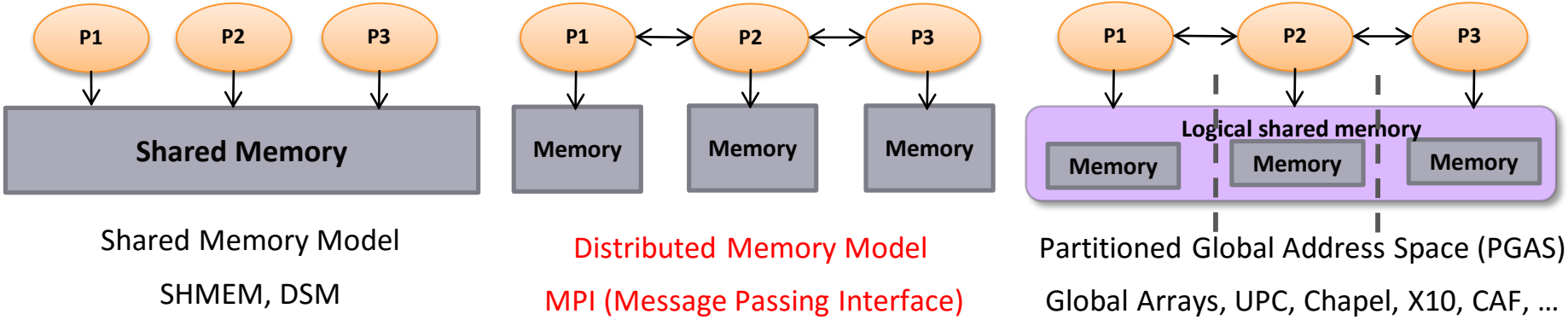
Can We Run Big Data and Deep Learning Jobs on Existing HPC Infrastructure?



HPC, Big Data, Deep Learning, and Cloud

- Traditional HPC
  - Message Passing Interface (MPI), including MPI + OpenMP
  - Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
  - Exploiting Accelerators
- Big Data/Enterprise/Commercial Computing
  - Spark and Hadoop (HDFS, HBase, MapReduce)
  - Memcached is also used for Web 2.0
- Deep Learning
  - Caffe, CNTK, TensorFlow, and many more
- Cloud for HPC and BigData
  - Virtualization with SR-IOV and Containers

Parallel Programming Models Overview



Shared Memory Model

SHMEM, DSM

Distributed Memory Model

MPI (Message Passing Interface)

Partitioned Global Address Space (PGAS)

Global Arrays, UPC, Chapel, X10, CAF, …

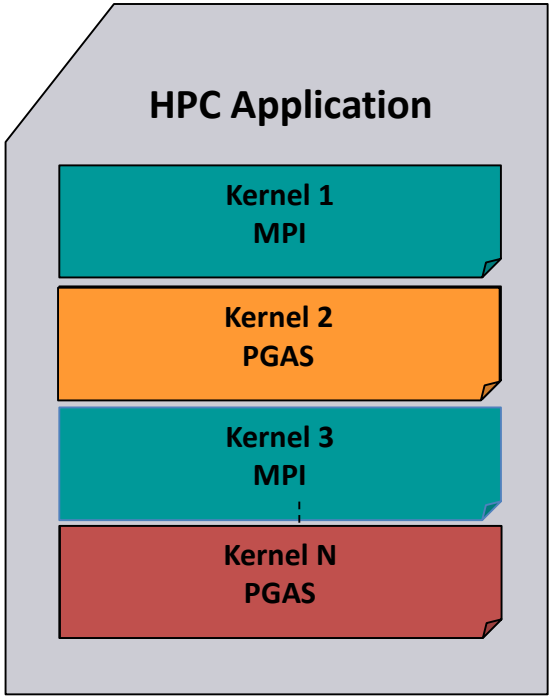
- Programming models provide abstract machine models
- Models can be mapped on different types of systems
  - e.g. Distributed Shared Memory (DSM), MPI within a node, etc.
- PGAS models and Hybrid MPI+PGAS models are gradually receiving importance

Partitioned Global Address Space (PGAS) Models

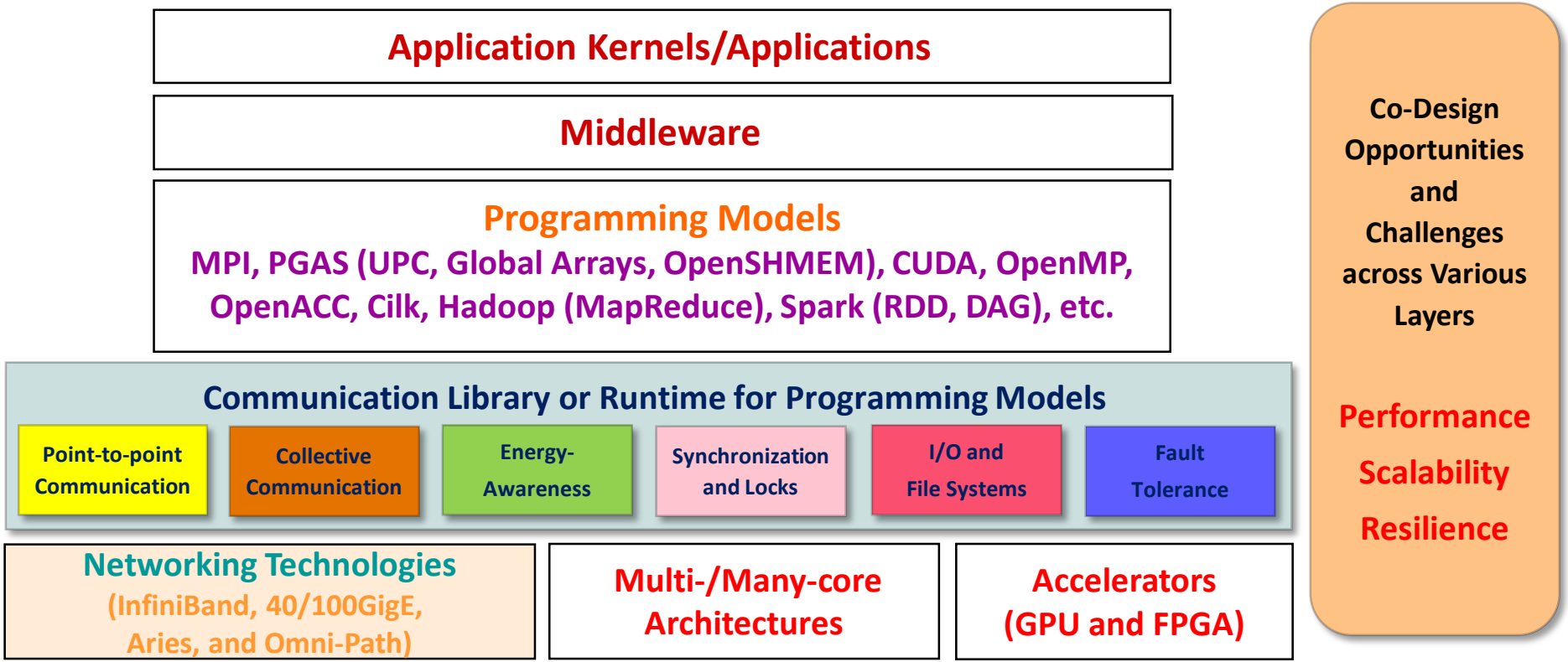
- Key features
  - Simple shared memory abstractions
  - Light weight one-sided communication
  - Easier to express irregular communication
- Different approaches to PGAS
  - Languages
    - Unified Parallel C (UPC)
    - Co-Array Fortran (CAF)
    - X10
    - Chapel
  - Libraries
    - OpenSHMEM
  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

Hybrid (MPI+PGAS) Programming

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



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



Broad Challenges in Designing Communication Middleware for (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
- 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 GPGPUs and Accelerators
- 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

Additional Challenges for Designing Exascale Software Libraries

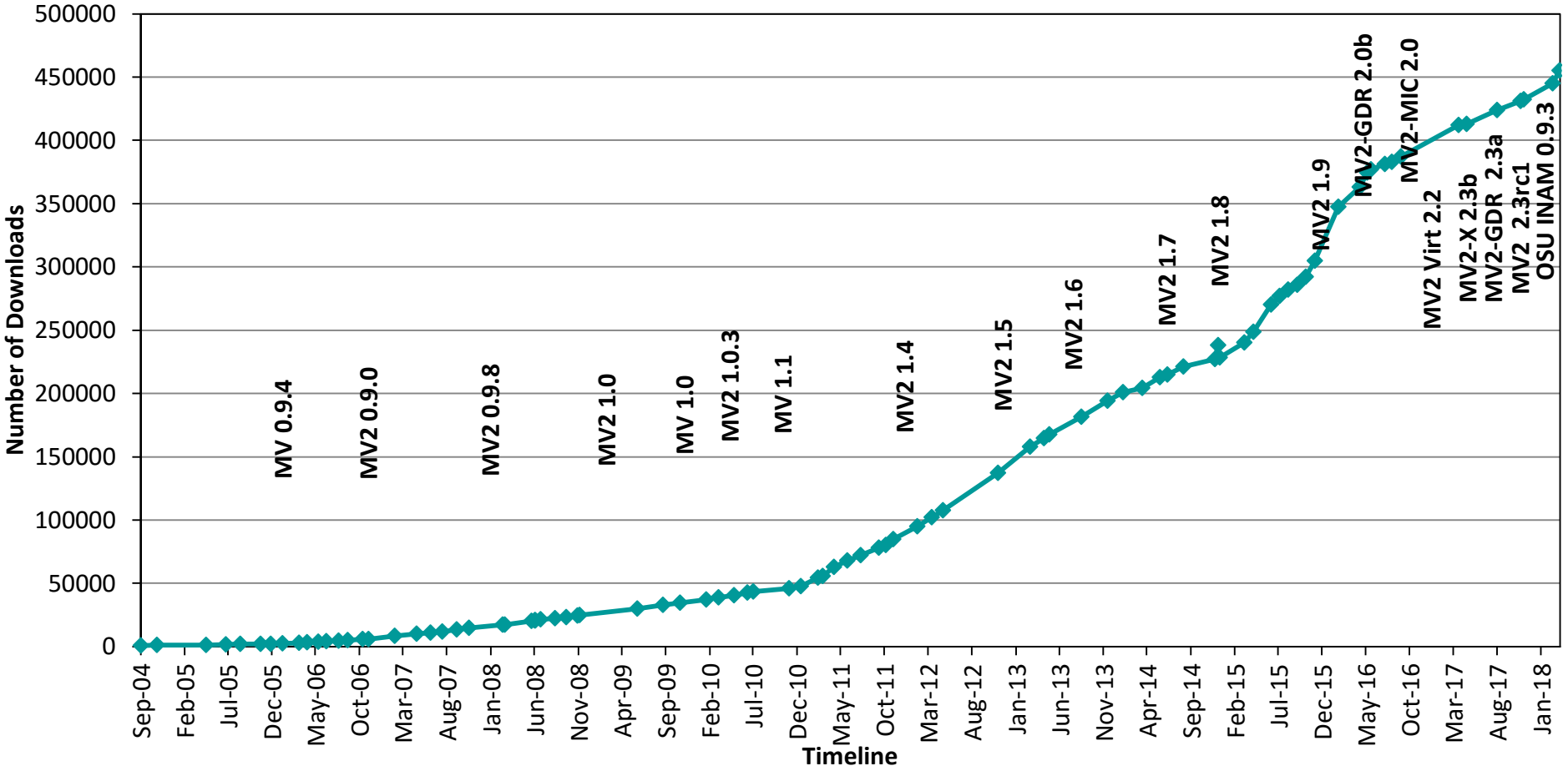
- **Extreme Low Memory Footprint**
  - Memory per core continues to decrease

- **D-L-A Framework**
  - **D**iscover
    - Overall network topology (fat-tree, 3D, …), Network topology for processes for a given job
    - Node architecture, Health of network and node
  - **L**earn
    - Impact on performance and scalability
    - Potential for failure
  - **A**dapt
    - Internal protocols and algorithms
    - Process mapping
    - Fault-tolerance solutions
  - Low overhead techniques while delivering performance, scalability and fault-tolerance

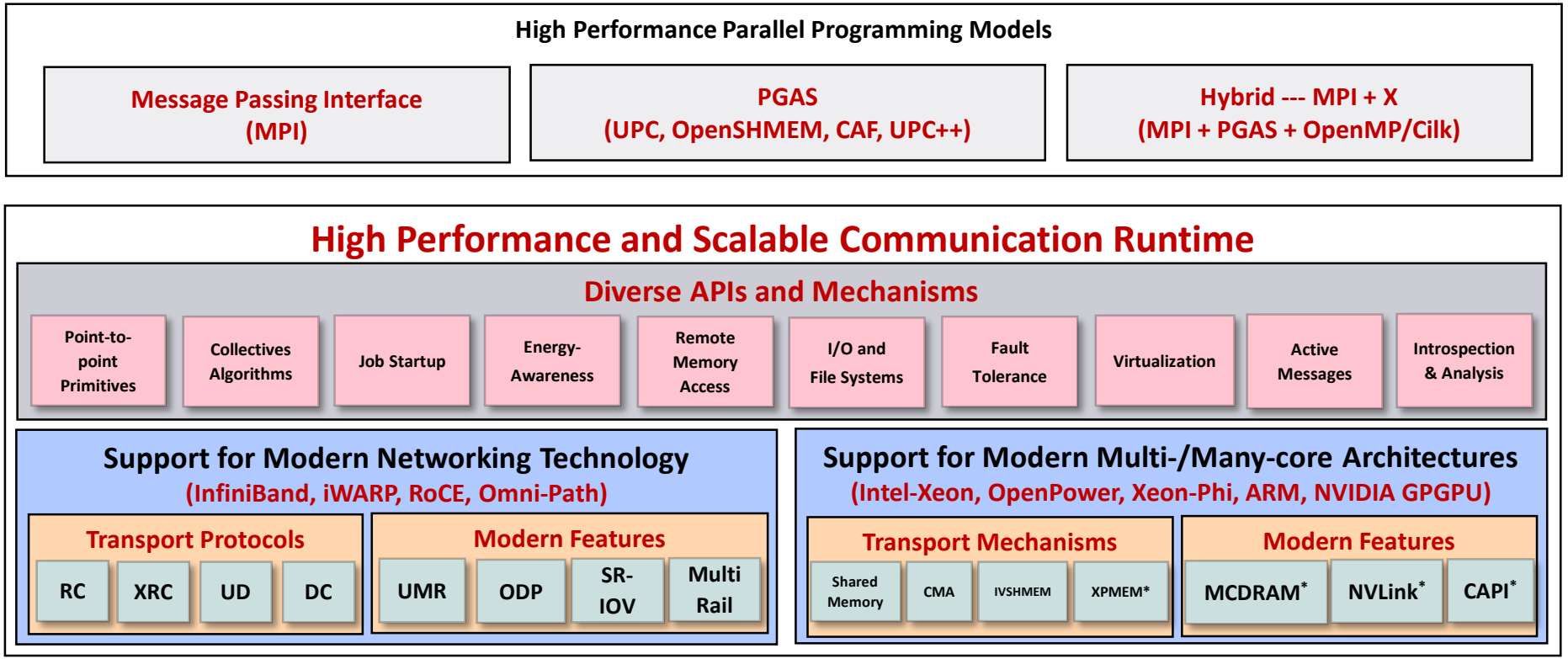
Overview of the MVAPICH2 Project

- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.1), Started in 2001, First version available in 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
  - **Used by more than 2,875 organizations in 86 countries**
  - **More than 460,000 (> 0.46 million) downloads from the OSU site directl**
  - Empowering many TOP500 clusters (Nov '17 ranking)
    - **1st, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China**
    - 9th, 556,104 cores (Oakforest-PACS) in Japan
    - 12th, 368,928-core (Stampede2) at TACC
    - 17th, 241,108-core (Pleiades) at NASA
    - 48th, 76,032-core (Tsubame 2.5) at Tokyo Institute of Technology
  - Available with software stacks of many vendors and Linux Distros (RedHat and SuSE)
  - **http://mvapich.cse.ohio-state.edu**
- Empowering Top500 systems for over a decade

MVAPICH2 Release Timeline and Downloads



Architecture of MVAPICH2 Software Family

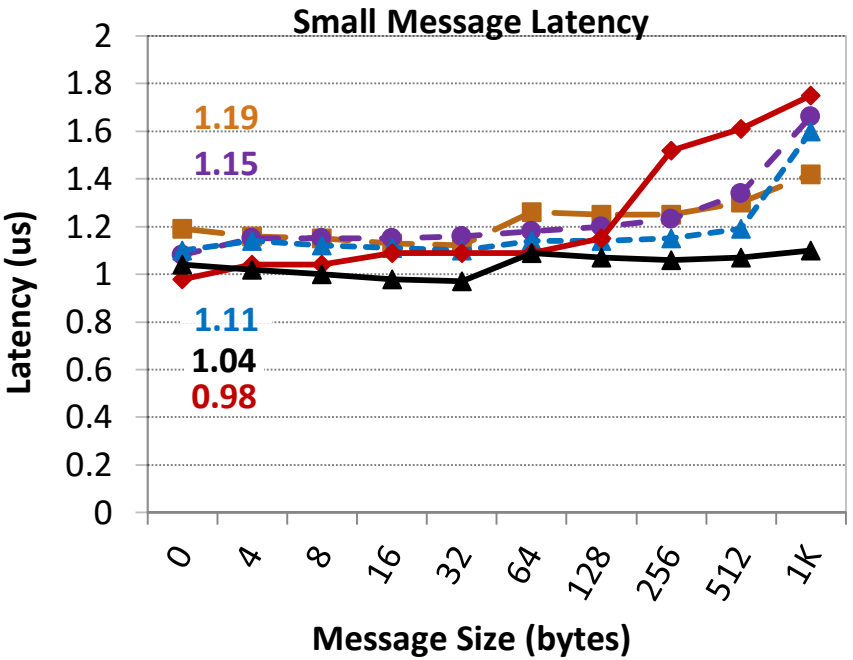


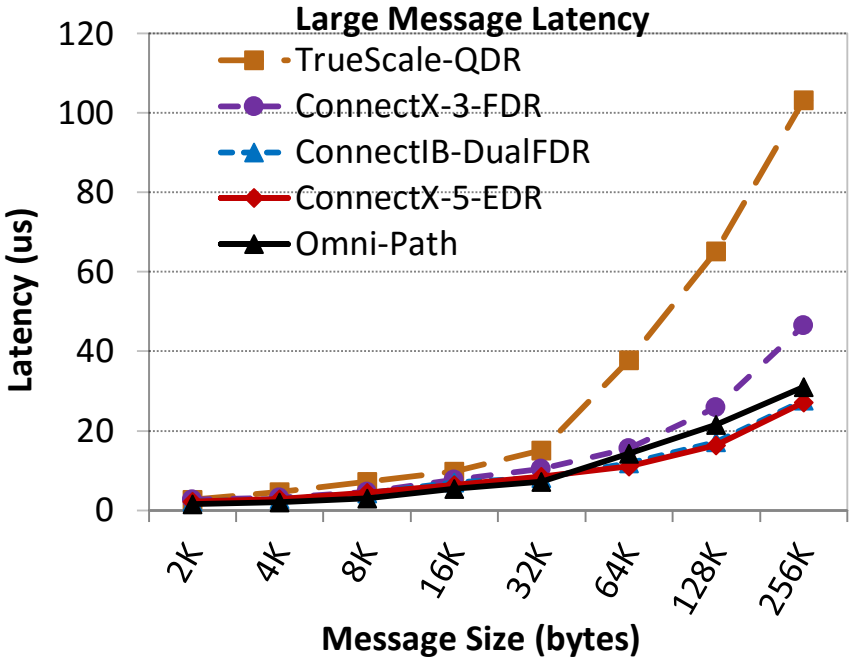
* Upcoming

Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
  - Support for highly-efficient inter-node and intra-node communication
  - Scalable Start-up
  - Optimized Collectives using SHArP and Multi-Leaders
  - Optimized CMA-based Collectives
  - Upcoming Optimized XPMEM-based Collectives
  - Performance Engineering with MPI-T Support
  - Integrated Network Analysis and Monitoring
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, UPC++, …)
- Integrated Support for GPGPUs
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- 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

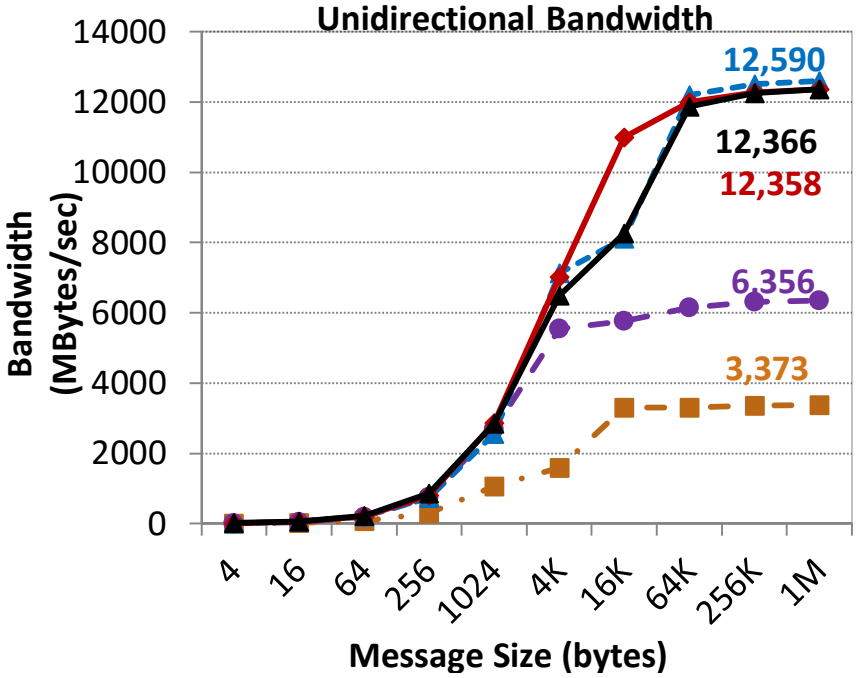
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

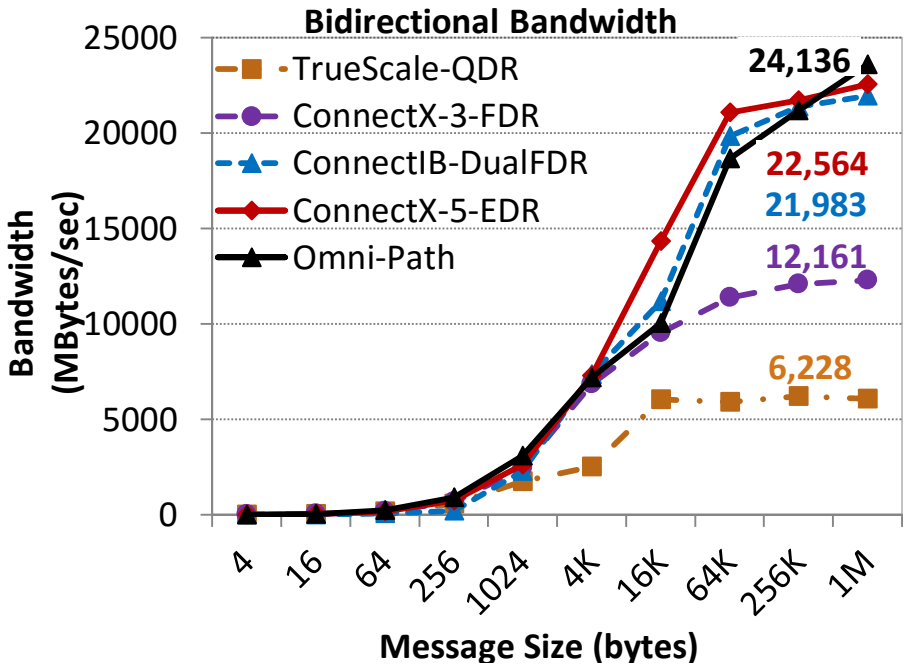
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

