MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

Designing Scalable, High-Performance Communication Runtimes for HPC and Deep Learning: The MVAPICH2 Approach

Talk at OpenFabrics Workshop (April '18)

by

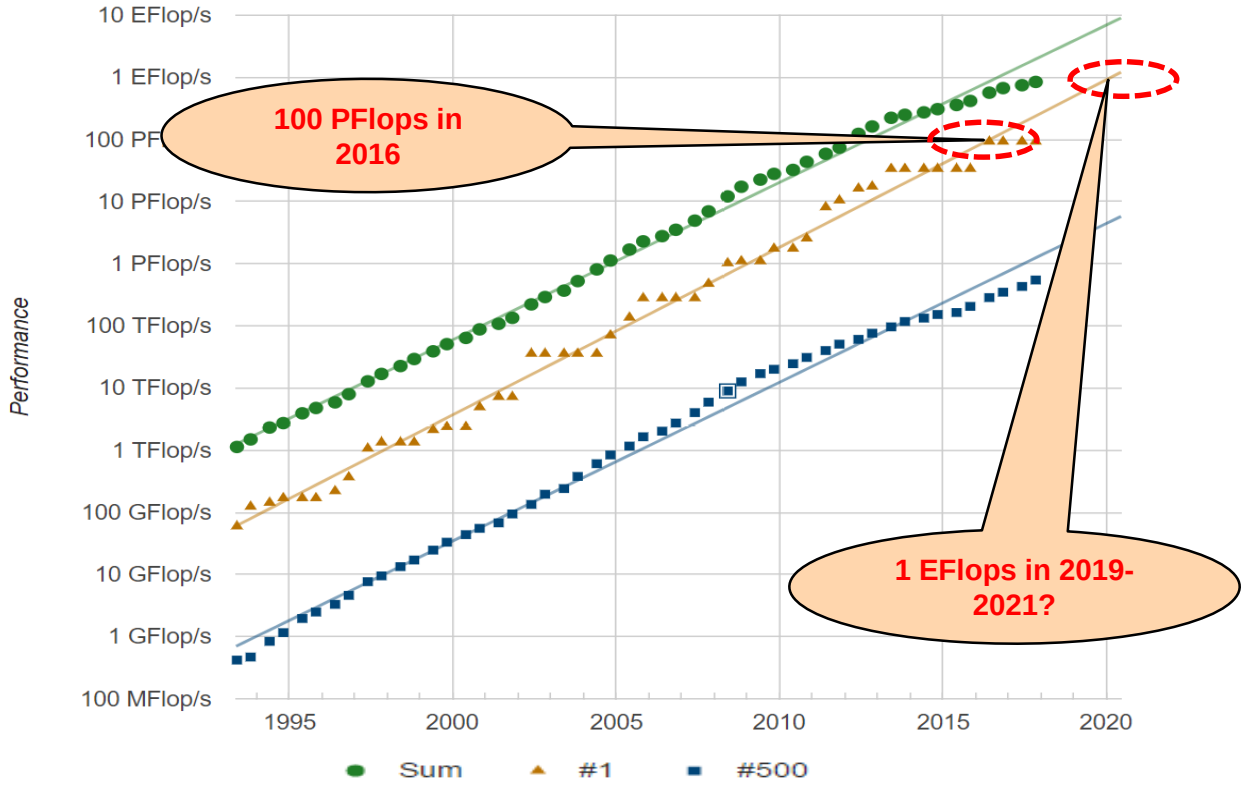
Hari Subramoni

The Ohio State University

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

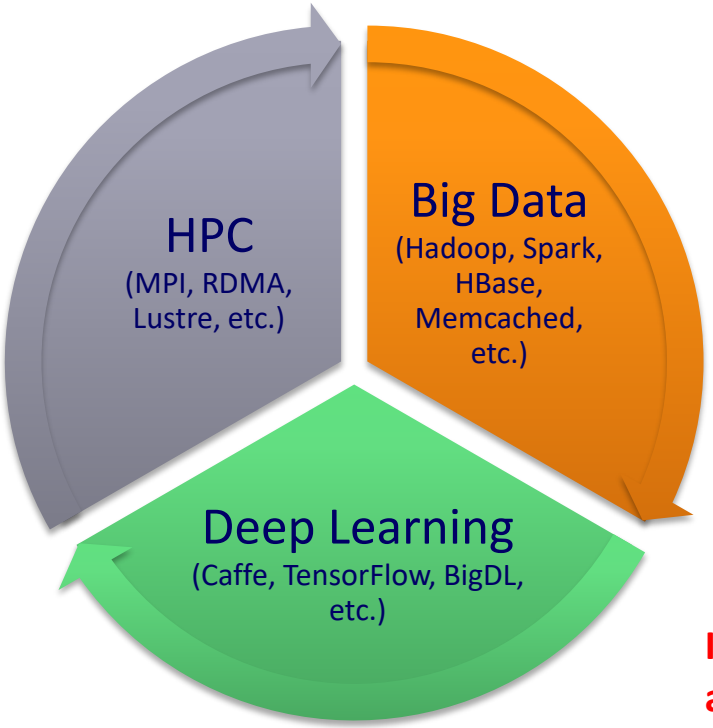
http://www.cse.ohio-state.edu/~subramon

High-End Computing (HEC): Towards Exascale



Expected to have an ExaFlop system in 2019-2021!

Increasing Usage of HPC, Big Data and Deep Learning



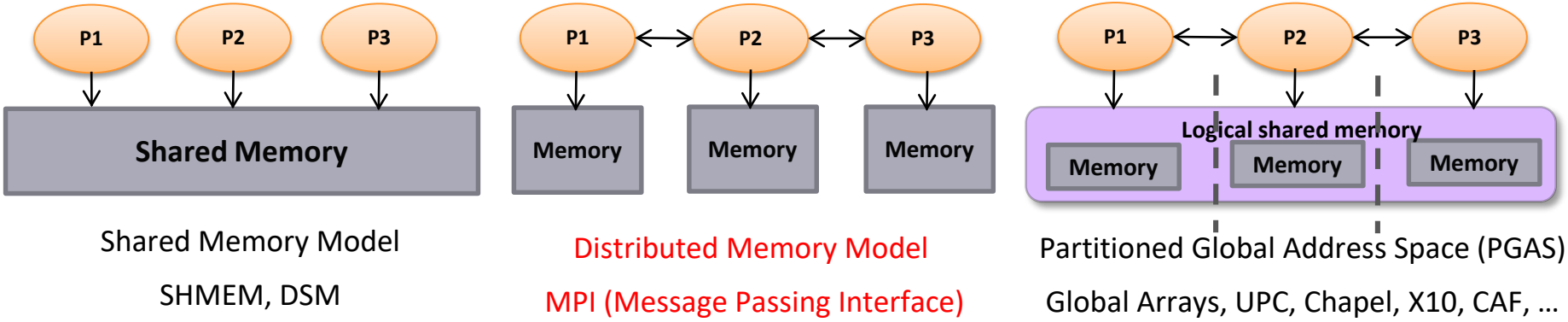
Convergence of HPC, Big Data, and Deep Learning!