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

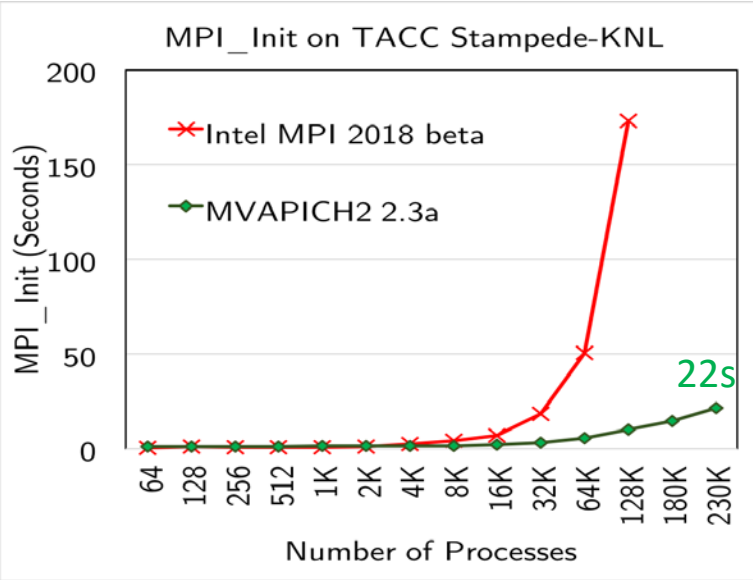
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

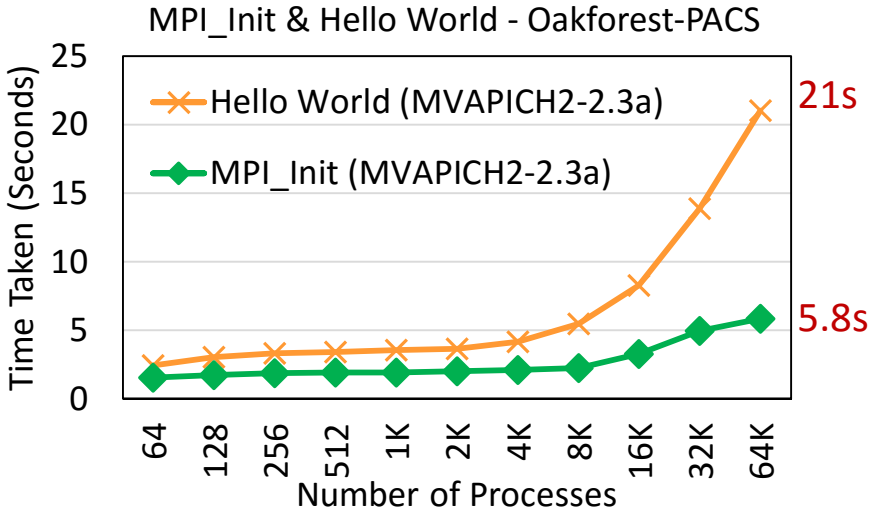
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

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

Startup Performance on KNL + Omni-Path

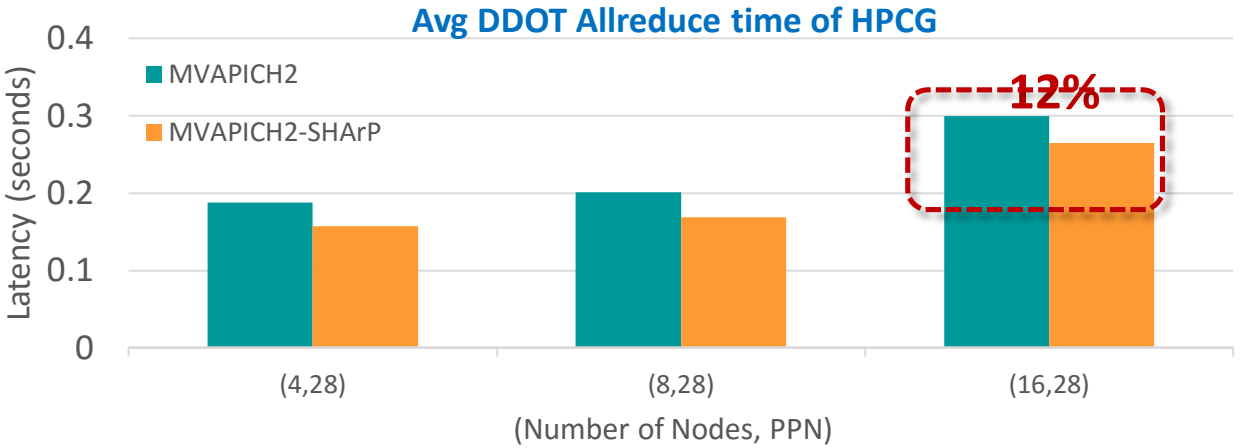


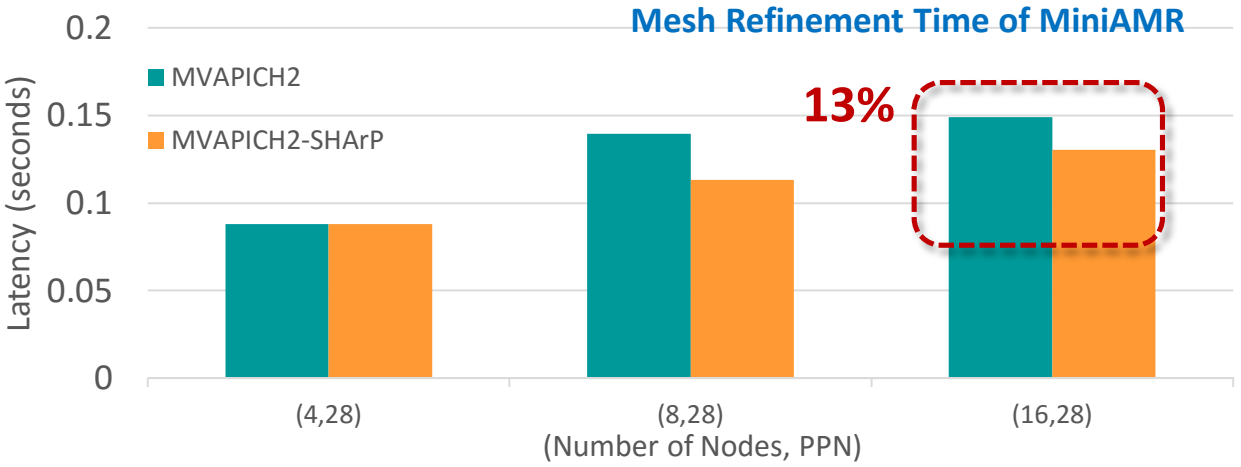


- MPI_Init takes 22 seconds on 229,376 processes on 3,584 KNL nodes (Stampede2 – Full scale)
- 8.8 times faster than Intel MPI at 128K processes (Courtesy: TACC)
- At 64K processes, MPI_Init and Hello World takes 5.8s and 21s respectively (Oakforest-PACS)
- All numbers reported with 64 processes per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

New designs available since MVAPICH2-2.3a and as patch for SLURM 15, 16, and 17

Advanced Allreduce Collective Designs Using SHArP

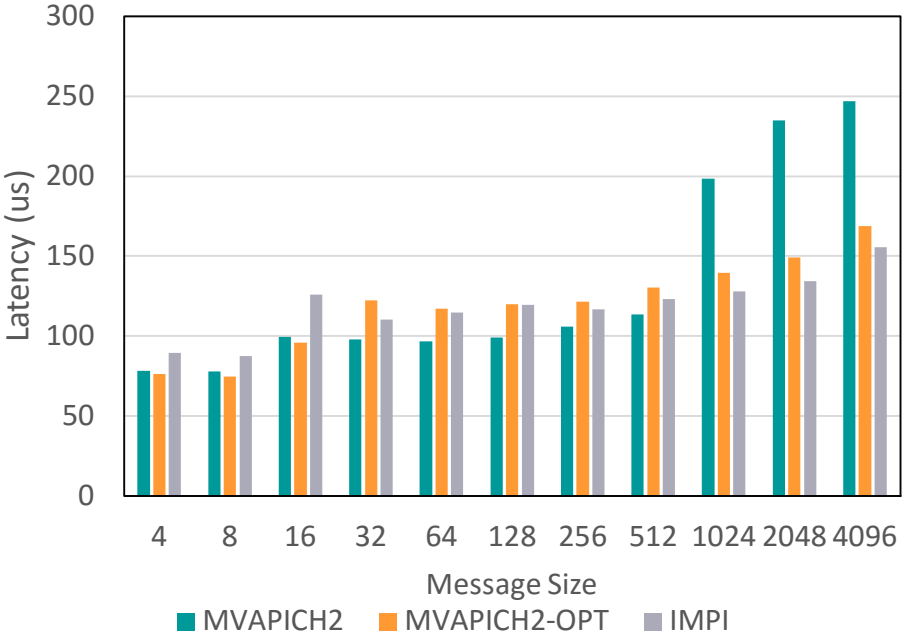


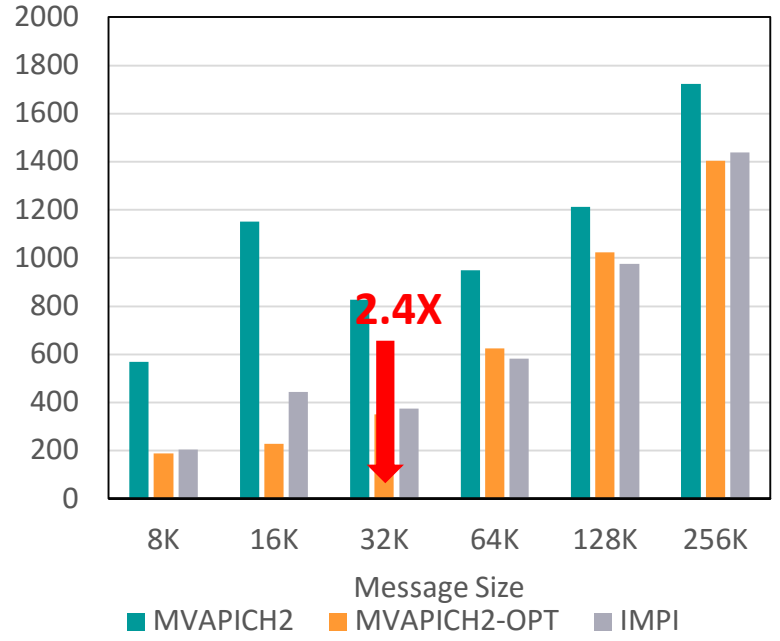


SHArP Support is available since MVAPICH2 2.3a

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

Performance of MPI_Allreduce On Stampede2 (10,240 Processes)





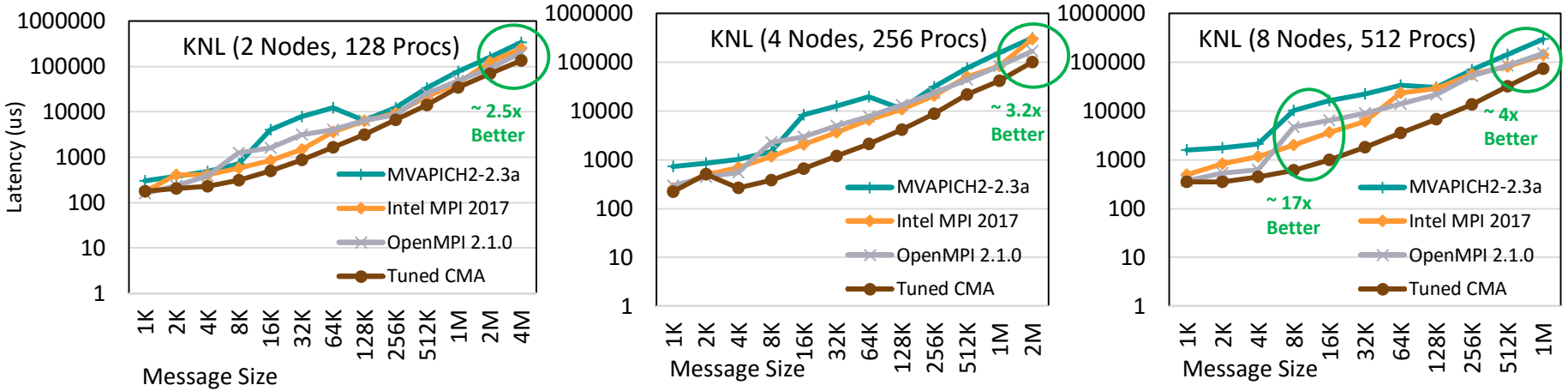
OSU Micro Benchmark 64 PPN

- For MPI_Allreduce latency with 32K bytes, MVAPICH2-OPT can reduce the latency by 2.4X

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

Available in MVAPICH2-X 2.3b

Optimized CMA-based Collectives for Large Messages



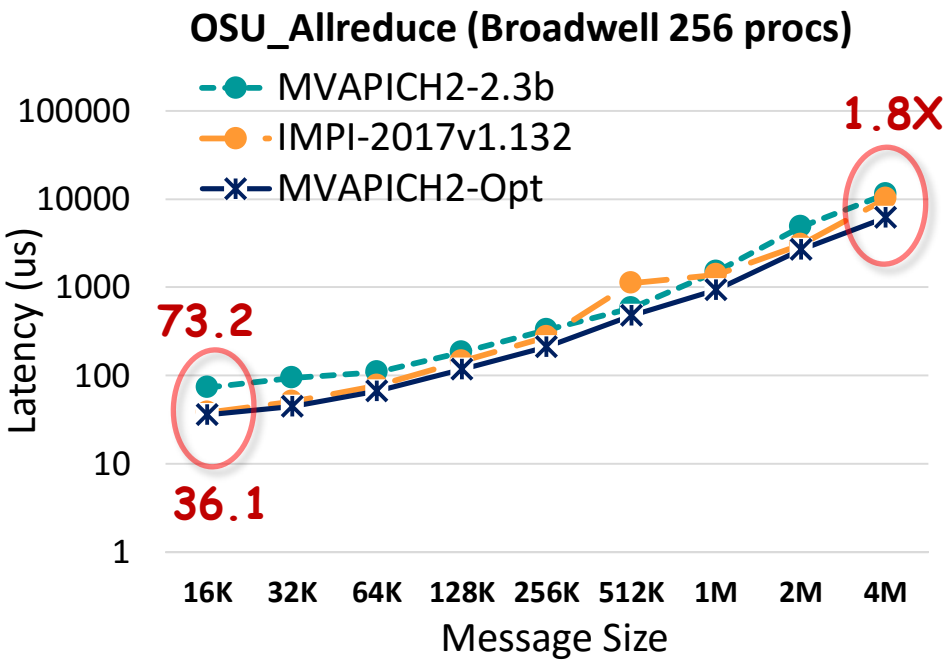
Performance of MPI_Gather on KNL nodes (64PPN)

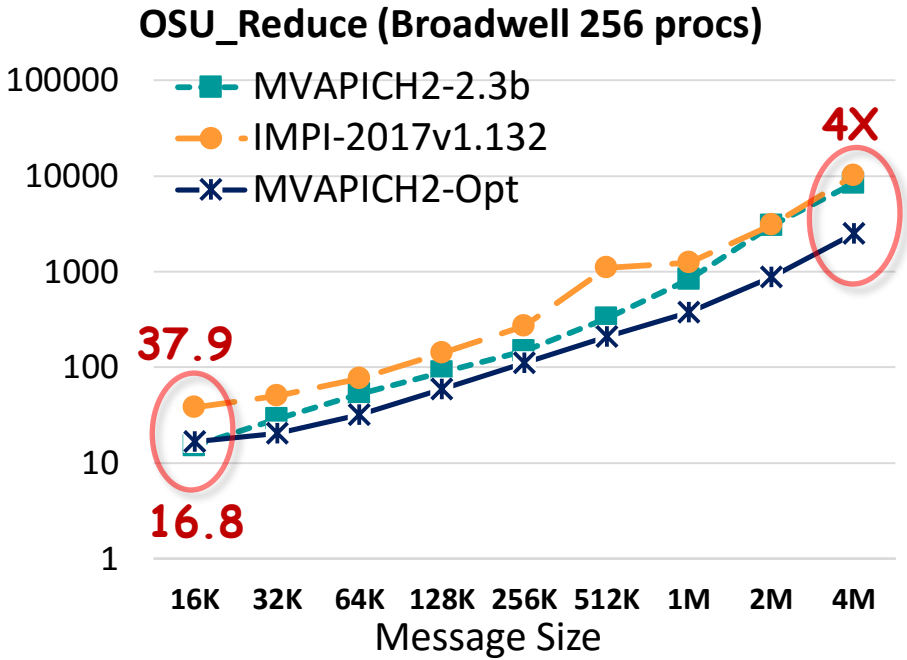
- Significant improvement over existing implementation for Scatter/Gather with 1MB messages (up to 4x on KNL, 2x on Broadwell, 14x on OpenPower)
- New two-level algorithms for better scalability
- Improved performance for other collectives (Bcast, Allgather, and Alltoall)

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 in MVAPICH2-X 2.3b

Shared Address Space (XPMEM)-based Collectives Design





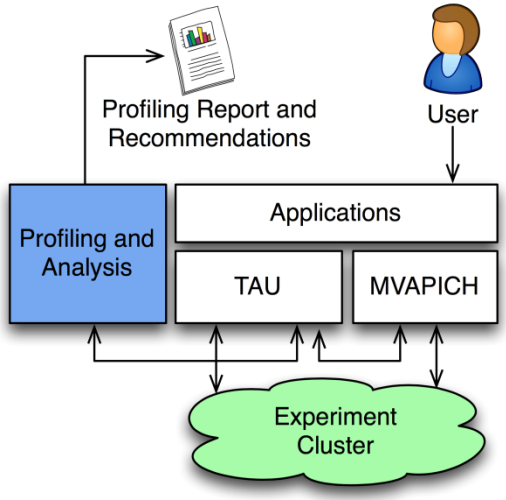
- "*Shared Address Space*"-based true *zero-copy* Reduction collective designs in MVAPICH2
- Offloaded computation/communication to peers ranks in reduction collective operation
- Up to **4X** improvement for 4MB Reduce and up to **1.8X** improvement for 4M AllReduce

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.

Will be available in future

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



VBUF usage without CVAR based tuning as displayed by ParaProf

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

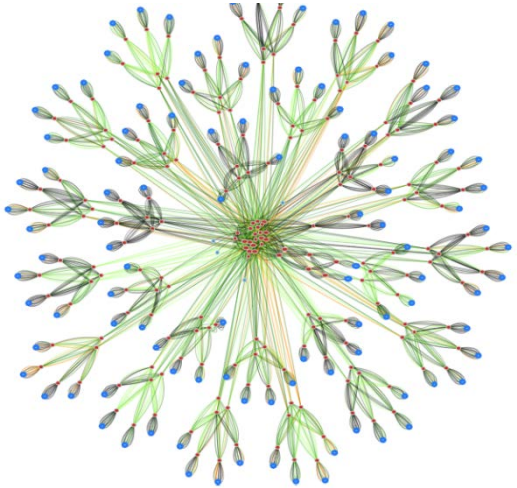
Name	MaxValue	MinValue	MeanValue	Std. Dev.	NumSamp...	Total
mv2_total_vbuf_memory (Total amount of memory in bytes used for VBUFs)	1,815,056	1,815,056	1,815,056	0	1	1,815,056
mv2_ud_vbuf_allocated (Number of UD VBUFs allocated)	0	0	0	0	0	0
mv2_ud_vbuf_available (Number of UD VBUFs available)	0	0	0	0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

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, Euro MPI, 2017 [Best Paper Award]

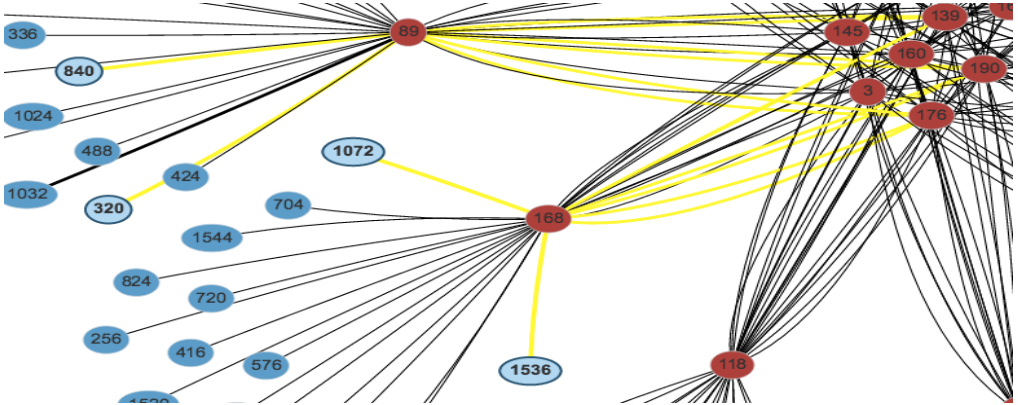
Overview of OSU INAM

- 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

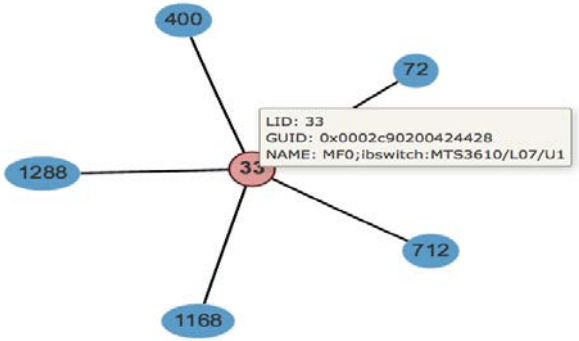


Finding Routes Between Nodes

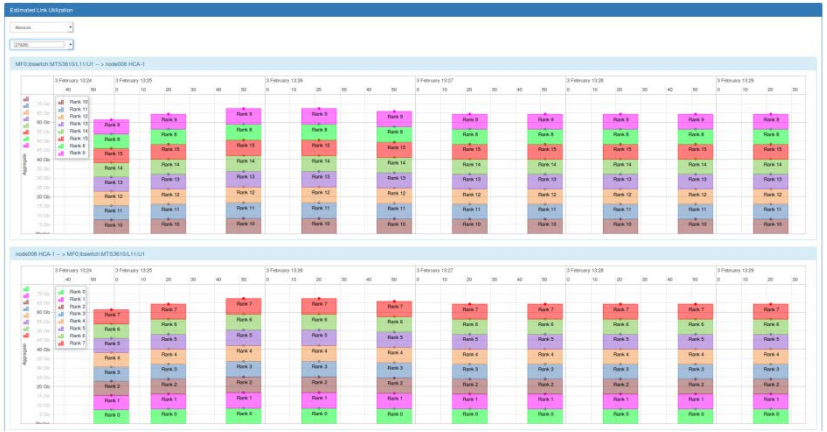
(1,879 nodes, 212 switches, 4,377 network links)

- 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)



Estimated Process Level Link Utilization

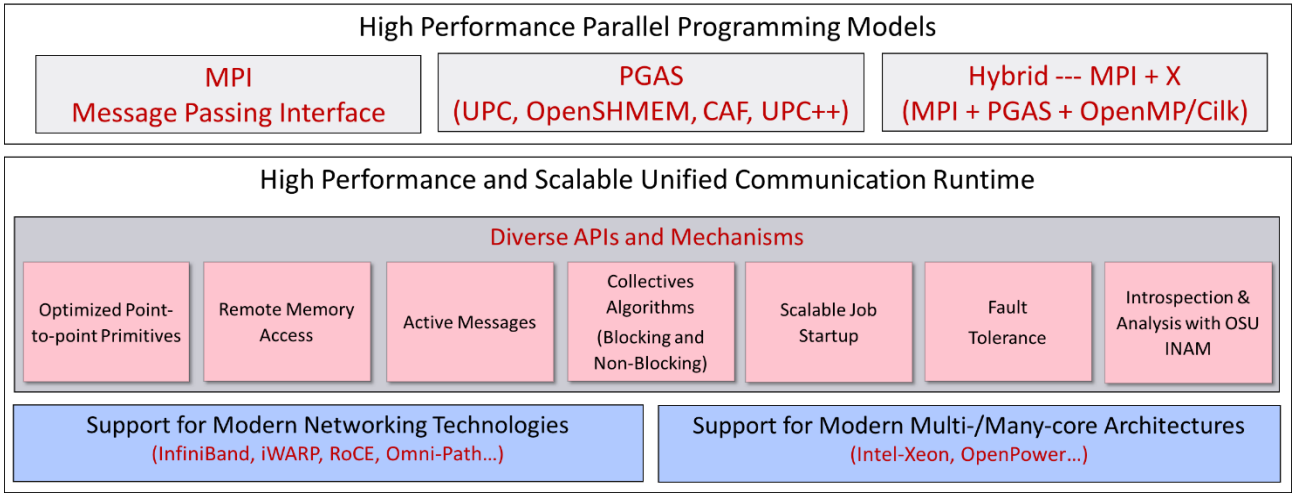
- Job level view
  - Show different network metrics (load, error, etc.) for any live job
  - Play back historical data for completed jobs to identify bottlenecks
- Node level view - details per process or per node
  - CPU utilization for each rank/node
  - Bytes sent/received for MPI operations (pt-to-pt,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

Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

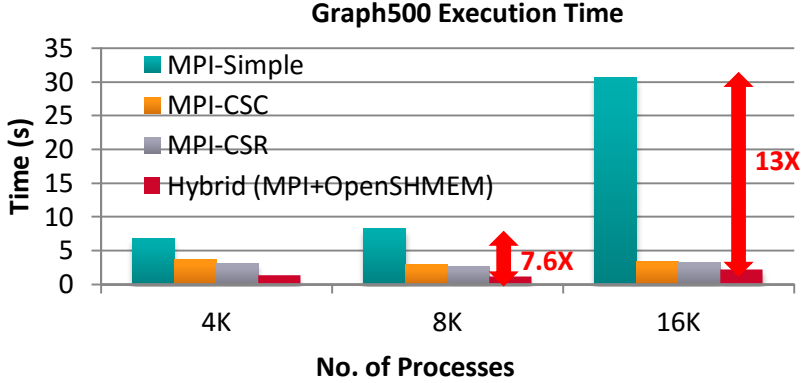
- Scalability for million to billion processors
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, UPC++, …)
- Integrated Support for GPGPUs
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- Application Scalability and Best Practices

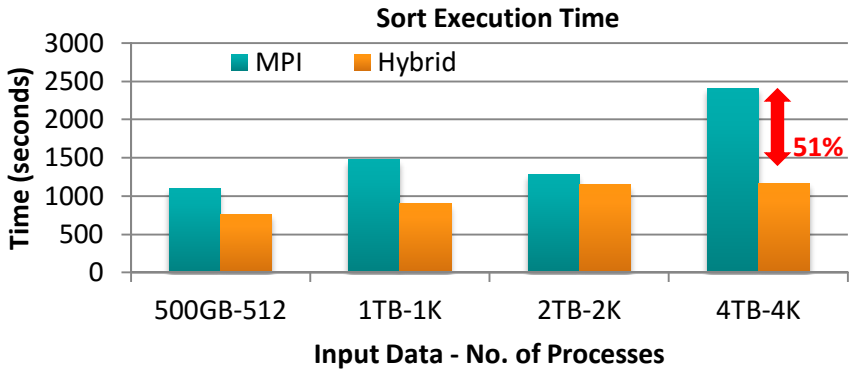
MVAPICH2-X for Hybrid MPI + PGAS Applications



- 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

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

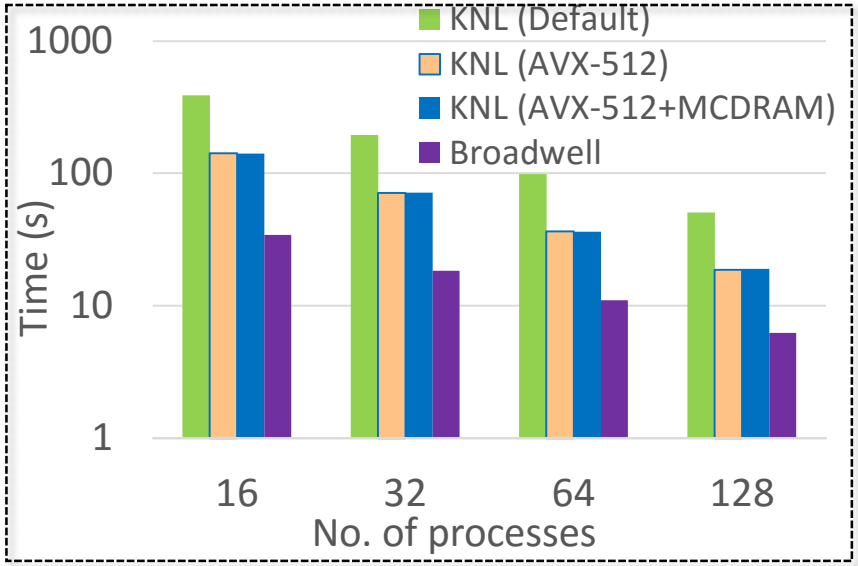
J. Jose, K. Kandalla, S. Potluri, J. Zhang and D. K. Panda, Optimizing Collective Communication in OpenSHMEM, Int'l Conference on Partitioned Global Address Space Programming Models (PGAS '13), October 2013.

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

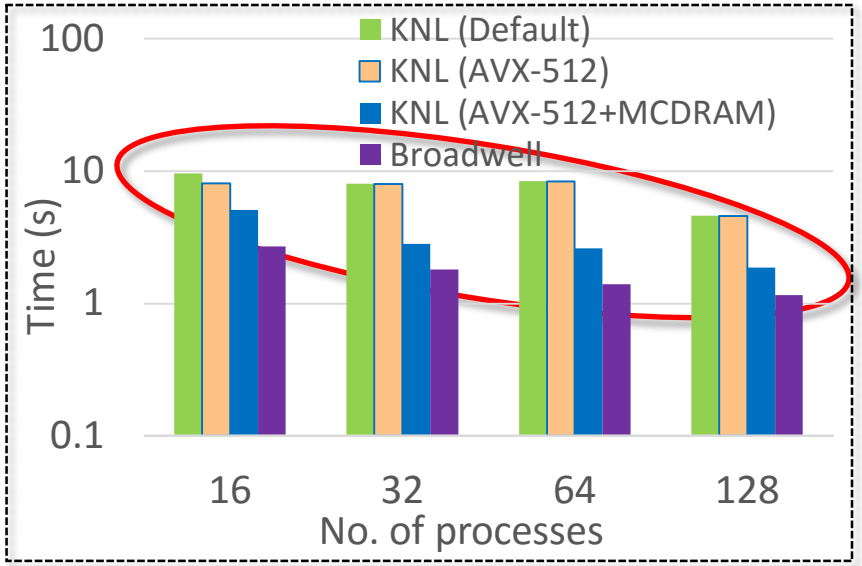
J. Jose, K. Kandalla, M. Luo and D. K. Panda, Supporting Hybrid MPI and OpenSHMEM over InfiniBand: Design and Performance Evaluation, Int'l Conference on Parallel Processing (ICPP '12), September 2012

Optimized OpenSHMEM with AVX and MCDRAM: Application Kernels Evaluation

Heat-2D Kernel using Jacobi method



Heat Image Kernel



- On heat diffusion based kernels AVX-512 vectorization showed better performance
- MCDRAM showed significant benefits on Heat-Image kernel for all process counts. Combined with AVX-512 vectorization, it showed up to 4X improved performance

Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, UPC++, …)
- Integrated Support for GPGPUs
  - CUDA-aware MPI
  - GPUDirect RDMA (GDR) Support
  - Support for Streaming Applications
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- Application Scalability and Best Practices

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

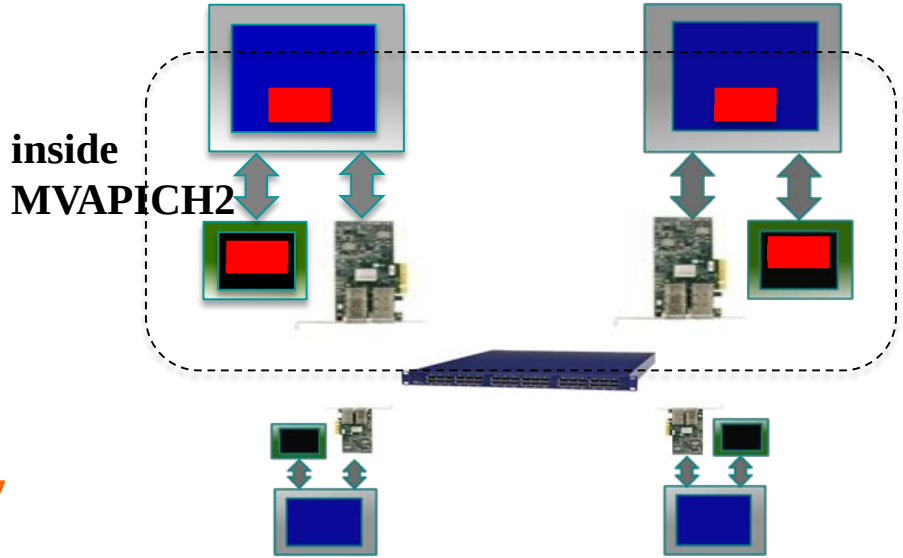
- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

At Sender:

```
MPI_Send(s_devbuf, size, …);
```

At Receiver:

```
MPI_Recv(r_devbuf, size, …);
```

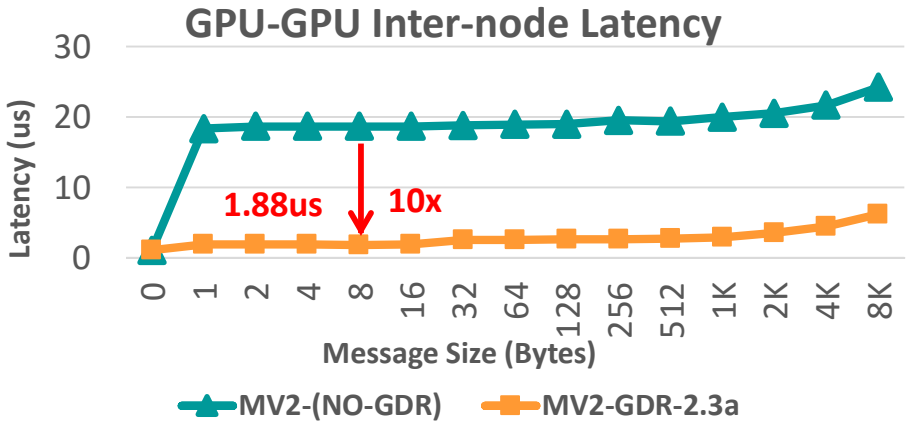


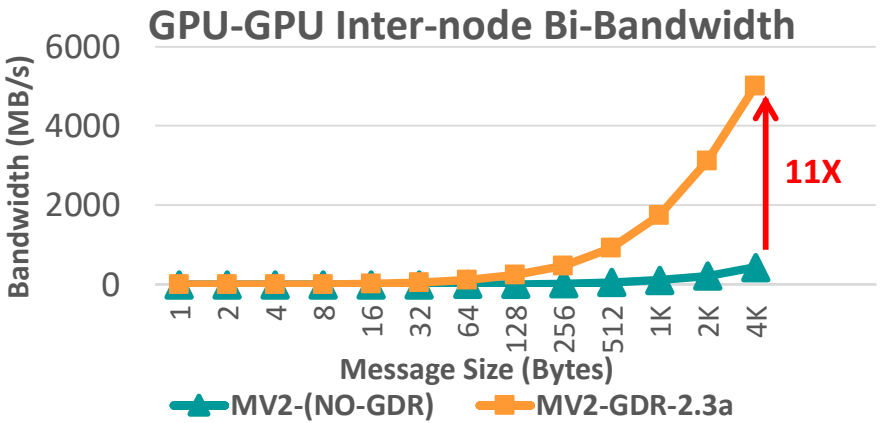
High Performance and High Productivity

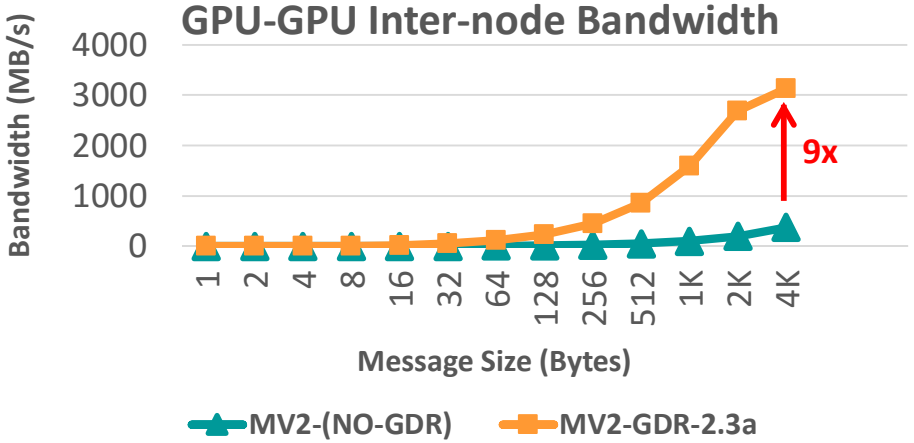
CUDA-Aware MPI: MVAPICH2-GDR 1.8-2.3 Releases

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

Optimized MVAPICH2-GDR Design

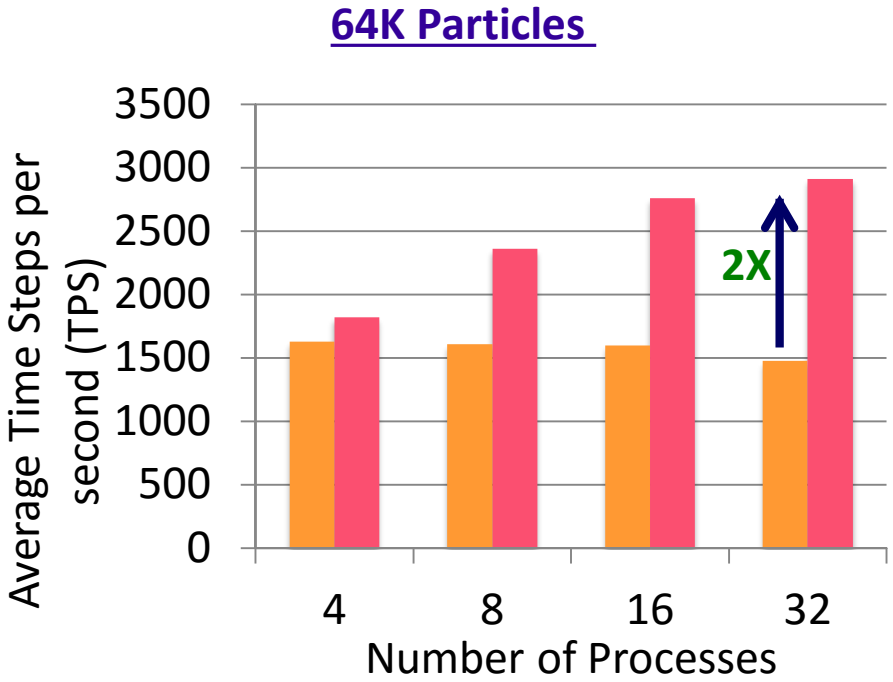


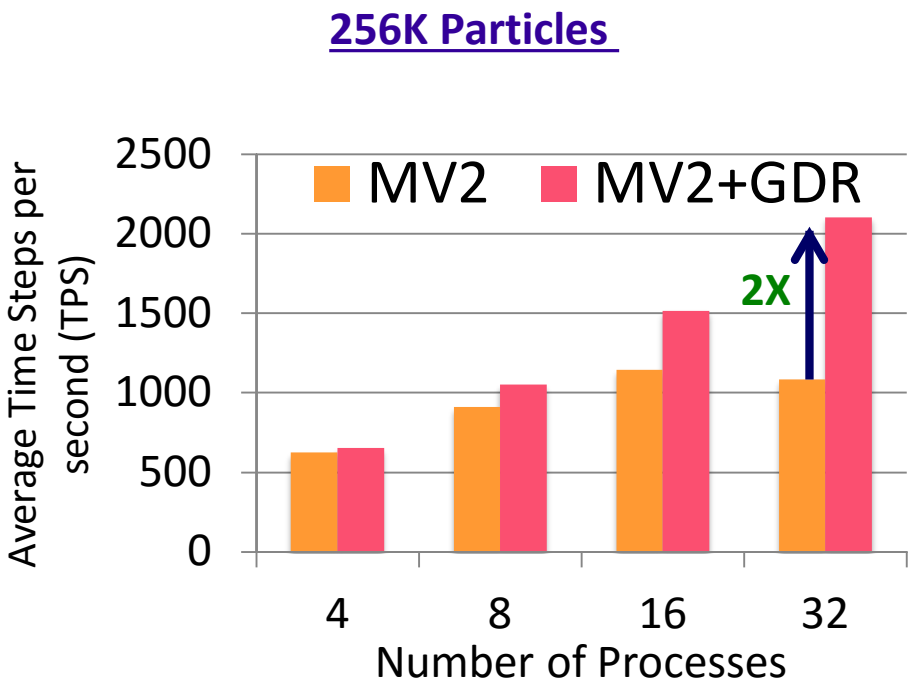




MVAPICH2-GDR-2.3a
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

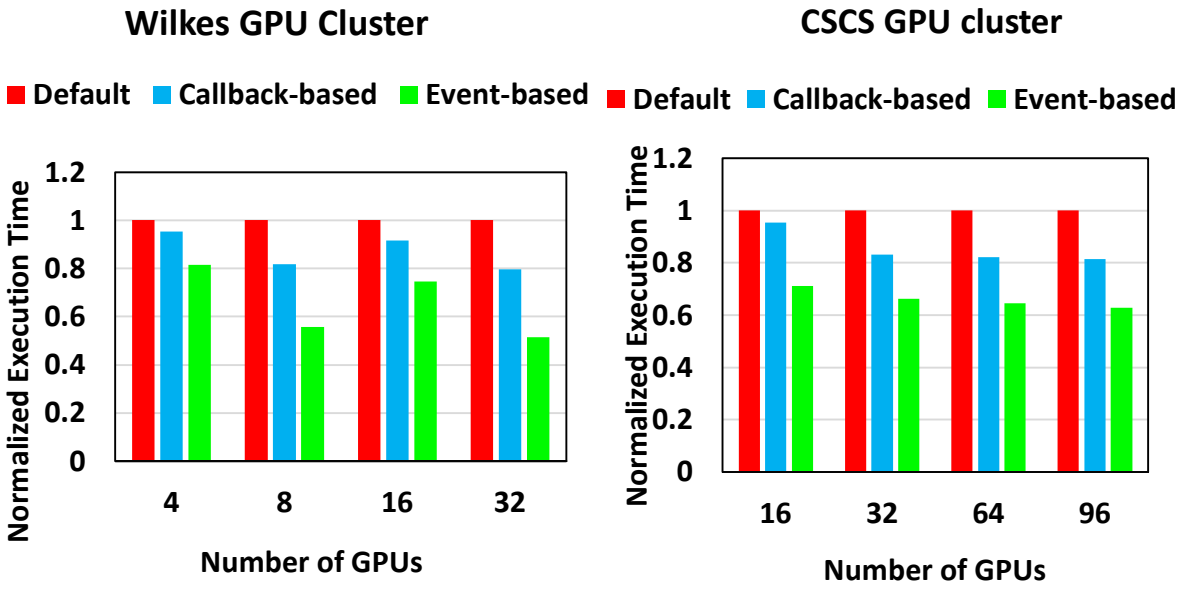
Application-Level Evaluation (HOOMD-blue)

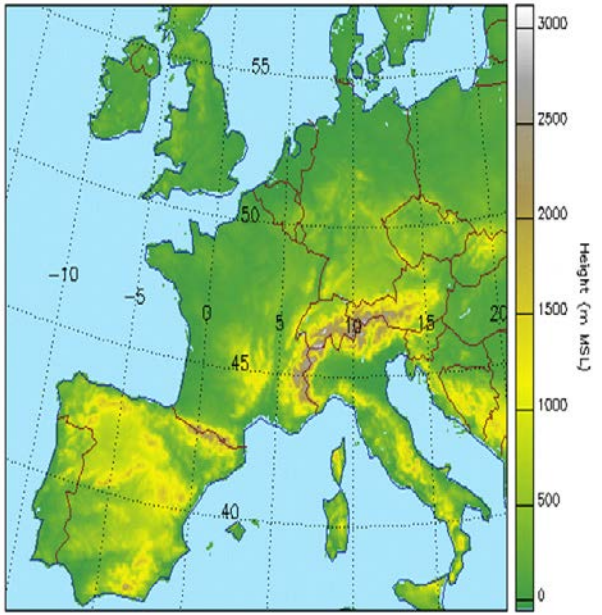




- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- HoomdBlue Version 1.0.5
  - GDRCOPY enabled: MV2_USE_CUDA=1 MV2_IBA_HCA=mlx5_0 MV2_IBA_EAGER_THRESHOLD=32768 MV2_VBUF_TOTAL_SIZE=32768 MV2_USE_GPUDIRECT_LOOPBACK_LIMIT=32768 MV2_USE_GPUDIRECT_GDRCOPY=1 MV2_USE_GPUDIRECT_GDRCOPY_LIMIT=16384