Increasing Need to Run these applications on the Cloud!!

HPC and Deep Learning

- Traditional HPC
  - Message Passing Interface (MPI), including MPI + OpenMP
  - Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
  - Exploiting Accelerators
- Deep Learning
  - MPI-level Challenges
  - MVAPICH2-GDR Support
  - OSU-Caffe
  - Out-of-core Processing

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, …

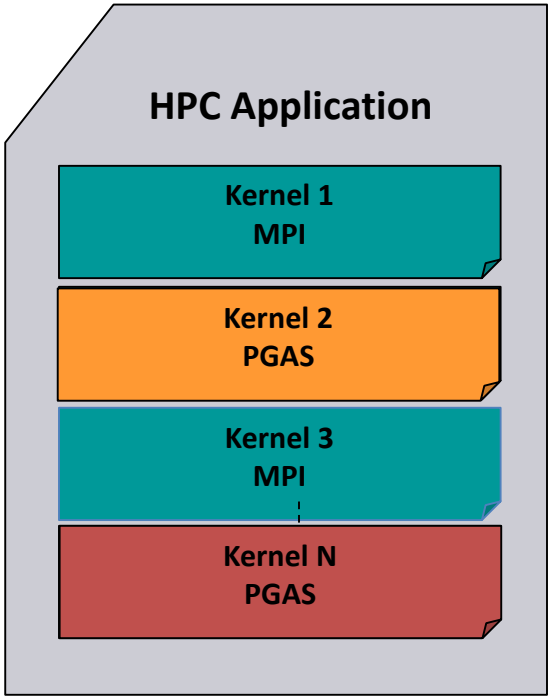
- Programming models provide abstract machine models
- Models can be mapped on different types of systems
  - e.g. Distributed Shared Memory (DSM), MPI within a node, etc.
- PGAS models and Hybrid MPI+PGAS models are gradually receiving importance