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





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

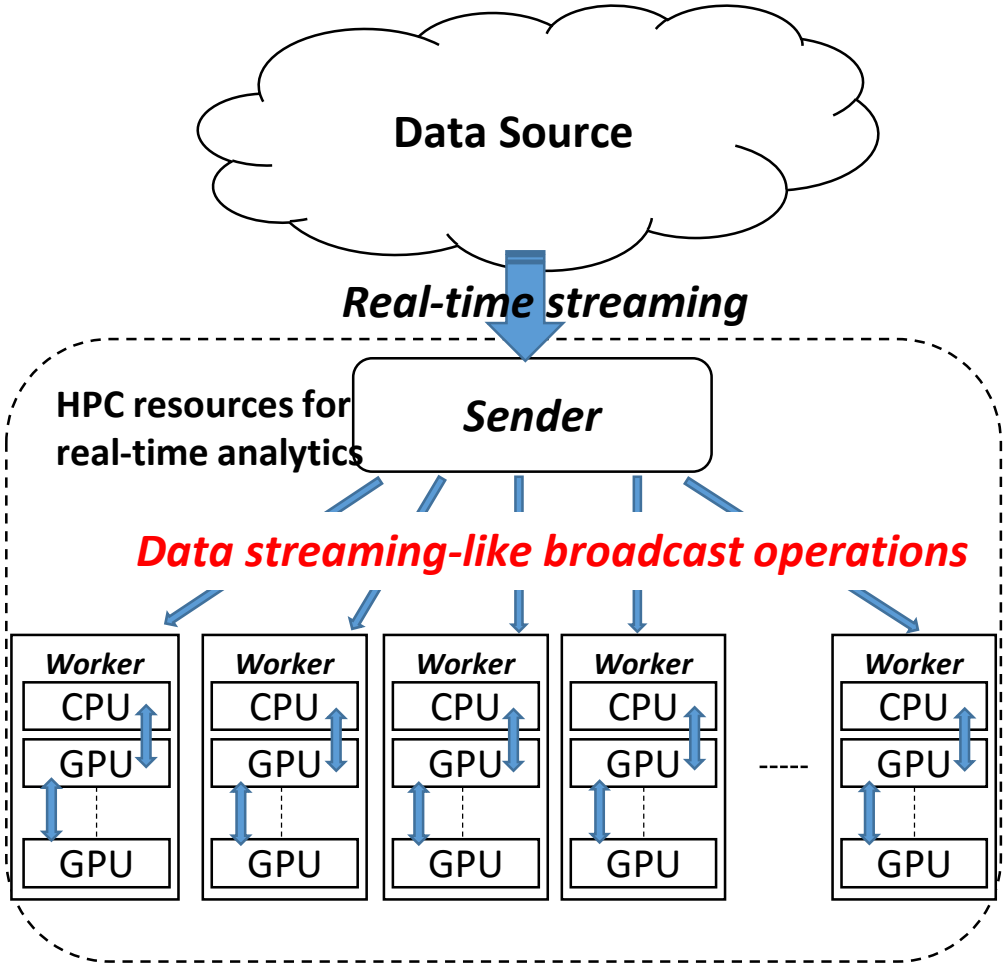
Cosmo model: http://www2.cosmo-model.org/content/tasks/operational/meteoSwiss/

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

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

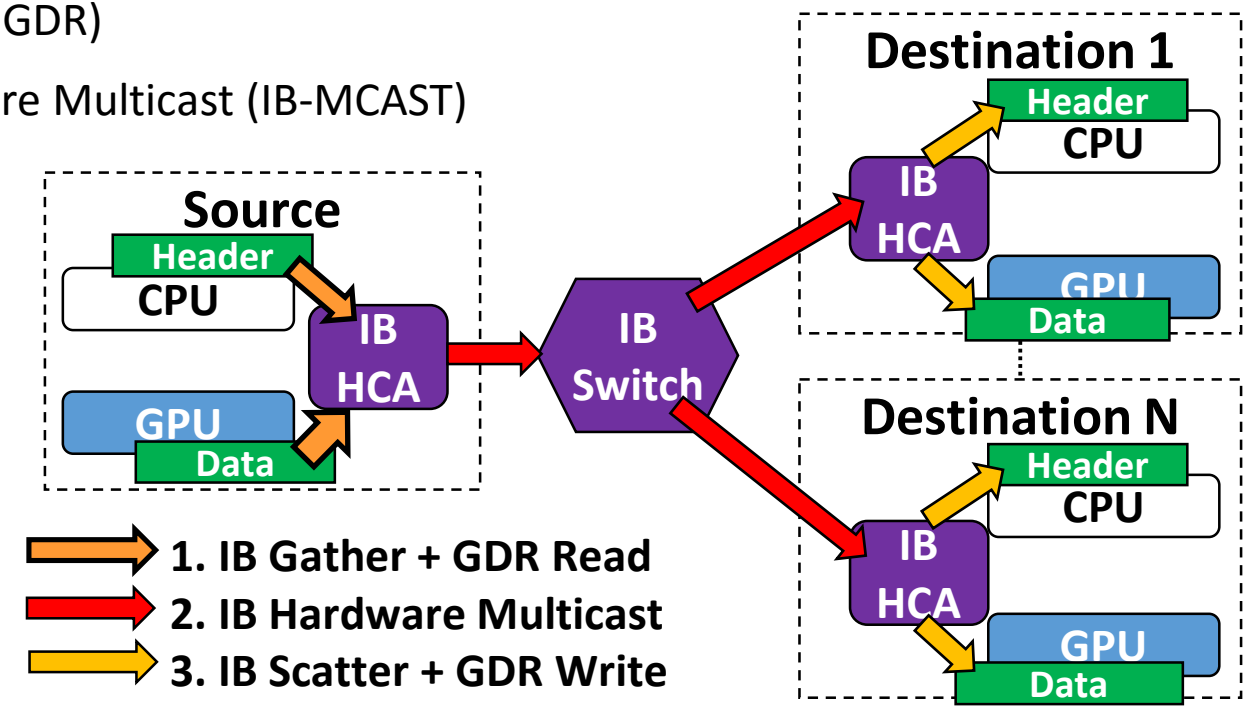
Streaming Applications

- Streaming applications on HPC systems
  1. Communication (MPI)
     - Broadcast-type operations
  2. Computation (CUDA)
     - Multiple GPU nodes as workers



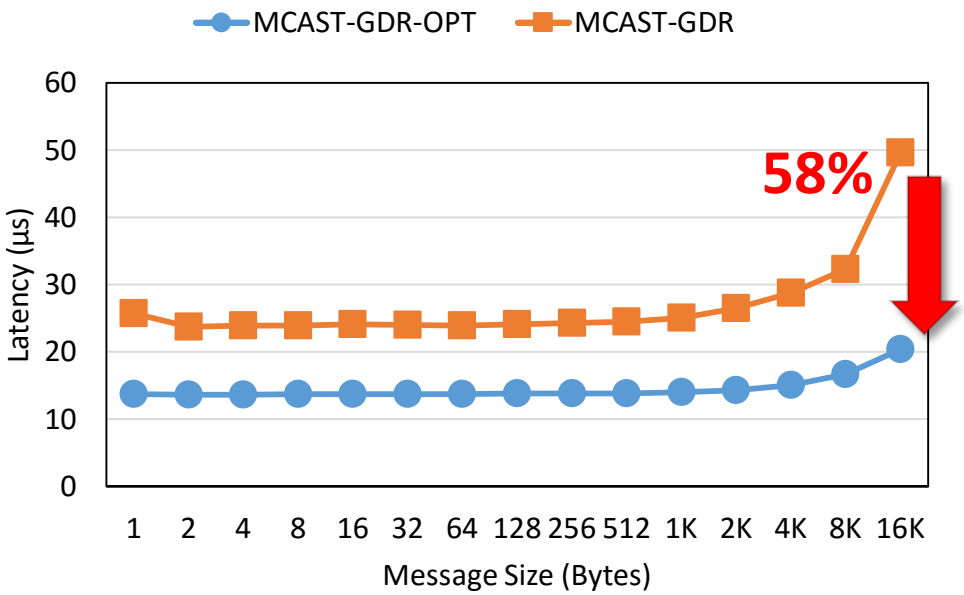
Hardware Multicast-based Broadcast

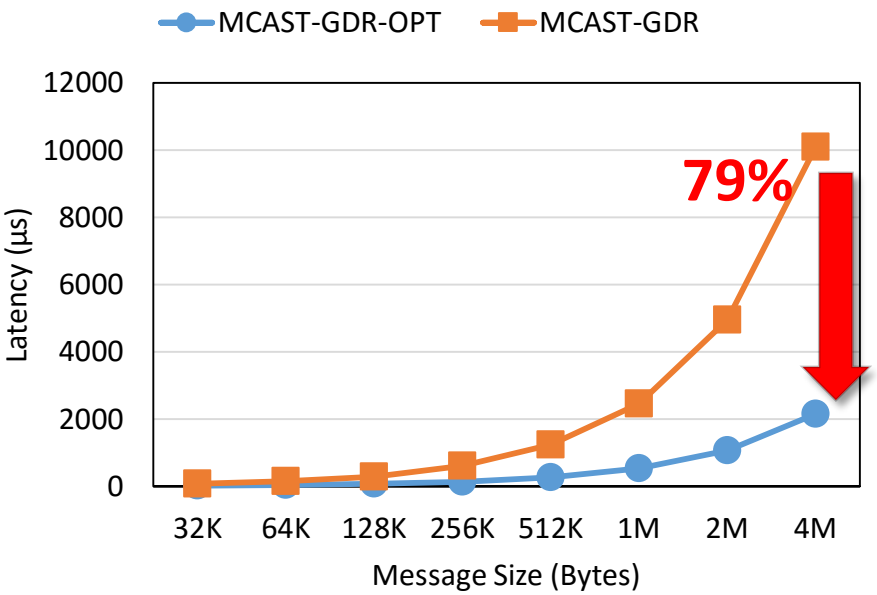
- For GPU-resident data, using
  - GPUDirect RDMA (GDR)
  - InfiniBand Hardware Multicast (IB-MCAST)
- Overhead
  - IB UD limit
  - GDR limit



A. Venkatesh, H. Subramoni, K. Hamidouche, and D. K. Panda, "A High Performance Broadcast Design with Hardware Multicast and GPUDirect RDMA for Streaming Applications on InfiniBand Clusters," in *HiPC 2014*, Dec 2014.

Streaming Benchmark @ CSCS (88 GPUs)





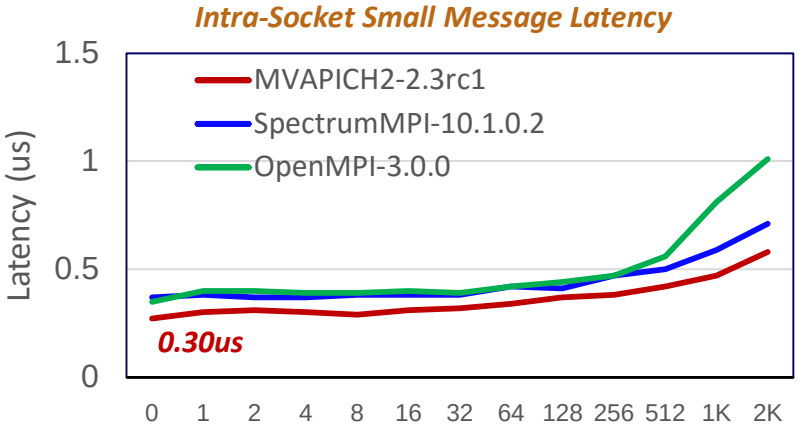
- **IB-MCAST + GDR + Topology-aware IPC-based schemes**
  - Up to **58% and 79% reduction** for small and large messages

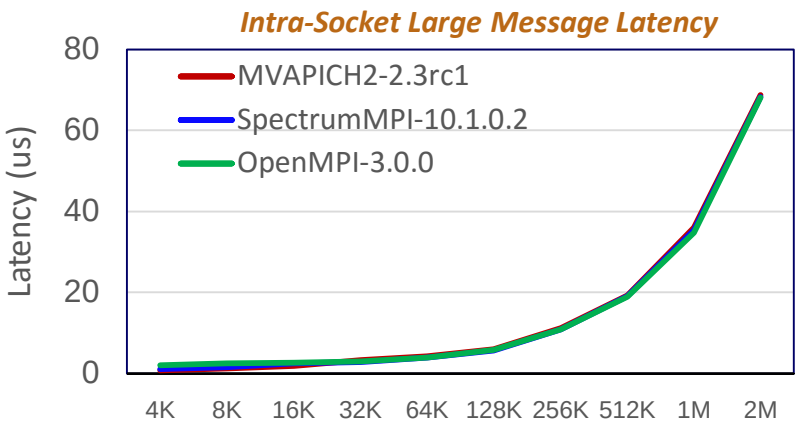
C.-H. Chu, K. Hamidouche, H. Subramoni, A. Venkatesh, B. Elton, and D. K. Panda, "Designing High Performance Heterogeneous Broadcast for Streaming Applications on GPU Clusters, " SBAC-PAD'16, Oct. 26-28, 2016.

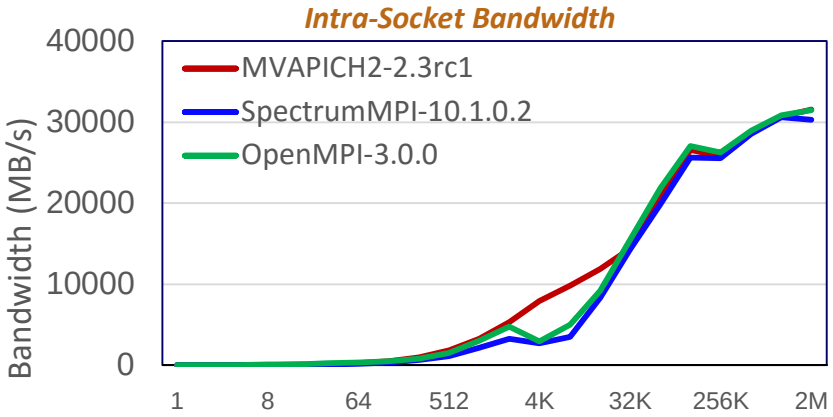
Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

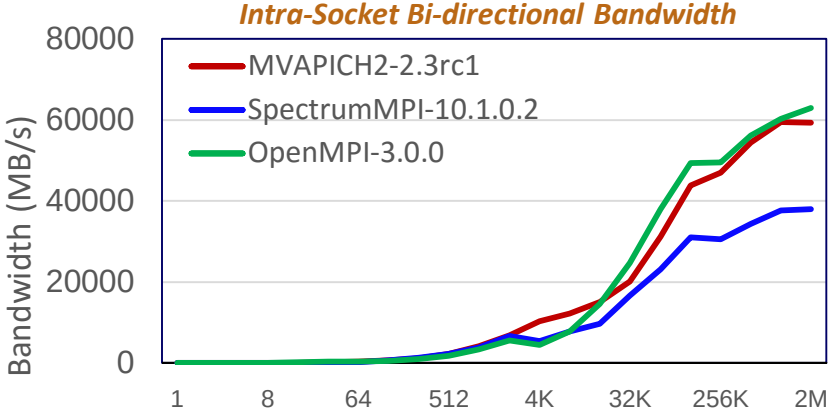
- Scalability for million to billion processors
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, UPC++, …)
- Integrated Support for GPGPUs
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- Application Scalability and Best Practices

Intra-node Point-to-Point Performance on OpenPower



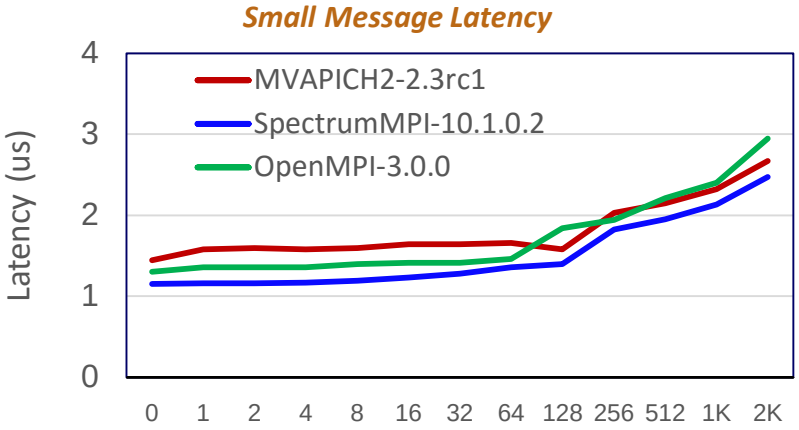


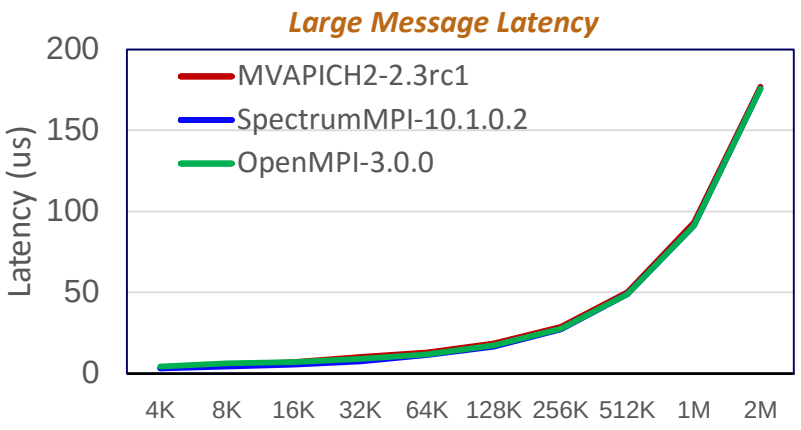


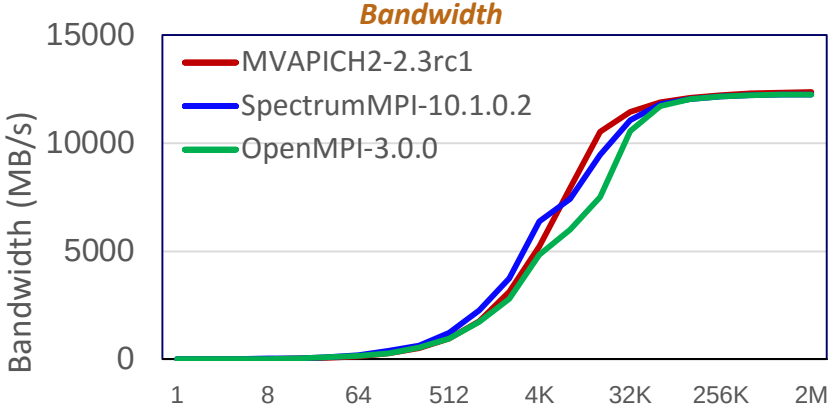


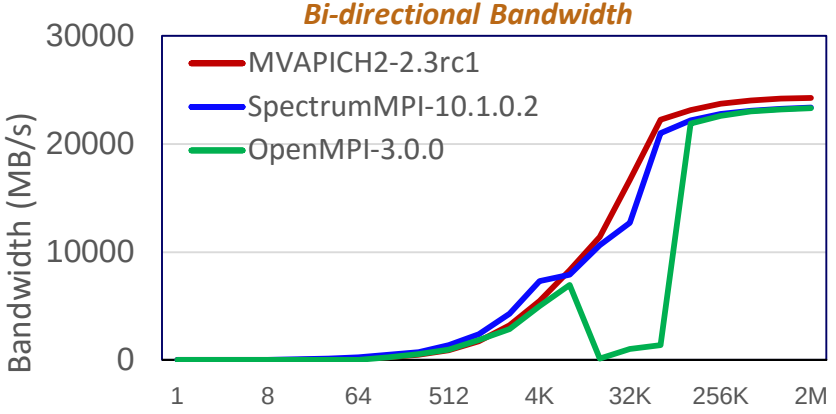
Platform: Two nodes of OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA

Inter-node Point-to-Point Performance on OpenPower



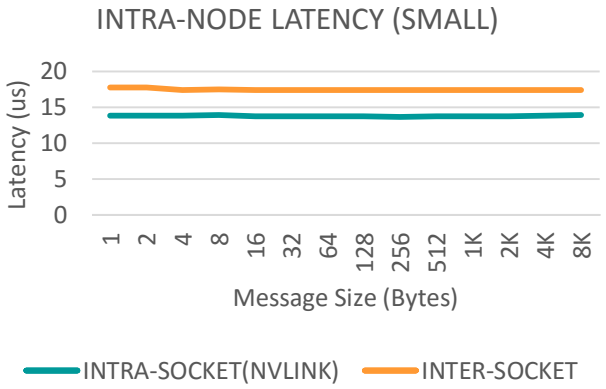




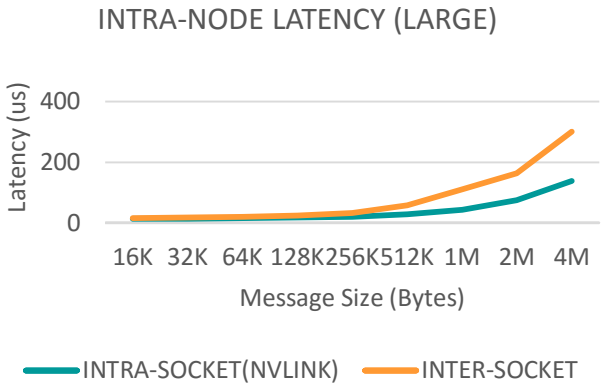


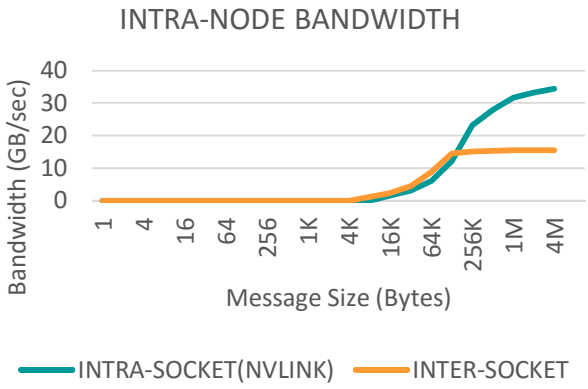
Platform: Two nodes of OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA

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

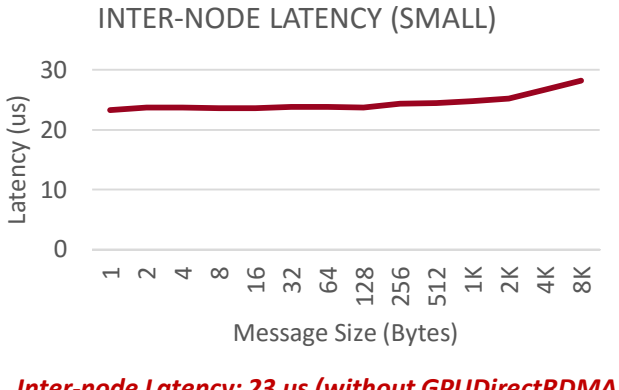


Intra-node Latency: 13.8 us (without GPUDirectRDMA)

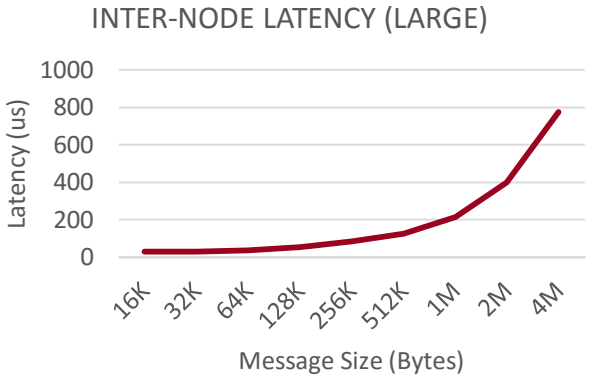


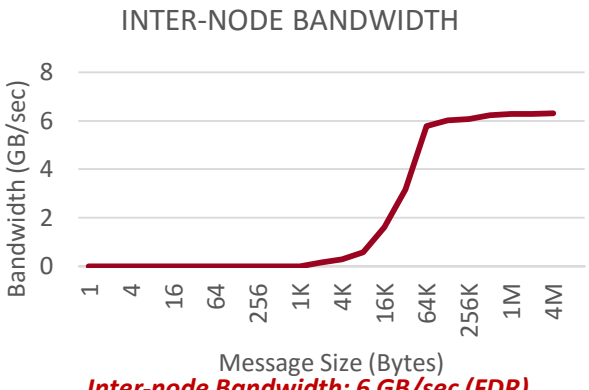


Intra-node Bandwidth: 33.2 GB/sec (NVLINK)



Inter-node Latency: 23 us (without GPUDirectRDMA)





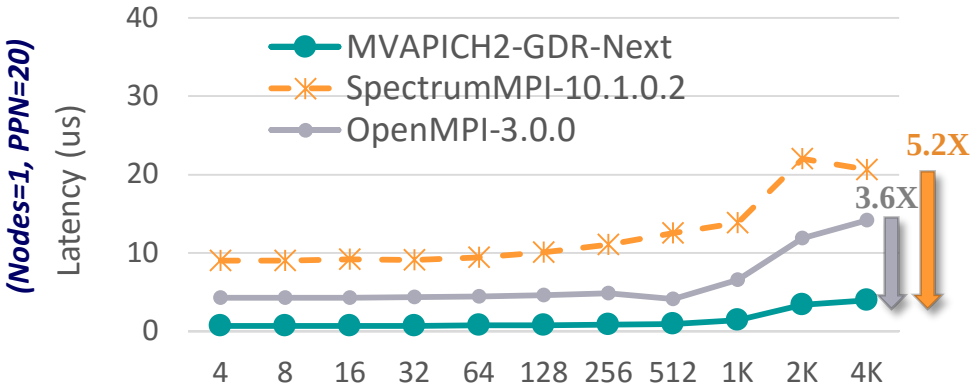
Inter-node Bandwidth: 6 GB/sec (FDR)

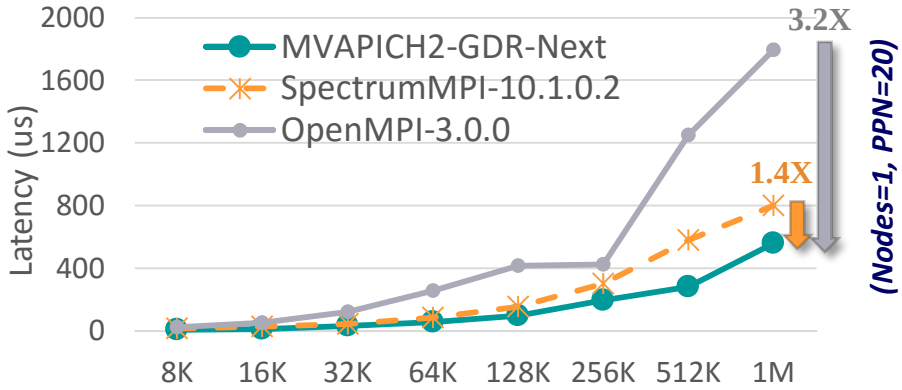
Available in MVAPICH2-GDR 2.3a

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

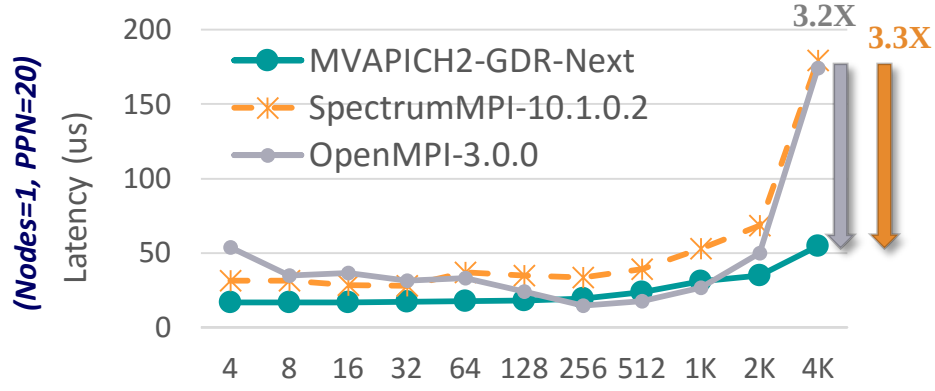
Scalable Host-based Collectives with CMA on OpenPOWER (Intra-node Reduce & AlltoAll)

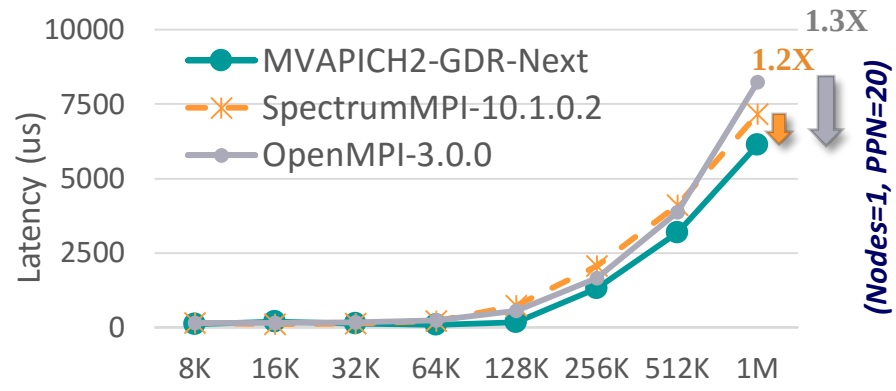
Reduce





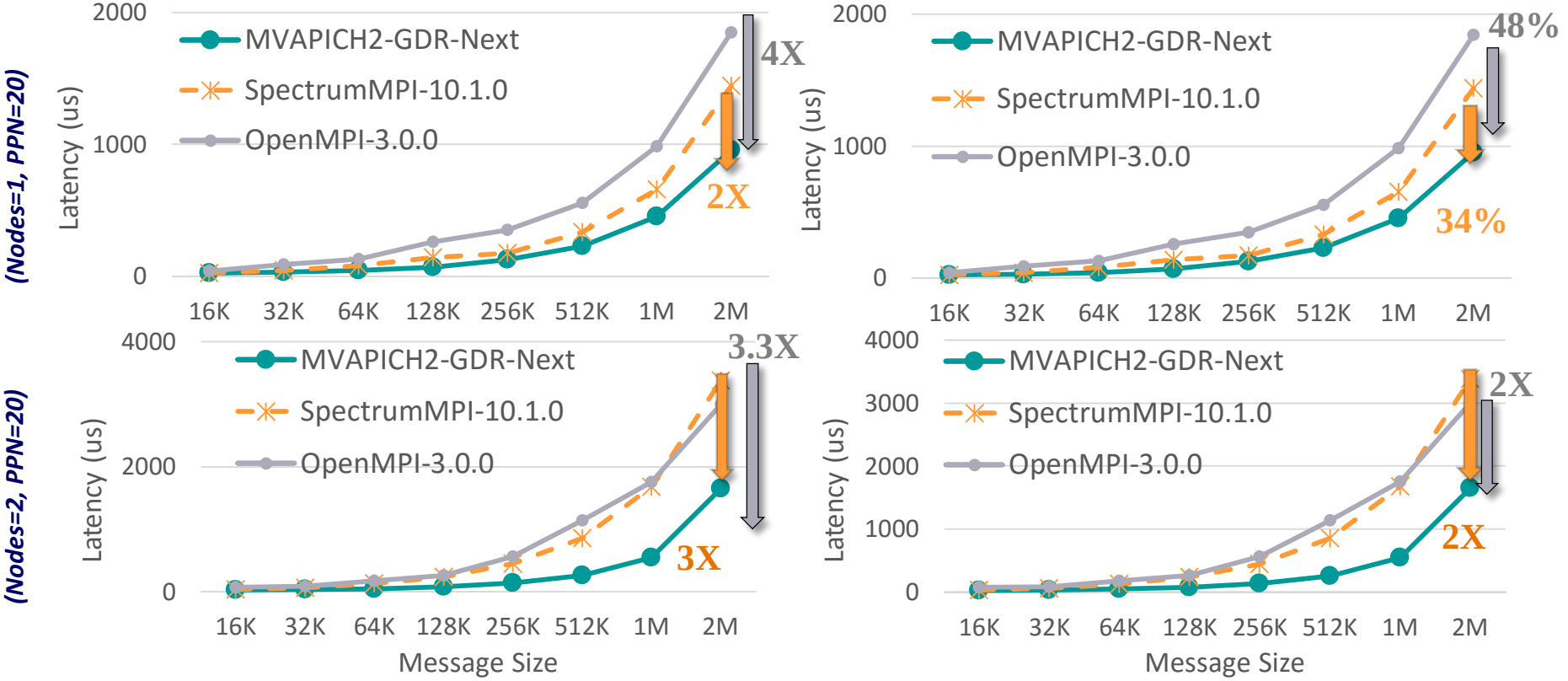
Alltoall





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

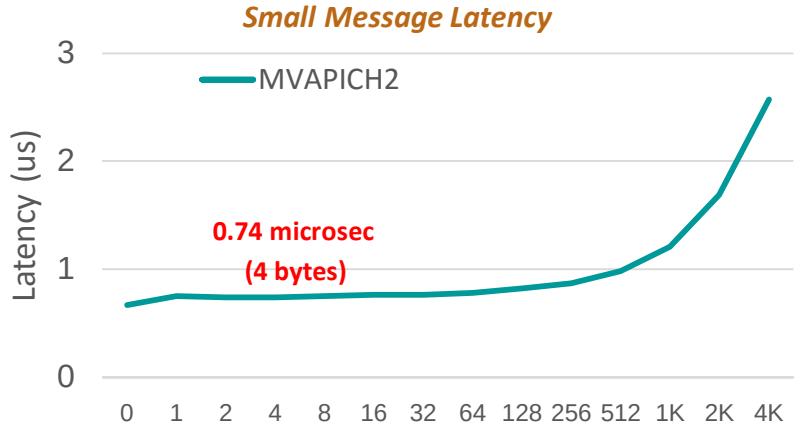
Optimized All-Reduce with XPMEM on OpenPOWER

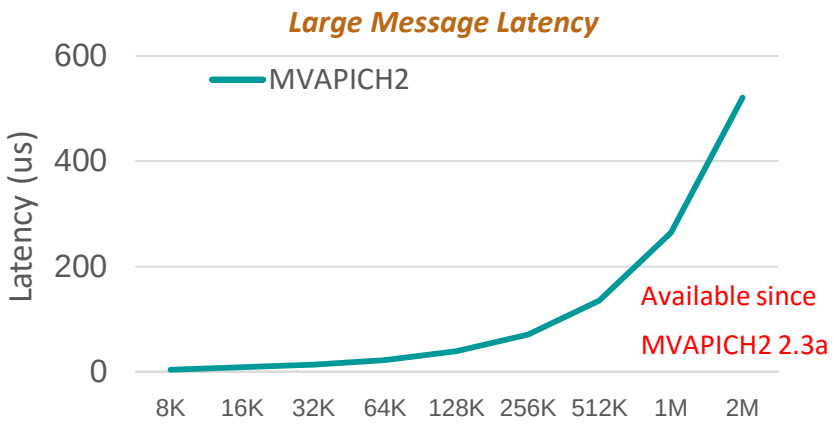


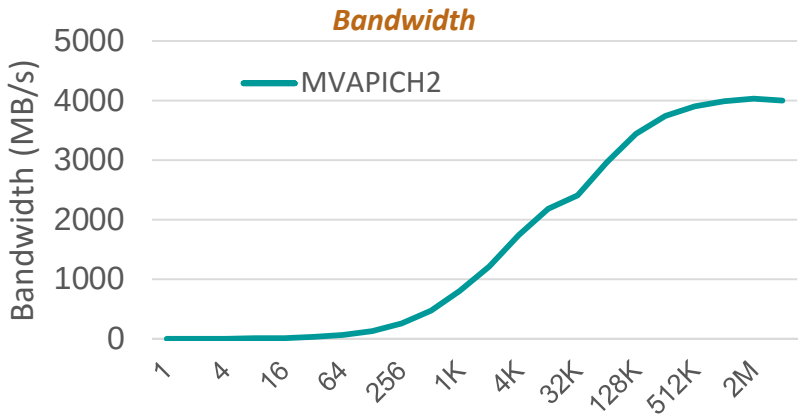
- **Optimized MPI All-Reduce Design in MVAPICH2**
  - ***Up to 2X*** performance improvement over Spectrum MPI and 4X over OpenMPI for intra-node

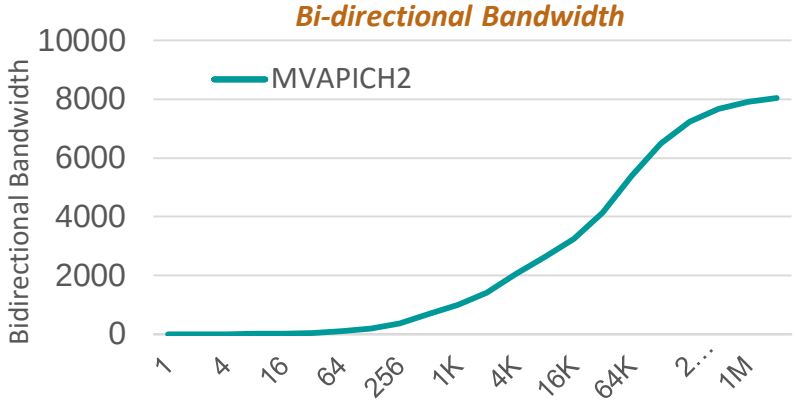
Optimized Runtime Parameters: MV2_CPU_BINDING_POLICY=hybrid MV2_HYBRID_BINDING_POLICY=bunch

Intra-node Point-to-point Performance on ARMv8







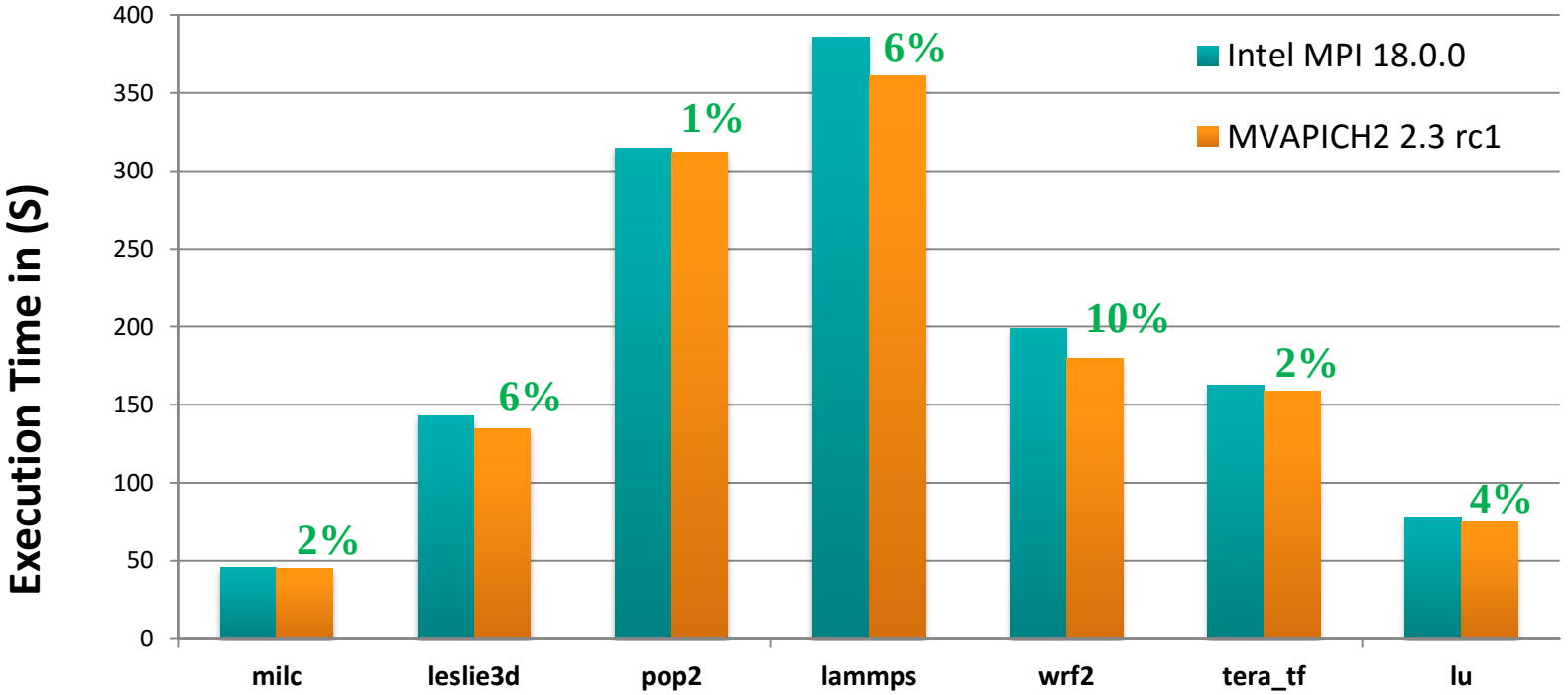


Platform: ARMv8 (aarch64) MIPS processor with 96 cores dual-socket CPU. Each socket contains 48 cores.

Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

- Scalability for million to billion processors
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, UPC++, …)
- Integrated Support for GPGPUs
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- Application Scalability and Best Practices

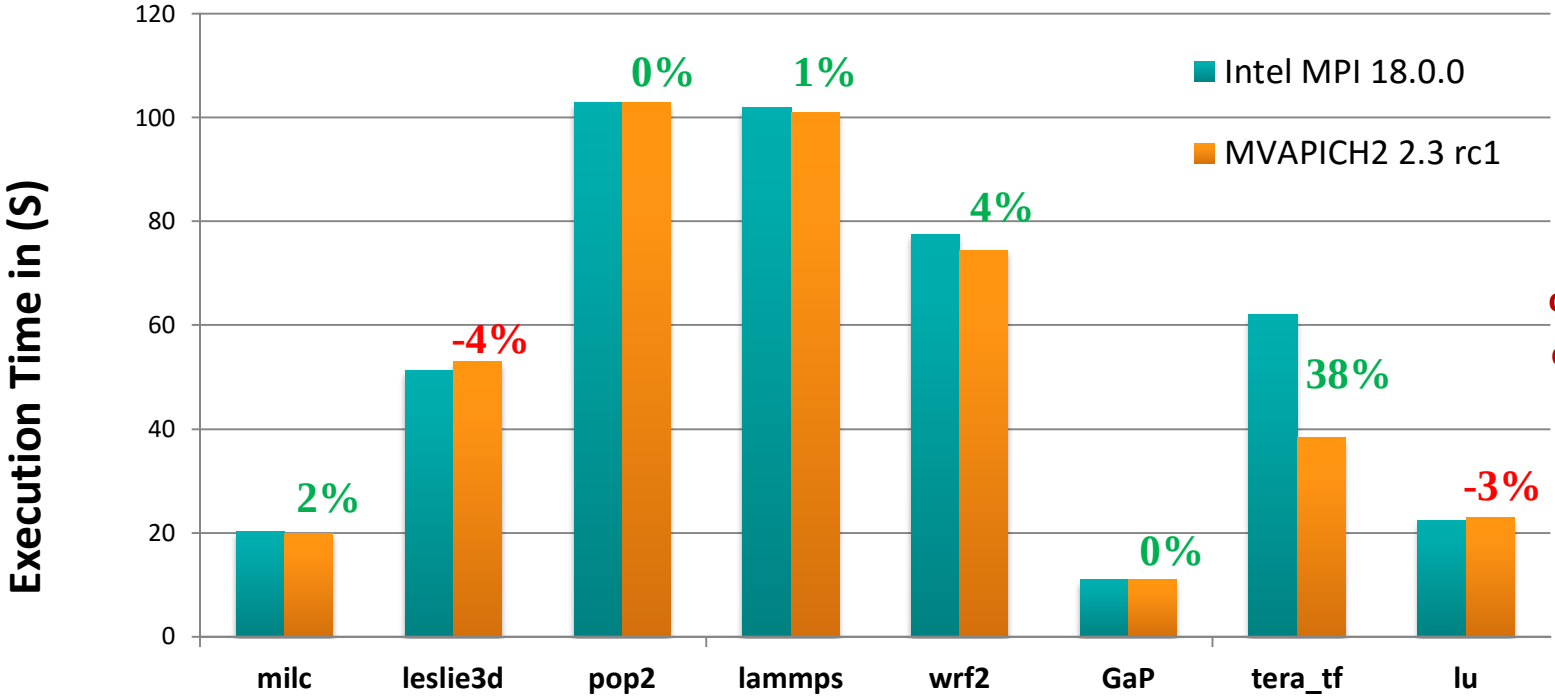
Performance of SPEC MPI 2007 Benchmarks (KNL + Omni-Path)



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%

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.

HPC, Big Data, Deep Learning, and Cloud

- Traditional HPC
  - Message Passing Interface (MPI), including MPI + OpenMP
  - Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
  - Exploiting Accelerators
- Big Data/Enterprise/Commercial Computing
  - Spark and Hadoop (HDFS, HBase, MapReduce)
  - Memcached is also used for Web 2.0
- Deep Learning
  - Caffe, CNTK, TensorFlow, and many more
- Cloud for HPC and BigData
  - Virtualization with SR-IOV and Containers

How Can HPC Clusters with High-Performance Interconnect and Storage Architectures Benefit Big Data Applications?

Can the bottlenecks be alleviated with new designs by taking advantage of HPC technologies?

Can RDMA-enabled high-performance interconnects benefit Big Data processing?

Can HPC Clusters with high-performance storage systems (e.g. SSD, parallel file systems) benefit Big Data applications?

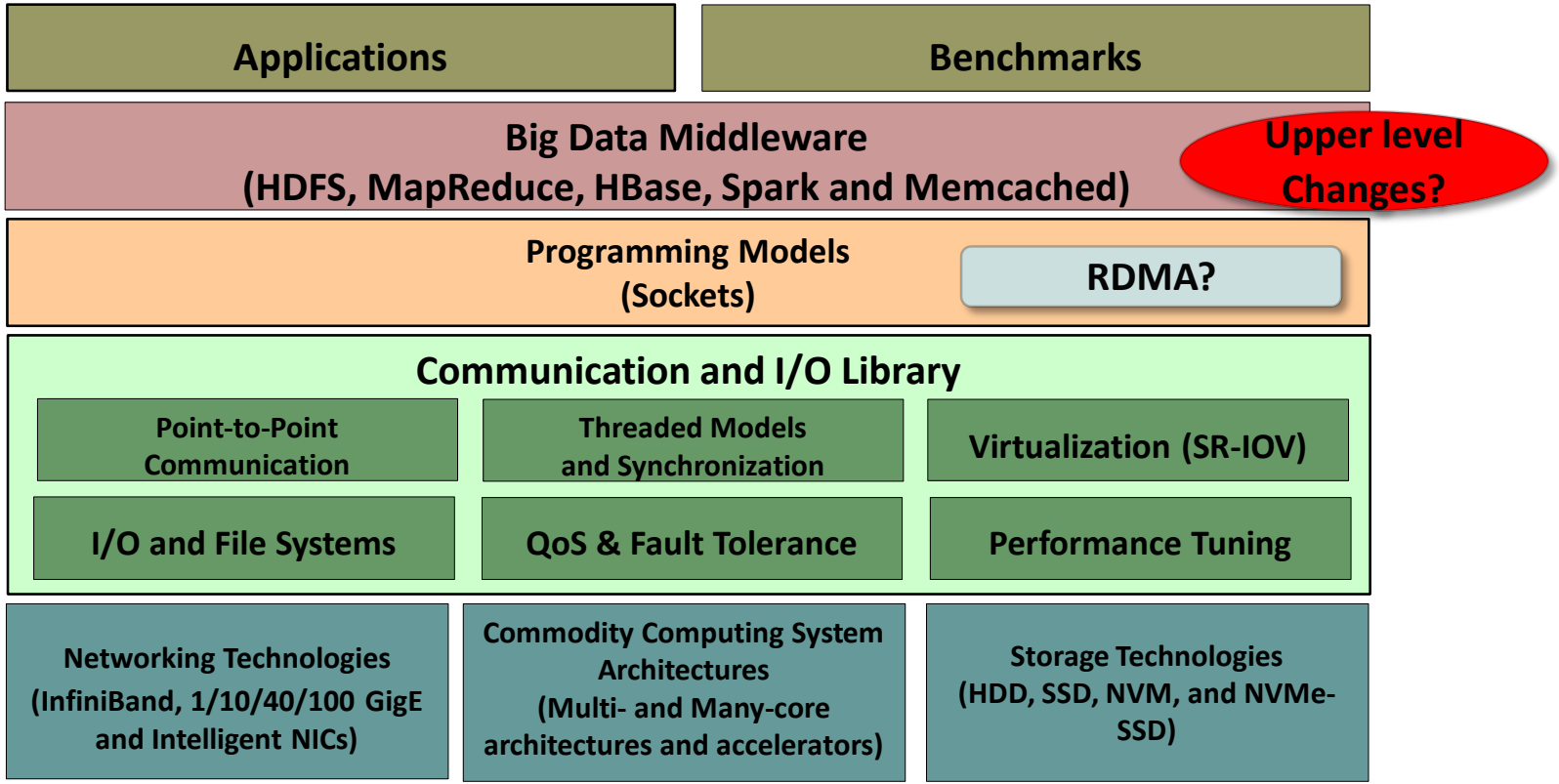
How much performance benefits can be achieved through enhanced designs?

What are the major bottlenecks in current Big Data processing middleware (e.g. Hadoop, Spark, and Memcached)?

How to design benchmarks for evaluating the performance of Big Data middleware on HPC clusters?

Bring HPC and Big Data processing into a "convergent trajectory"!

Designing Communication and I/O Libraries for Big Data Systems: Challenges



The High-Performance Big Data (HiBD) Project

- RDMA for Apache Spark
- RDMA for Apache Hadoop 2.x (RDMA-Hadoop-2.x)
  - Plugins for Apache, Hortonworks (HDP) and Cloudera (CDH) Hadoop distributions
- RDMA for Apache HBase
- RDMA for Memcached (RDMA-Memcached)
- RDMA for Apache Hadoop 1.x (RDMA-Hadoop)
- OSU HiBD-Benchmarks (OHB)
  - HDFS, Memcached, HBase, and Spark Micro-benchmarks
- http://hibd.cse.ohio-state.edu
- Users Base: 275 organizations from 34 countries
- More than 25,550 downloads from the project site

Available for InfiniBand and RoCE

Also run on Ethernet

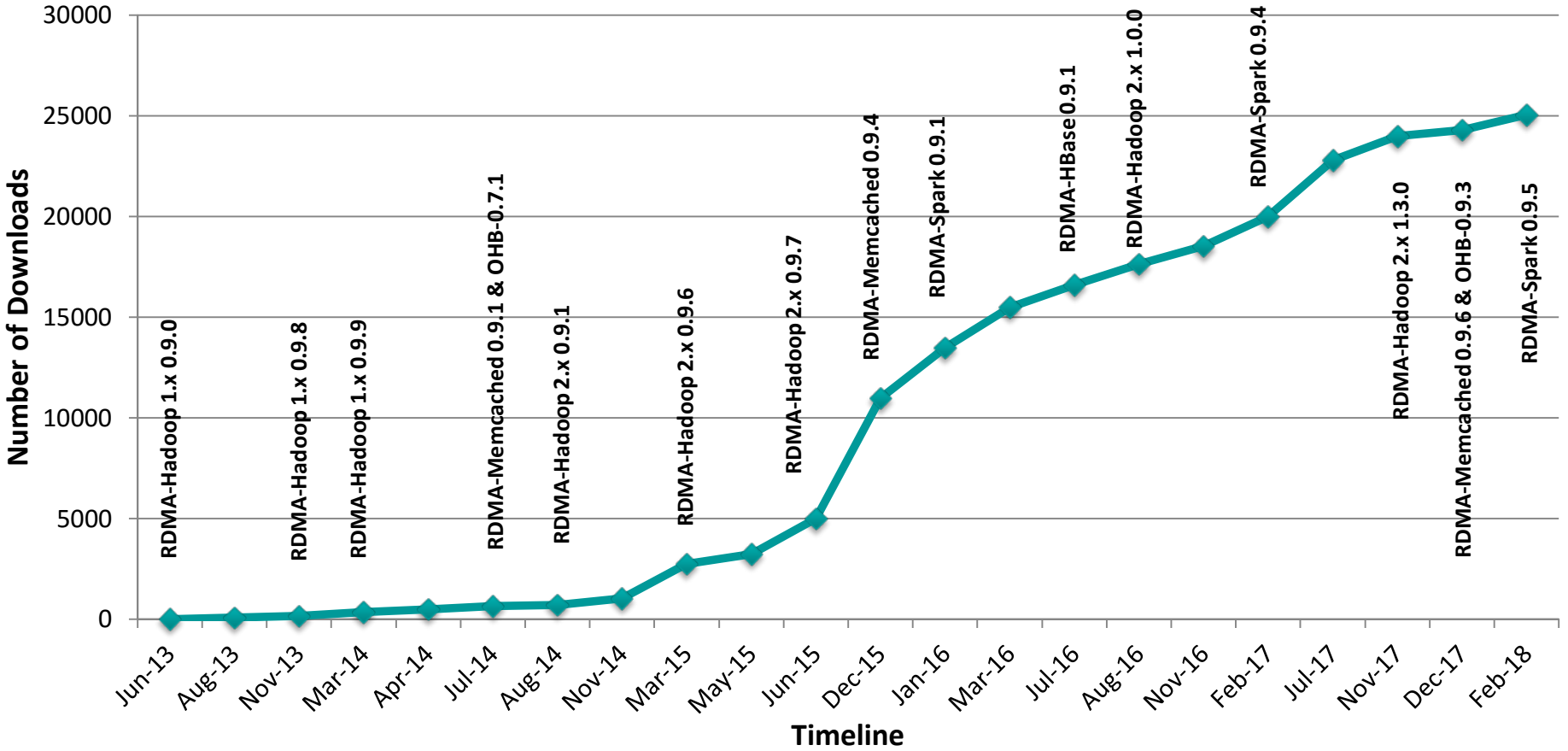
Available for x86 and OpenPOWER

HiBD
High-Performance Big Data

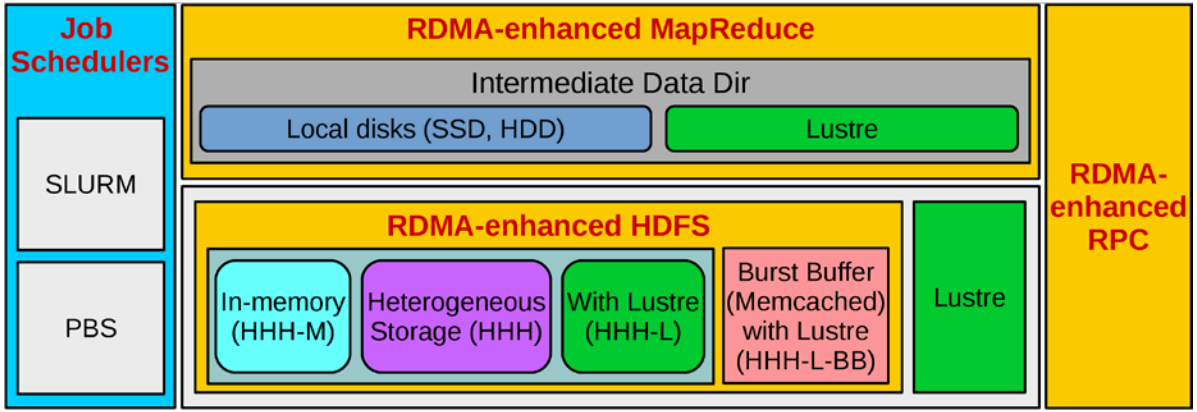
Network Based Computing Laboratory

The Ohio State University

HiBD Release Timeline and Downloads

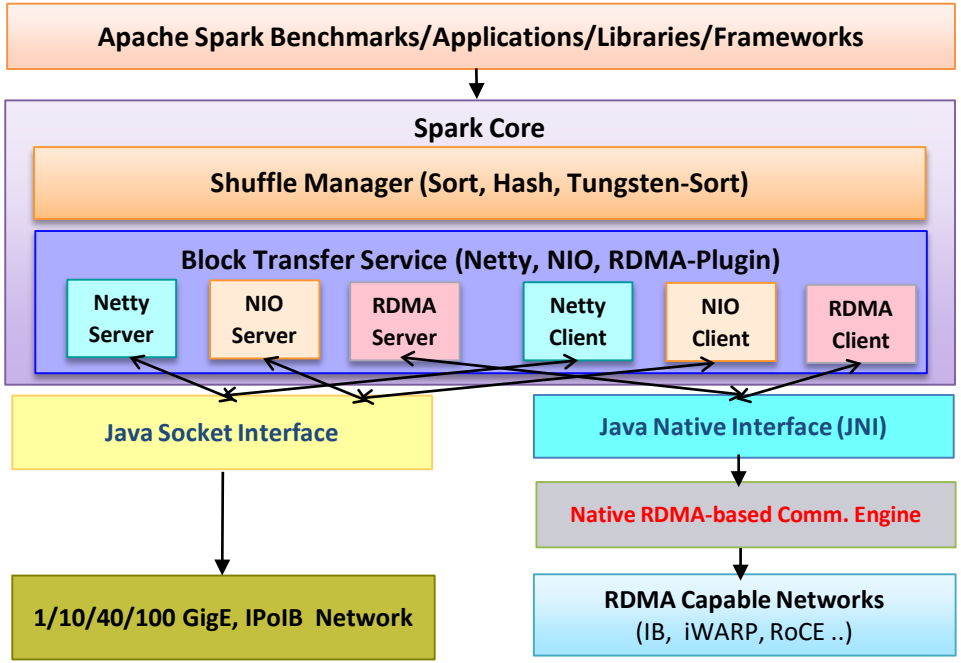


Different Modes of RDMA for Apache Hadoop 2.x



- **HHH**: Heterogeneous storage devices with hybrid replication schemes are supported in this mode of operation to have better fault-tolerance as well as performance. This mode is enabled by **default** in the package.
- **HHH-M**: A high-performance in-memory based setup has been introduced in this package that can be utilized to perform all I/O operations in-memory and obtain as much performance benefit as possible.
- **HHH-L**: With parallel file systems integrated, HHH-L mode can take advantage of the Lustre available in the cluster.
- **HHH-L-BB**: This mode deploys a Memcached-based burst buffer system to reduce the bandwidth bottleneck of shared file system access. The burst buffer design is hosted by Memcached servers, each of which has a local SSD.
- **MapReduce over Lustre, with/without local disks**: Besides, HDFS based solutions, this package also provides support to run MapReduce jobs on top of Lustre alone. Here, two different modes are introduced: with local disks and without local disks.
- **Running with Slurm and PBS**: Supports deploying RDMA for Apache Hadoop 2.x with Slurm and PBS in different running modes (HHH, HHH-M, HHH-L, and MapReduce over Lustre).

Design Overview of Spark with RDMA



- Design Features
  - RDMA based shuffle plugin
  - SEDA-based architecture
  - Dynamic connection management and sharing
  - Non-blocking data transfer
  - Off-JVM-heap buffer management
  - InfiniBand/RoCE support

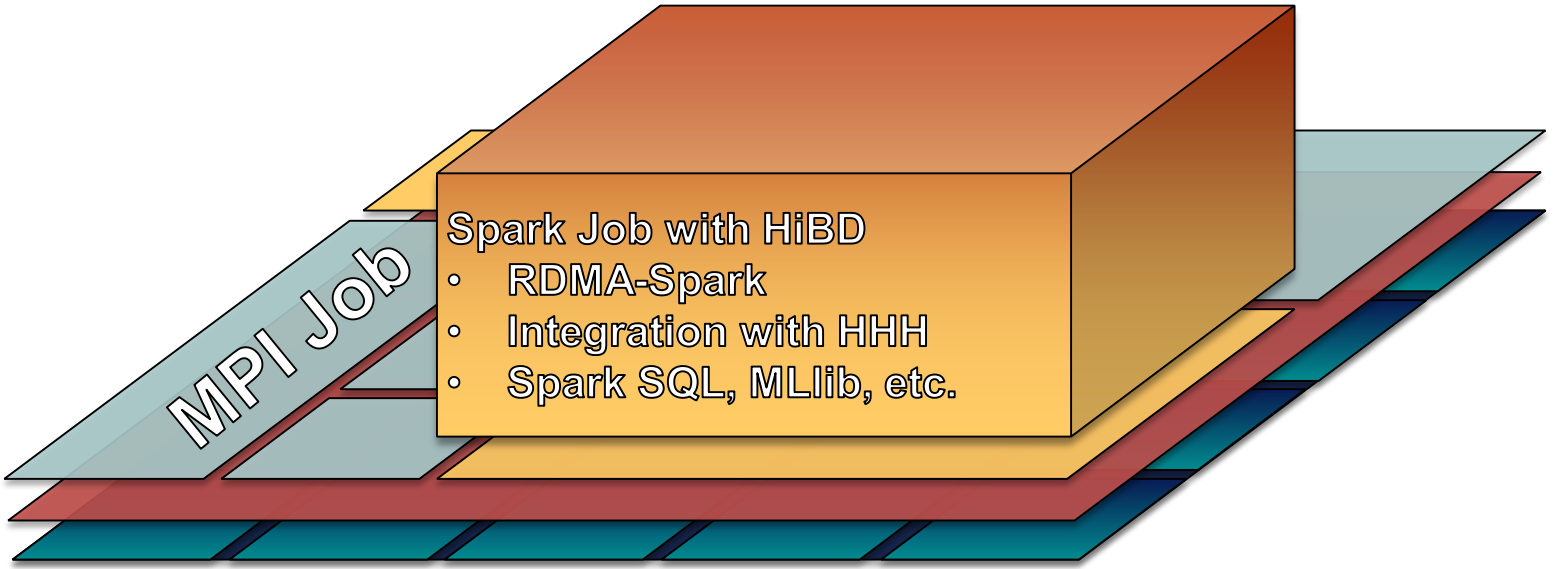
- Enables high performance RDMA communication, while supporting traditional socket interface
- JNI Layer bridges Scala based Spark with communication library written in native code

X. Lu, M. W. Rahman, N. Islam, D. Shankar, and D. K. Panda, Accelerating Spark with RDMA for Big Data Processing: Early Experiences, Int'l Symposium on High Performance Interconnects (HotI'14), August 2014

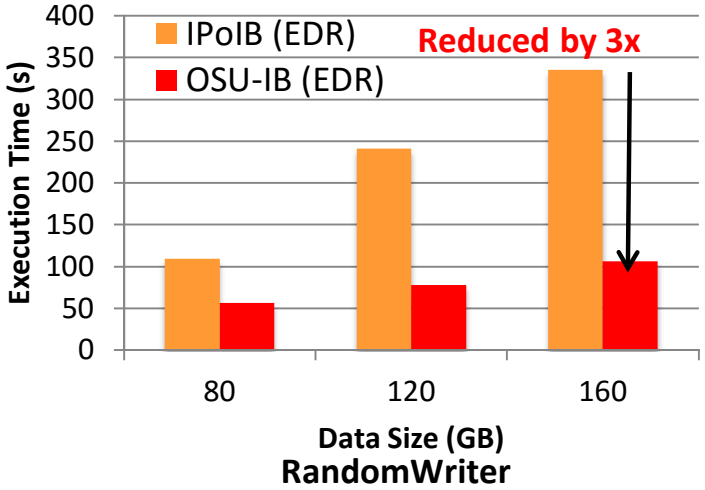
X. Lu, D. Shankar, S. Gugnani, and D. K. Panda, High-Performance Design of Apache Spark with RDMA and Its Benefits on Various Workloads, IEEE BigData '16, Dec. 2016.

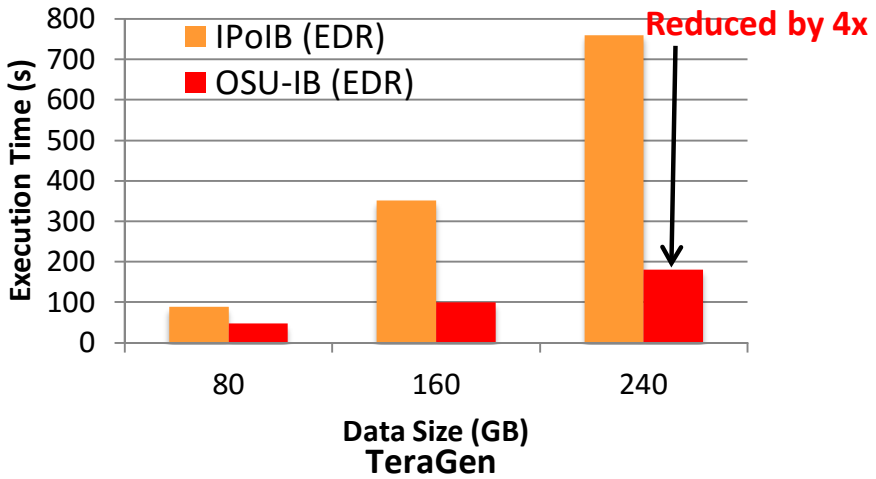
Using HiBD Packages for Big Data Processing on Existing HPC Infrastructure

Using HiBD Packages for Big Data Processing on Existing HPC Infrastructure



Performance Numbers of RDMA for Apache Hadoop 2.x – RandomWriter & TeraGen in OSU-RI2 (EDR)

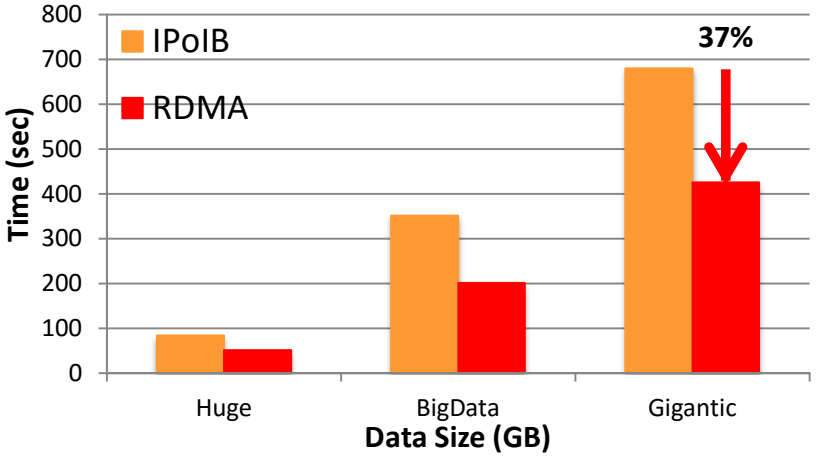




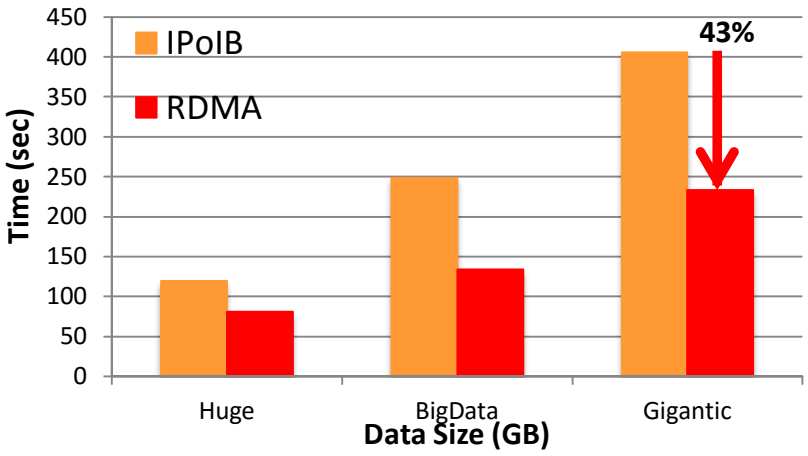
Cluster with 8 Nodes with a total of 64 maps

- RandomWriter
  - **3x** improvement over IPoIB for 80-160 GB file size
- TeraGen
  - **4x** improvement over IPoIB for 80-240 GB file size

Performance Evaluation of RDMA-Spark on SDSC Comet – HiBench PageRank



32 Worker Nodes, 768 cores, PageRank Total Time

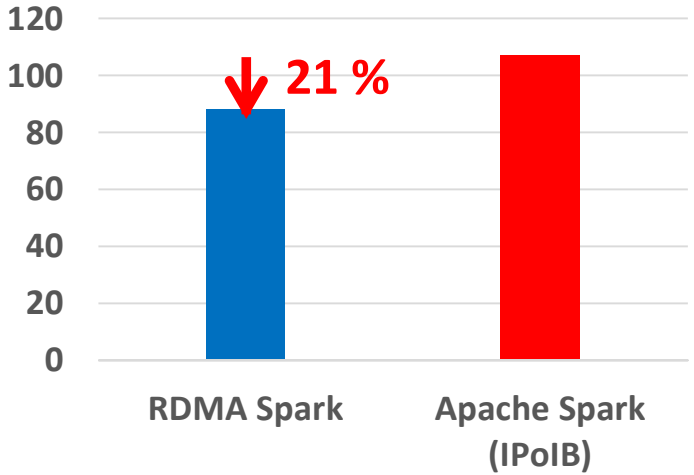


64 Worker Nodes, 1536 cores, PageRank Total Time

- InfiniBand FDR, SSD, 32/64 Worker Nodes, 768/1536 Cores, (768/1536M 768/1536R)
- RDMA vs. IPoIB with 768/1536 concurrent tasks, single SSD per node.
  - 32 nodes/768 cores: Total time reduced by 37% over IPoIB (56Gbps)
  - 64 nodes/1536 cores: Total time reduced by 43% over IPoIB (56Gbps)

Performance Evaluation on SDSC Comet: Astronomy Application

- **Kira Toolkit[1]:** Distributed astronomy image processing toolkit implemented using Apache Spark.
- Source extractor application, using a 65GB dataset from the SDSS DR2 survey that comprises 11,150 image files.
- Compare RDMA Spark performance with the standard apache implementation using IPoIB.



Execution times (sec) for Kira SE benchmark using 65 GB dataset, 48 cores.

1. Z. Zhang, K. Barbary, F. A. Nothaft, E.R. Sparks, M.J. Franklin, D.A. Patterson, S. Perlmutter. Scientific Computing meets Big Data Technology: An Astronomy Use Case. *CoRR, vol: abs/1507.03325*, Aug 2015.

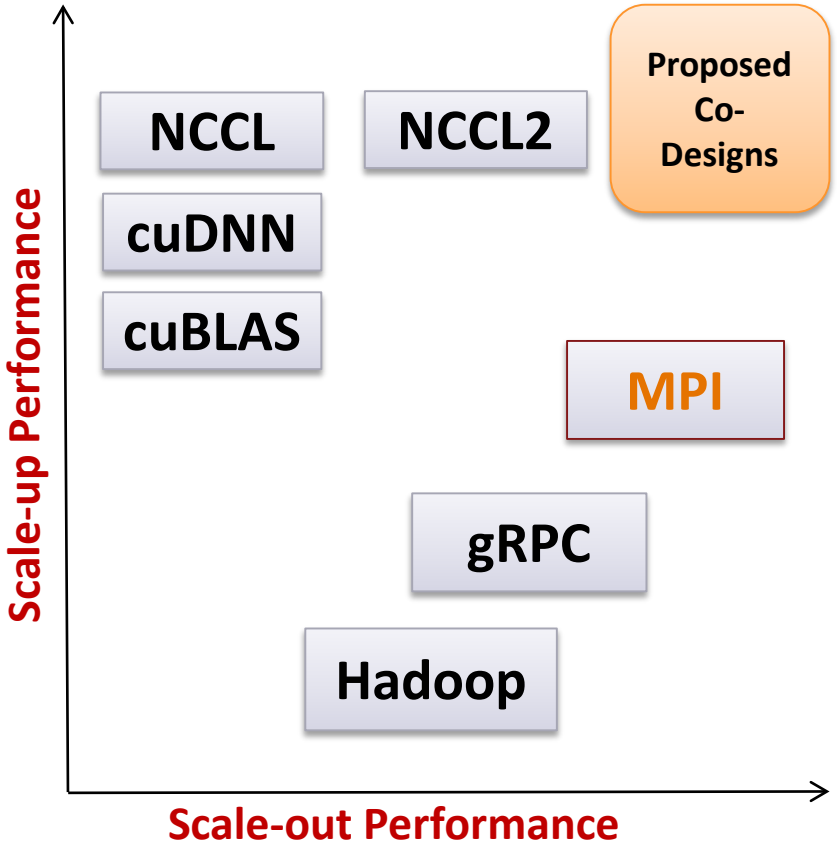
M. Tatineni, X. Lu, D. J. Choi, A. Majumdar, and D. K. Panda, Experiences and Benefits of Running RDMA Hadoop and Spark on SDSC Comet, XSEDE'16, July 2016

HPC, Big Data, Deep Learning, and Cloud

- Traditional HPC
  - Message Passing Interface (MPI), including MPI + OpenMP
  - Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
  - Exploiting Accelerators
- Big Data/Enterprise/Commercial Computing
  - Spark and Hadoop (HDFS, HBase, MapReduce)
  - Memcached is also used for Web 2.0
- Deep Learning
  - Caffe, CNTK, TensorFlow, and many more
- Cloud for HPC and BigData
  - Virtualization with SR-IOV and Containers

Deep Learning: New Challenges for MPI Runtimes

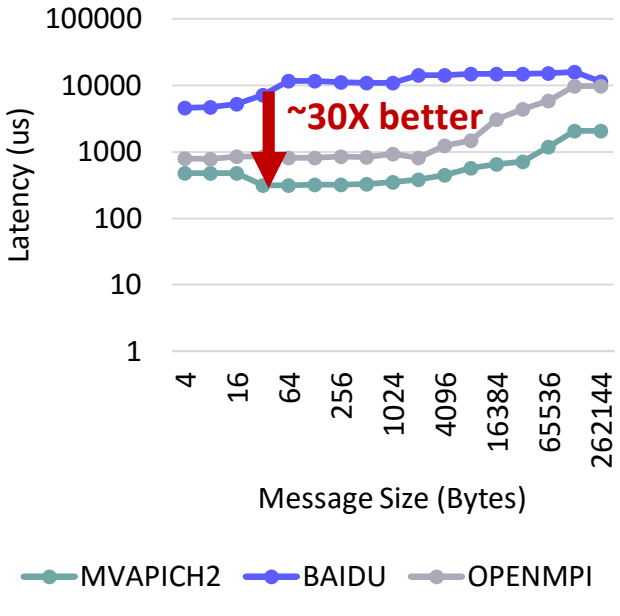
- Deep Learning frameworks are a different game altogether
  - Unusually large message sizes (order of megabytes)
  - Most communication based on GPU buffers
- Existing State-of-the-art
  - cuDNN, cuBLAS, NCCL --> **scale-up** performance
  - NCCL2, CUDA-Aware MPI --> **scale-out** performance
    - 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**?

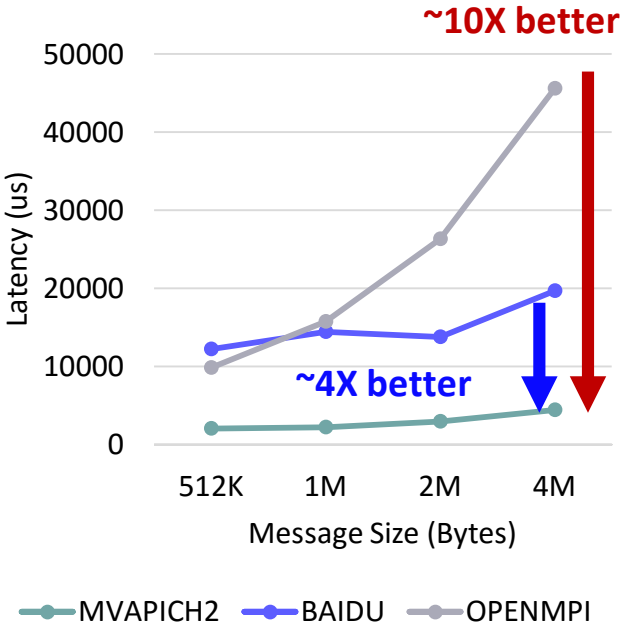


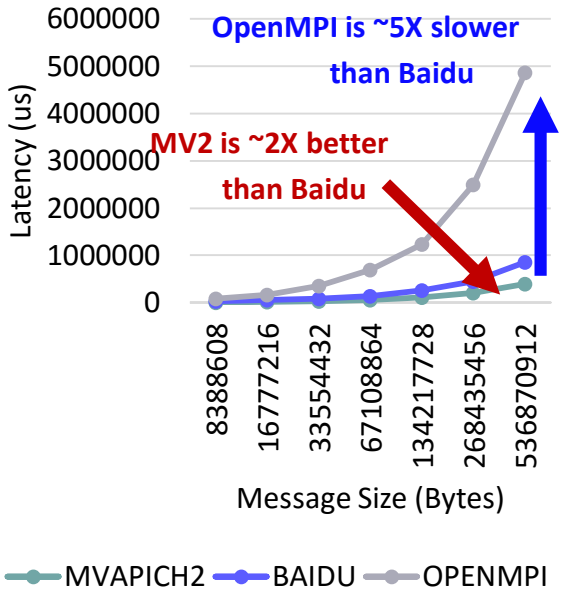
A. A. Awan, K. Hamidouche, J. M. Hashmi, and D. K. Panda, S-Caffe: Co-designing MPI Runtimes and Caffe for Scalable Deep Learning on Modern GPU Clusters. In *Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (PPoPP '17)

MVAPICH2: Allreduce Comparison with Baidu and OpenMPI

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



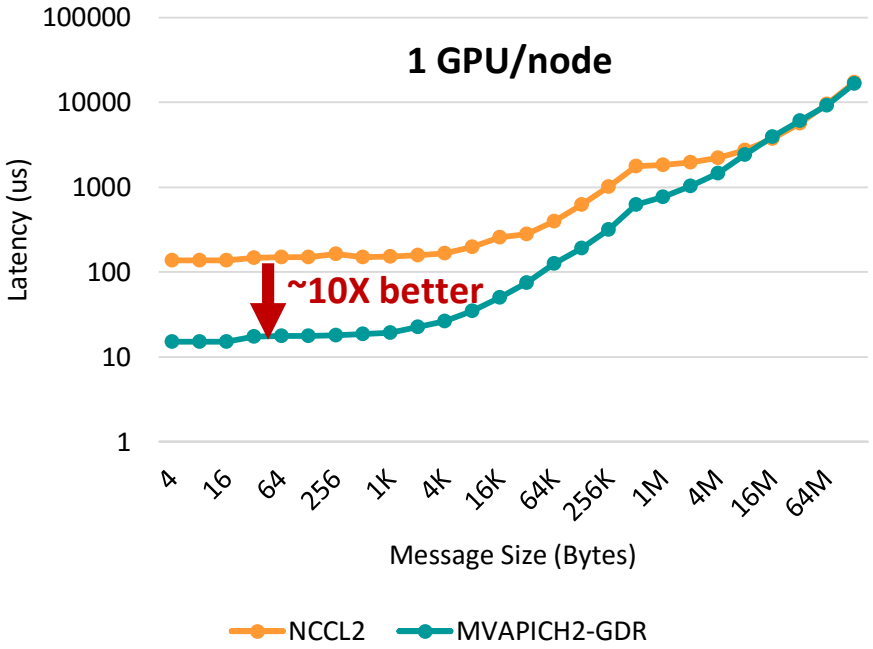


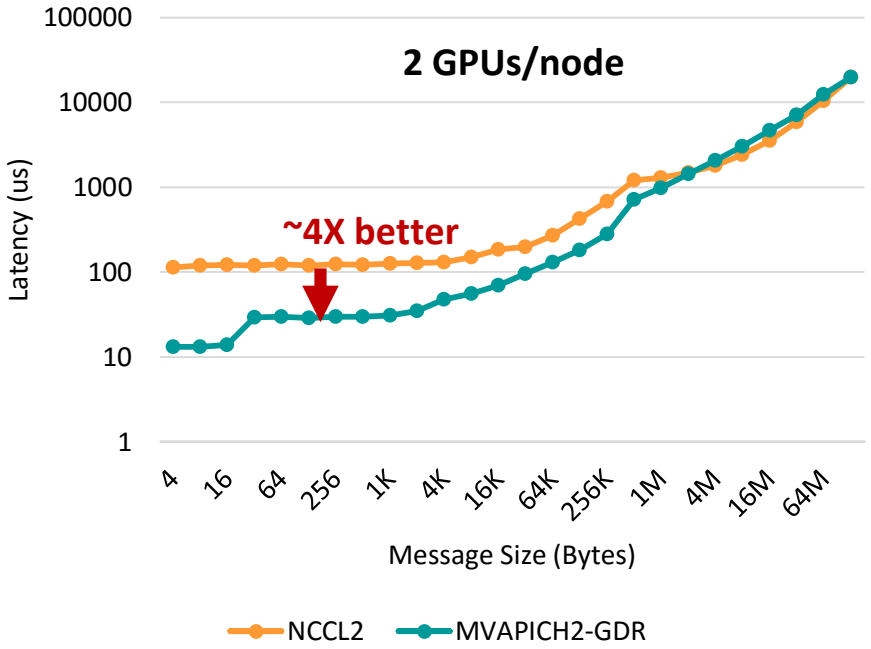


**Available with MVAPICH2-GDR 2.3a*

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



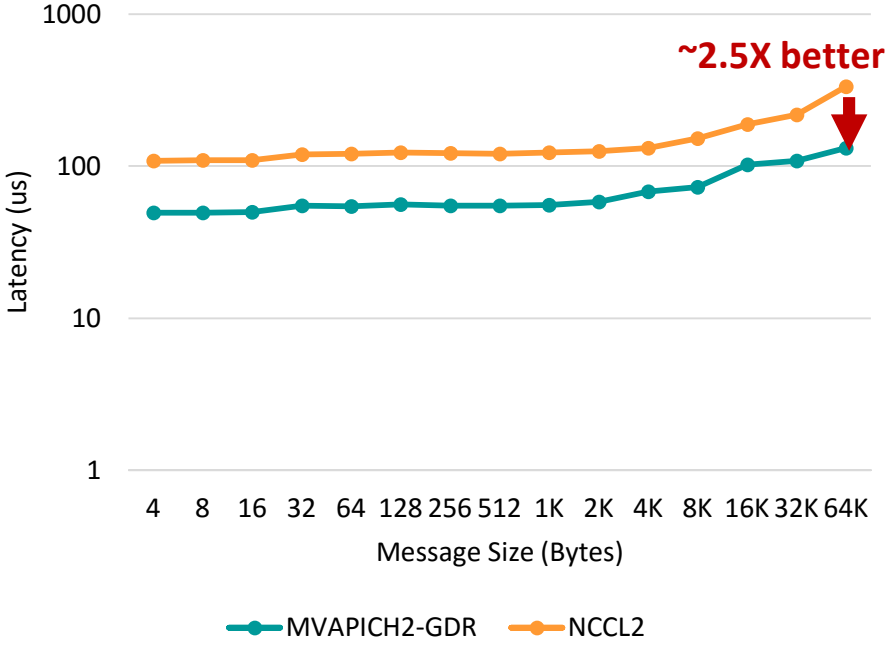


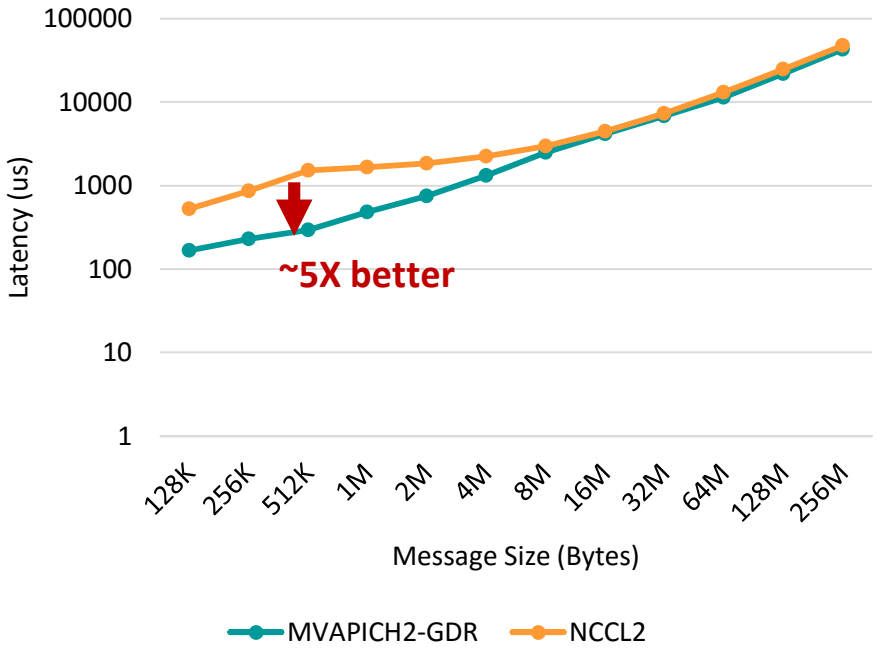
*Will be available with upcoming MVAPICH2-GDR 2.3b

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



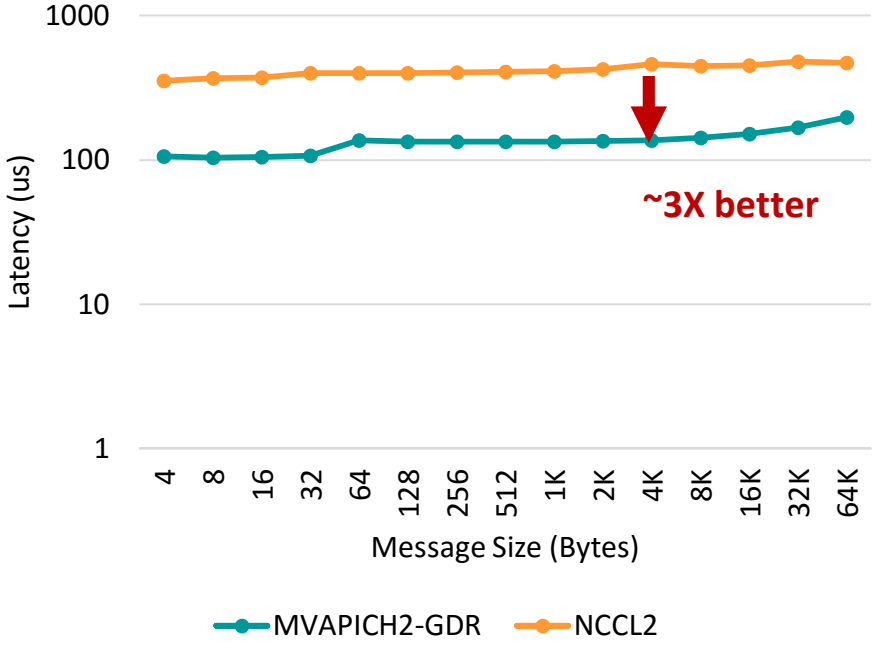


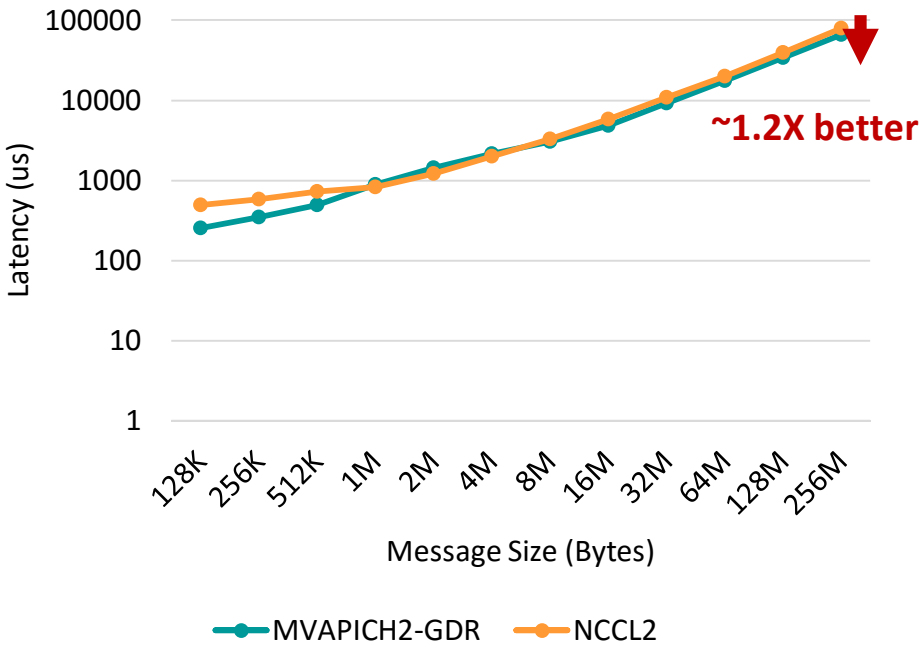
*Will be available with upcoming MVAPICH2-GDR 2.3b

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

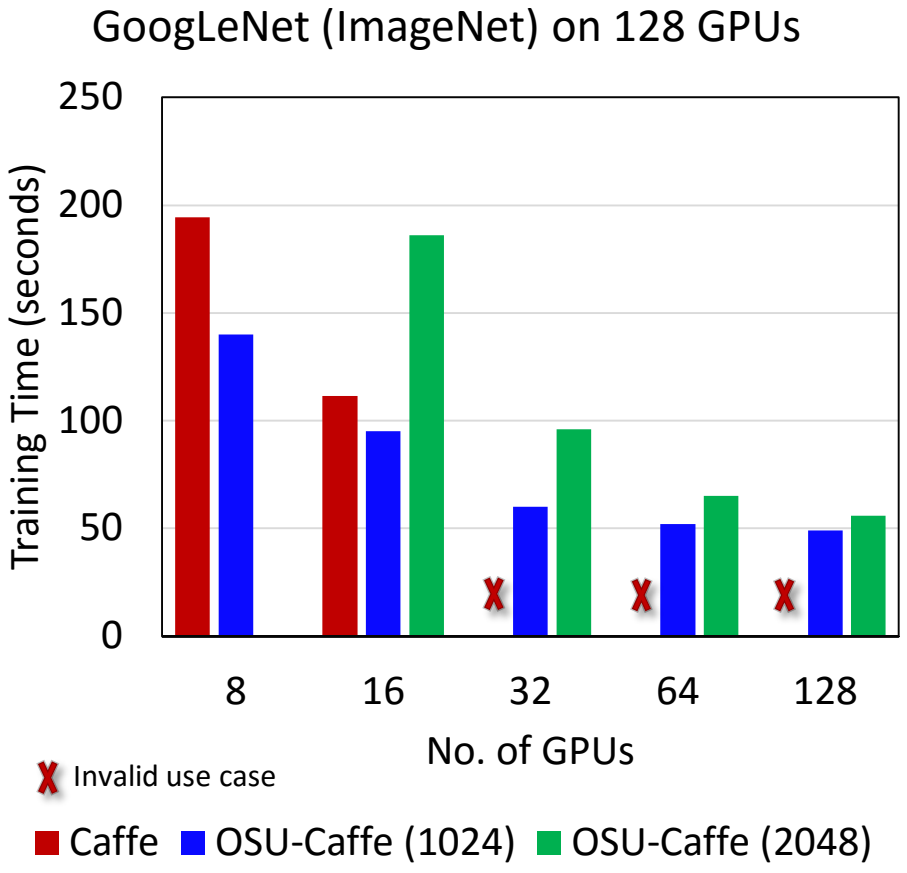
OSU-Caffe: Scalable Deep Learning

- Caffe : A flexible and layered Deep Learning framework.
- Benefits and Weaknesses
  - Multi-GPU Training within a single node
  - Performance degradation for GPUs across different sockets
  - Limited Scale-out
- OSU-Caffe: MPI-based Parallel Training
  - Enable Scale-up (within a node) and Scale-out (across multi-GPU nodes)
  - Scale-out on 64 GPUs for training CIFAR-10 network on CIFAR-10 dataset
  - Scale-out on 128 GPUs for training GoogLeNet network on ImageNet dataset

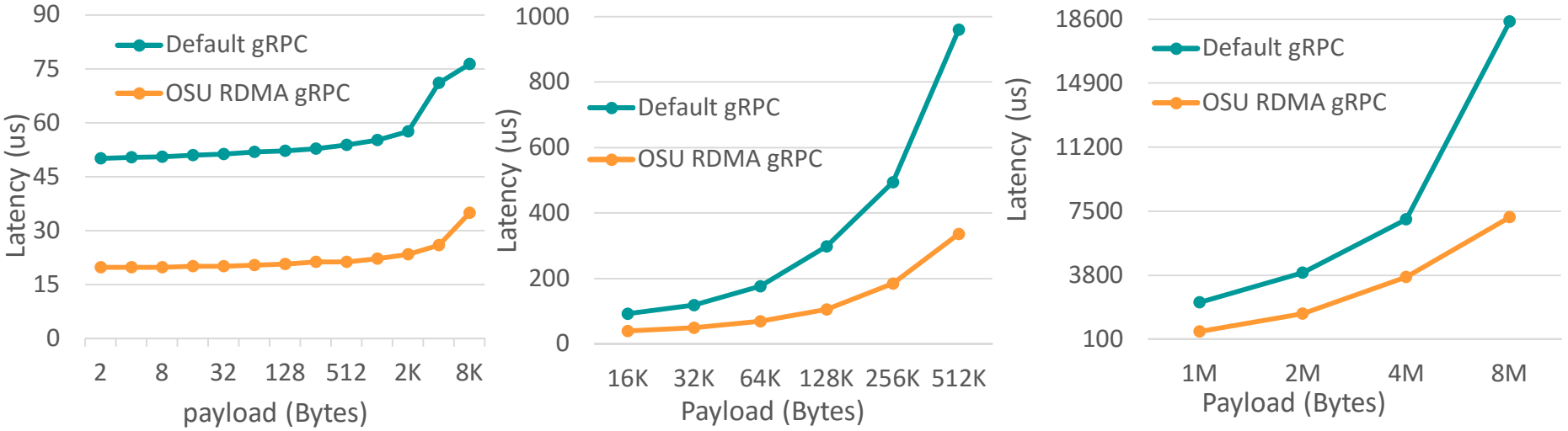
OSU-Caffe publicly available from

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

Support on OPENPOWER will be available soon



Performance Benefits for RDMA-gRPC with Micro-Benchmark

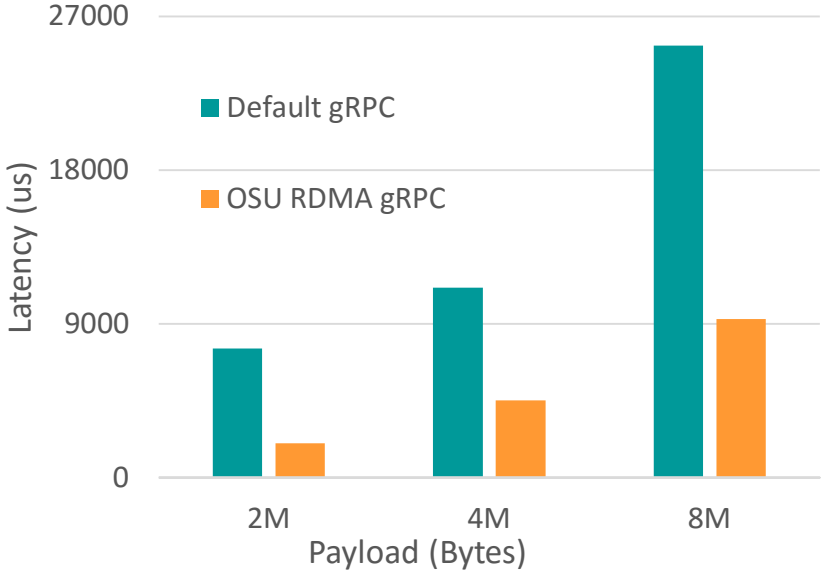


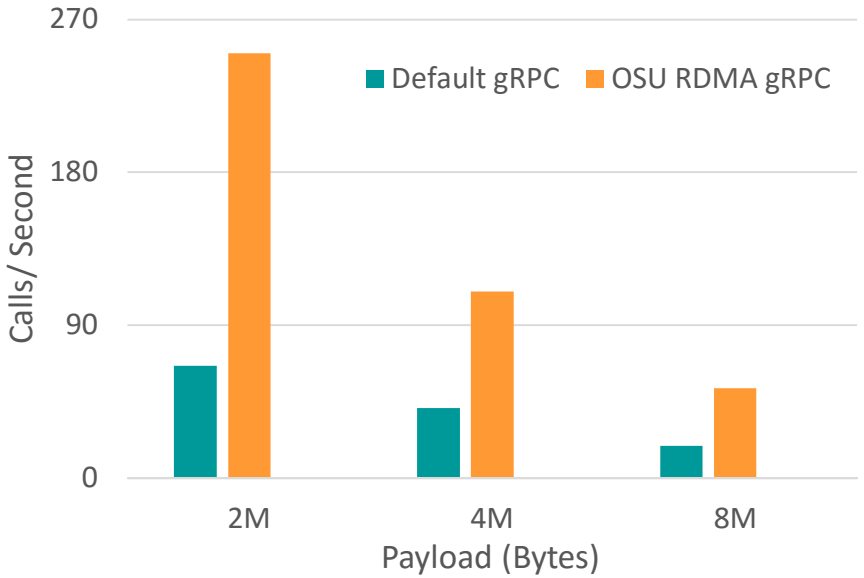
RDMA-gRPC RPC Latency

- **gRPC-RDMA Latency on SDSC-Comet-FDR**
  - **Up to 2.7x** performance speedup over IPoIB for Latency for small messages
  - **Up to 2.8x** performance speedup over IPoIB for Latency for medium messages
  - **Up to 2.5x** performance speedup over IPoIB for Latency for large messages

R. Biswas, X. Lu, and D. K. Panda, Accelerating gRPC and TensorFlow with RDMA for High-Performance Deep Learning over InfiniBand, Under Review.

Performance Benefits for RDMA-gRPC with TensorFlow Communication Mimic Benchmark



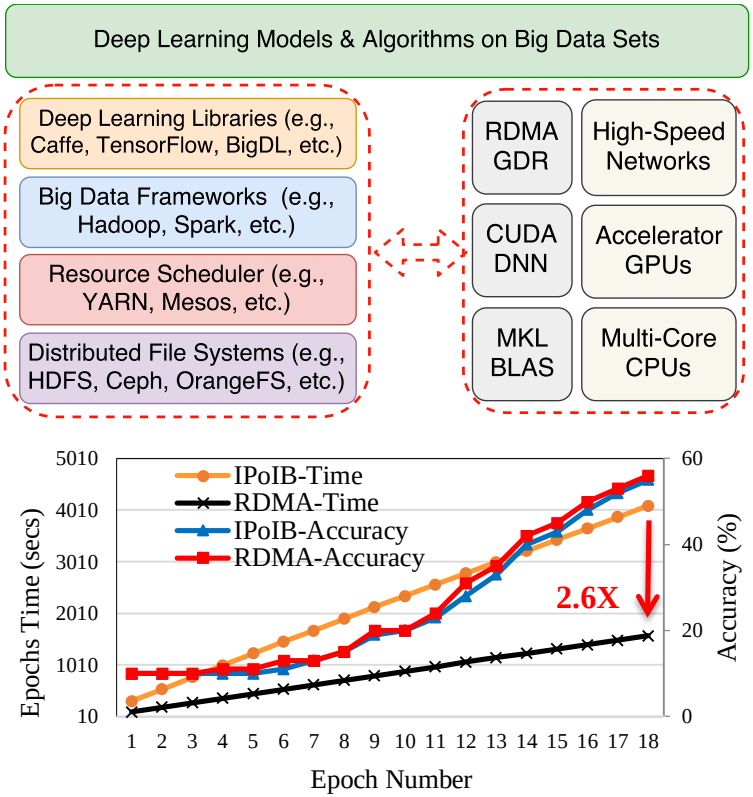


TensorFlow communication Pattern mimic benchmark

- **TensorFlow communication pattern mimic on SDSC-Comet-FDR**
  - Single process spawns both gRPC server and gRPC client
  - **Up to 3.6x** performance speedup over IPoIB for Latency
  - **Up to 3.7x** throughput improvement over IPoIB

High-Performance Deep Learning over Big Data (DLoBD) Stacks

- Benefits of Deep Learning over Big Data (DLoBD)
  - Easily integrate deep learning components into Big Data processing workflow
  - Easily access the stored data in Big Data systems
  - No need to set up new dedicated deep learning clusters; Reuse existing big data analytics clusters
- Challenges
  - Can RDMA-based designs in DLoBD stacks improve performance, scalability, and resource utilization on high-performance interconnects, GPUs, and multi-core CPUs?
  - What are the performance characteristics of representative DLoBD stacks on RDMA networks?
- Characterization on DLoBD Stacks
  - CaffeOnSpark, TensorFlowOnSpark, and BigDL
  - IPoIB vs. RDMA; In-band communication vs. Out-of-band communication; CPU vs. GPU; etc.
  - Performance, accuracy, scalability, and resource utilization
  - RDMA-based DLoBD stacks (e.g., BigDL over RDMA-Spark) can achieve 2.6x speedup compared to the IPoIB based scheme, while maintain similar accuracy



X. Lu, H. Shi, M. H. Javed, R. Biswas, and D. K. Panda, Characterizing Deep Learning over Big Data (DLoBD) Stacks on RDMA-capable Networks, HotI 2017.

HPC, Big Data, Deep Learning, and Cloud

- Traditional HPC
  - Message Passing Interface (MPI), including MPI + OpenMP
  - Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
  - Exploiting Accelerators
- Big Data/Enterprise/Commercial Computing
  - Spark and Hadoop (HDFS, HBase, MapReduce)
  - Memcached is also used for Web 2.0
- Deep Learning
  - Caffe, CNTK, TensorFlow, and many more
- Cloud for HPC and BigData
  - Virtualization with SR-IOV and Containers

Can HPC and Virtualization be Combined?

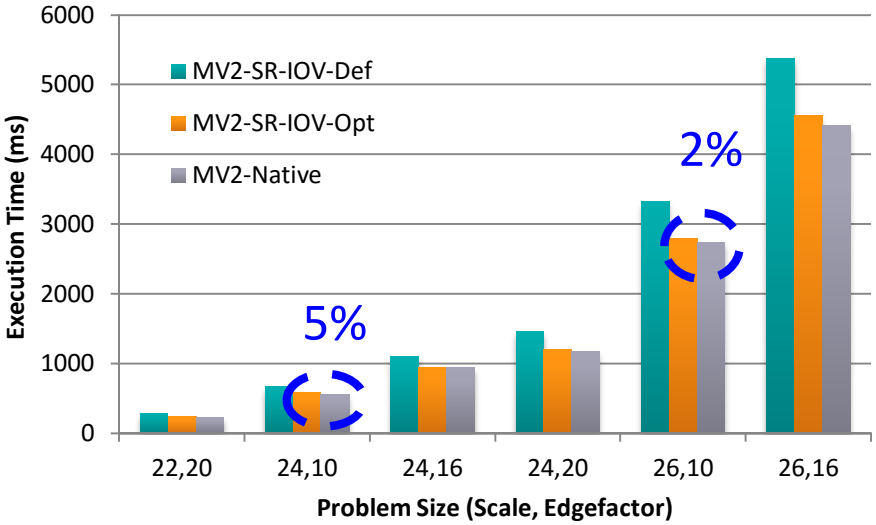
- Virtualization has many benefits
  - Fault-tolerance
  - Job migration
  - Compaction
- Have not been very popular in HPC due to overhead associated with Virtualization
- New SR-IOV (Single Root – IO Virtualization) support available with Mellanox InfiniBand adapters changes the field
- Enhanced MVAPICH2 support for SR-IOV
- MVAPICH2-Virt 2.2 supports:
  - 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

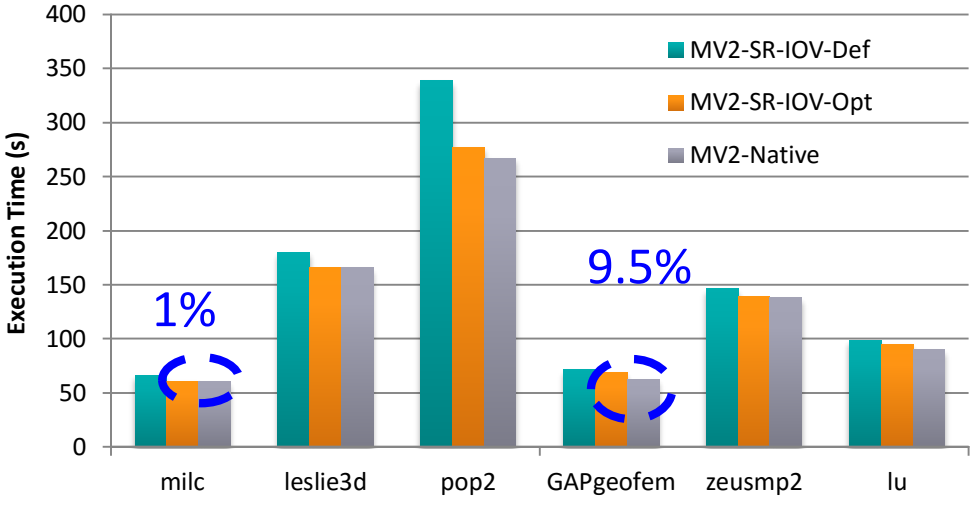
J. Zhang, X. Lu, J. Jose, M. Li, R. Shi and D.K. Panda, High Performance MPI Libray over SR-IOV enabled InfiniBand Clusters, HiPC'14

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



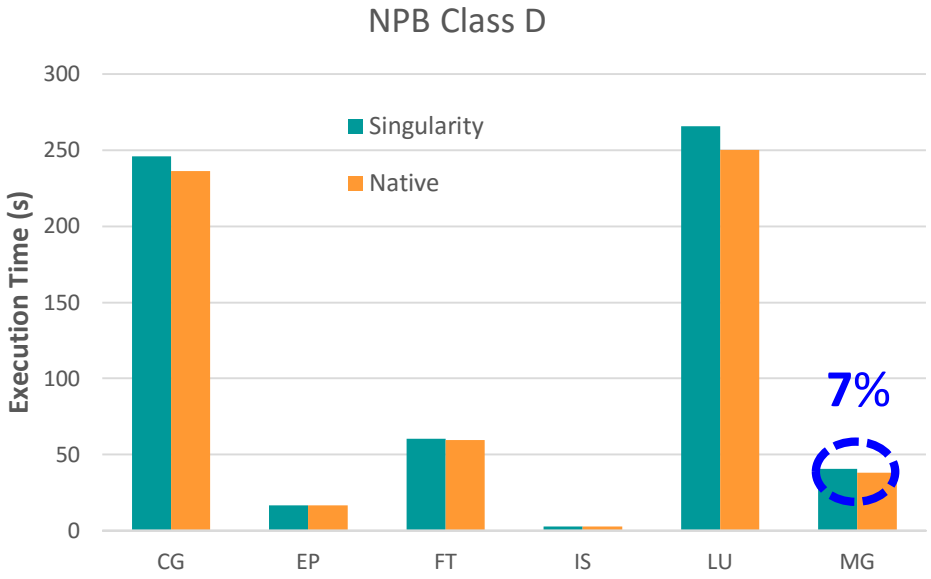
Graph500

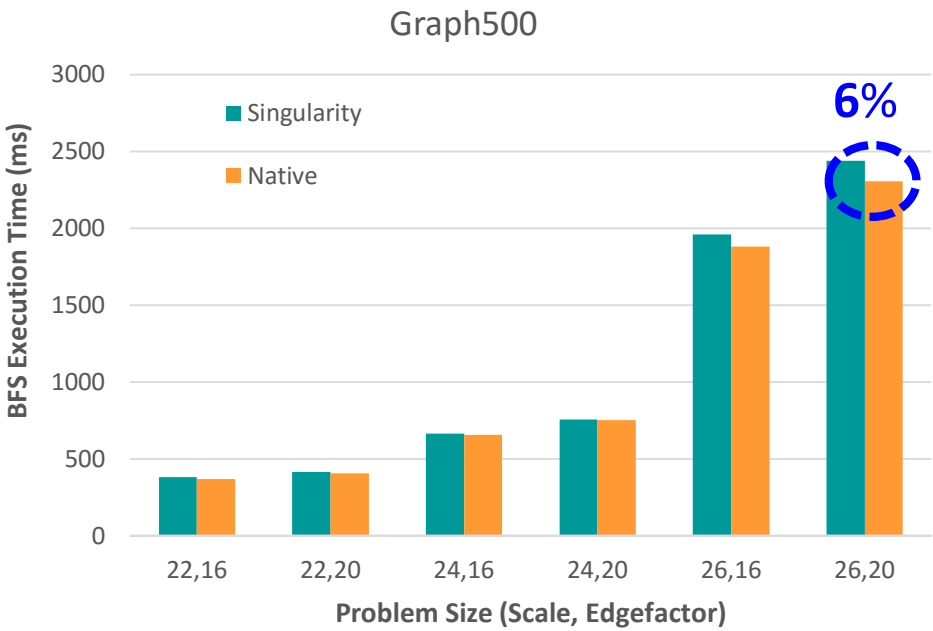


SPEC MPI2007

- 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 Singularity with MVAPICH2

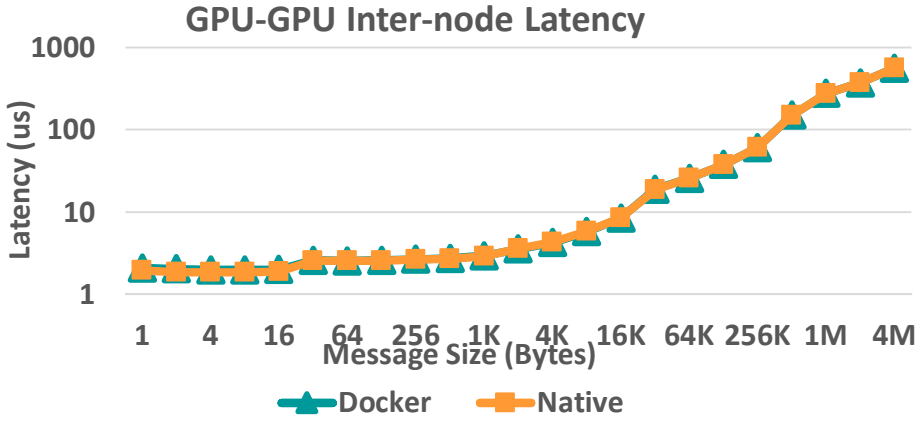


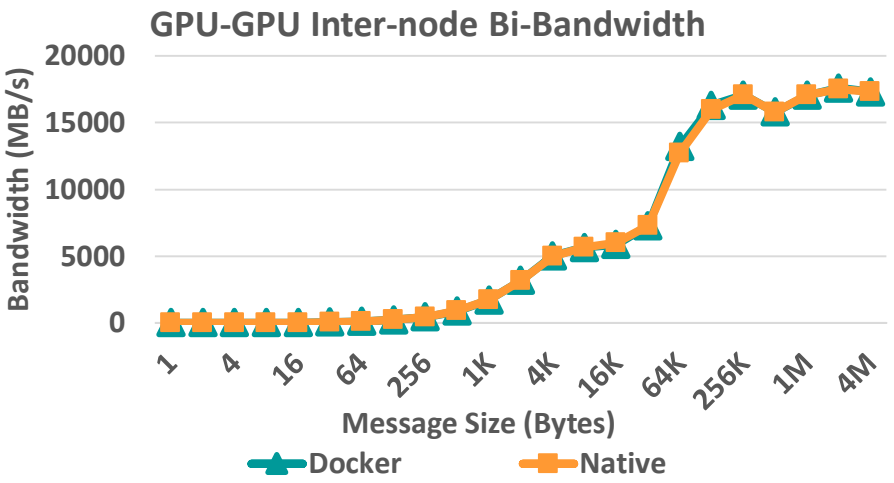


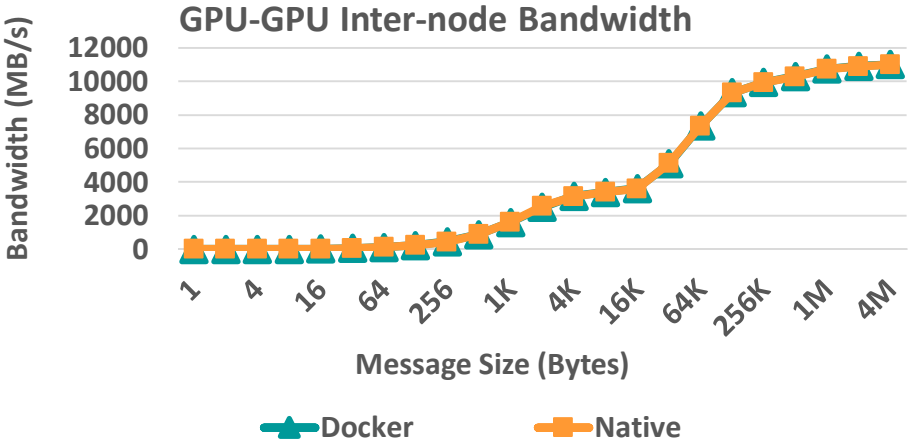
- 512 Processes across 32 nodes
- Less than 7% and 6% overhead for NPB and Graph500, respectively

J. Zhang, X .Lu and D. K. Panda, Is Singularity-based Container Technology Ready for Running MPI Applications on HPC Clouds?, UCC '17, Best Student Paper Award

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

Concluding Remarks

- Next generation HPC systems need to be designed with a holistic view of Big Data, Deep Learning and Cloud
- Presented some of the approaches and results along these directions
- Enable Big Data, Deep Learning and Cloud community to take advantage of modern HPC technologies
- Many other open issues need to be solved
- Next generation HPC professionals need to be trained along these directions

Additional Presentation and Tutorials

- Talk: Exploiting Computation and Communication Overlap in MVAPICH2 and MVAPICH2-GDR MPI Libraries
  - 04/04/18 at 9:00 am at Overlapping Communication with Computation Symposium
- Tutorial: How to Boost the Performance of Your MPI and PGAS Applications with MVAPICH2 Libraries
  - 04/05/18, 9:00 am-12:00 noon
- Tutorial: High Performance Distributed Deep Learning: A Beginner's Guide
  - 04/05/18, 1:00pm-4:00 pm

Funding Acknowledgments

Personnel Acknowledgments

Current Students (Graduate)

- A. Awan (Ph.D.)
- R. Biswas (M.S.)
- M. Bayatpour (Ph.D.)
- S. Chakraborthy (Ph.D.)
- C.-H. Chu (Ph.D.)
- S. Guganani (Ph.D.)
- J. Hashmi (Ph.D.)
- H. Javed (Ph.D.)
- P. Kousha (Ph.D.)
- D. Shankar (Ph.D.)
- H. Shi (Ph.D.)
- J. Zhang (Ph.D.)

Current Students (Undergraduate)

- N. Sarkauskas (B.S.)

Current Research Scientists

- X. Lu
- H. Subramoni

Current Research Specialist

- J. Smith
- M. Arnold

Current Post-doc

- A. Ruhela
- K. Manian

Past Students

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

Past Post-Docs

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

Past Research Scientist

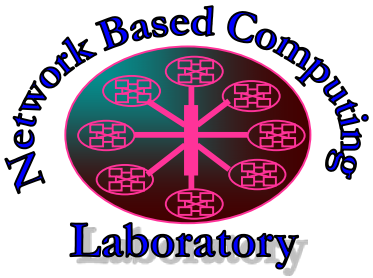
- K. Hamidouche
- S. Sur

Past Programmers

- D. Bureddy
- J. Perkins

Thank You!

panda@cse.ohio-state.edu



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

The High-Performance MPI/PGAS Project
http://mvapich.cse.ohio-state.edu/

The High-Performance Big Data Project
http://hibd.cse.ohio-state.edu/

The High-Performance Deep Learning Project
http://hidl.cse.ohio-state.edu/