Partitioned Global Address Space (PGAS) Models

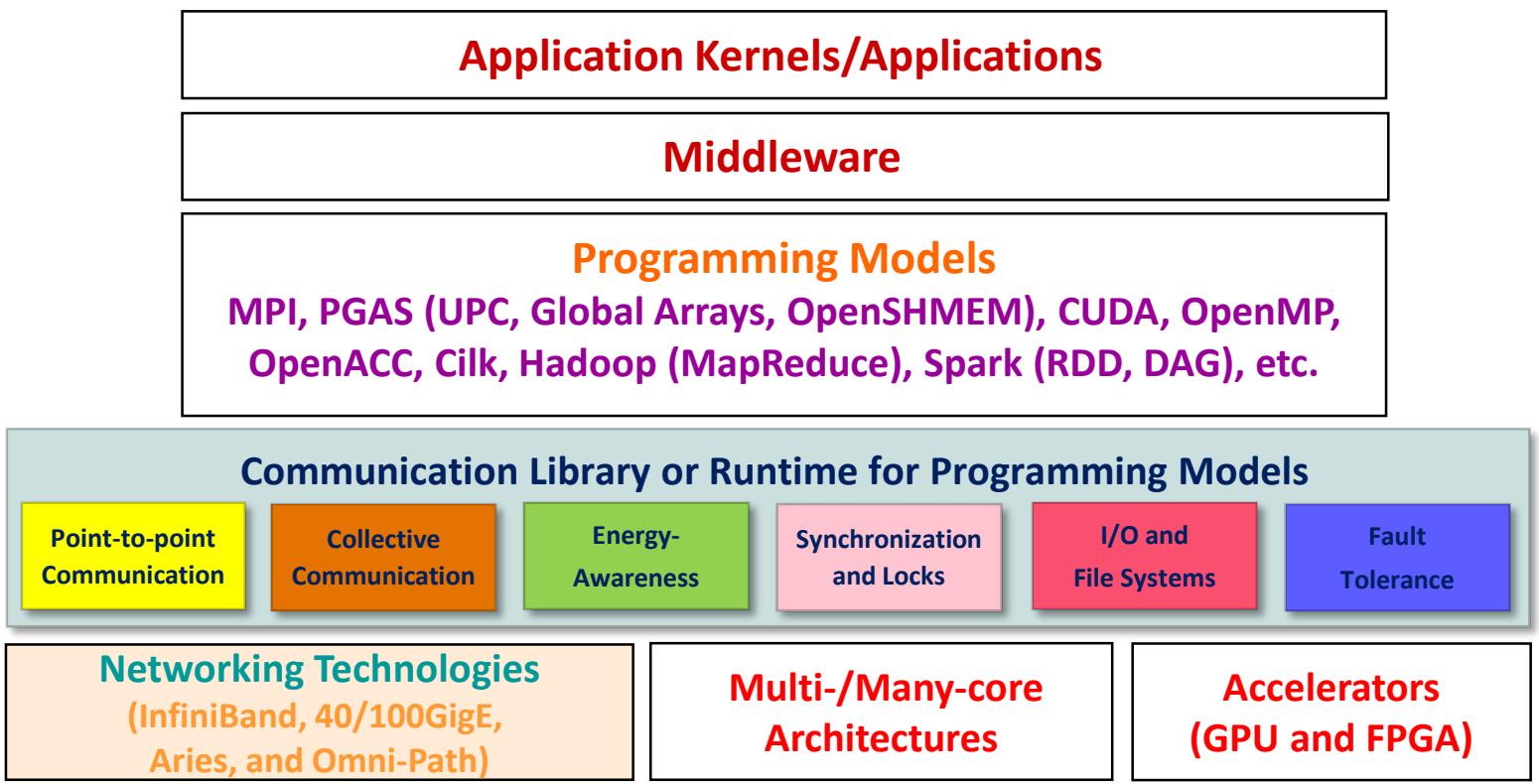
- Key features
  - Simple shared memory abstractions
  - Light weight one-sided communication
  - Easier to express irregular communication
- Different approaches to PGAS
  - Languages
    - Unified Parallel C (UPC)
    - Co-Array Fortran (CAF)
    - X10
    - Chapel
  - Libraries
    - OpenSHMEM
  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



Co-Design Opportunities and Challenges across Various Layers

Performance

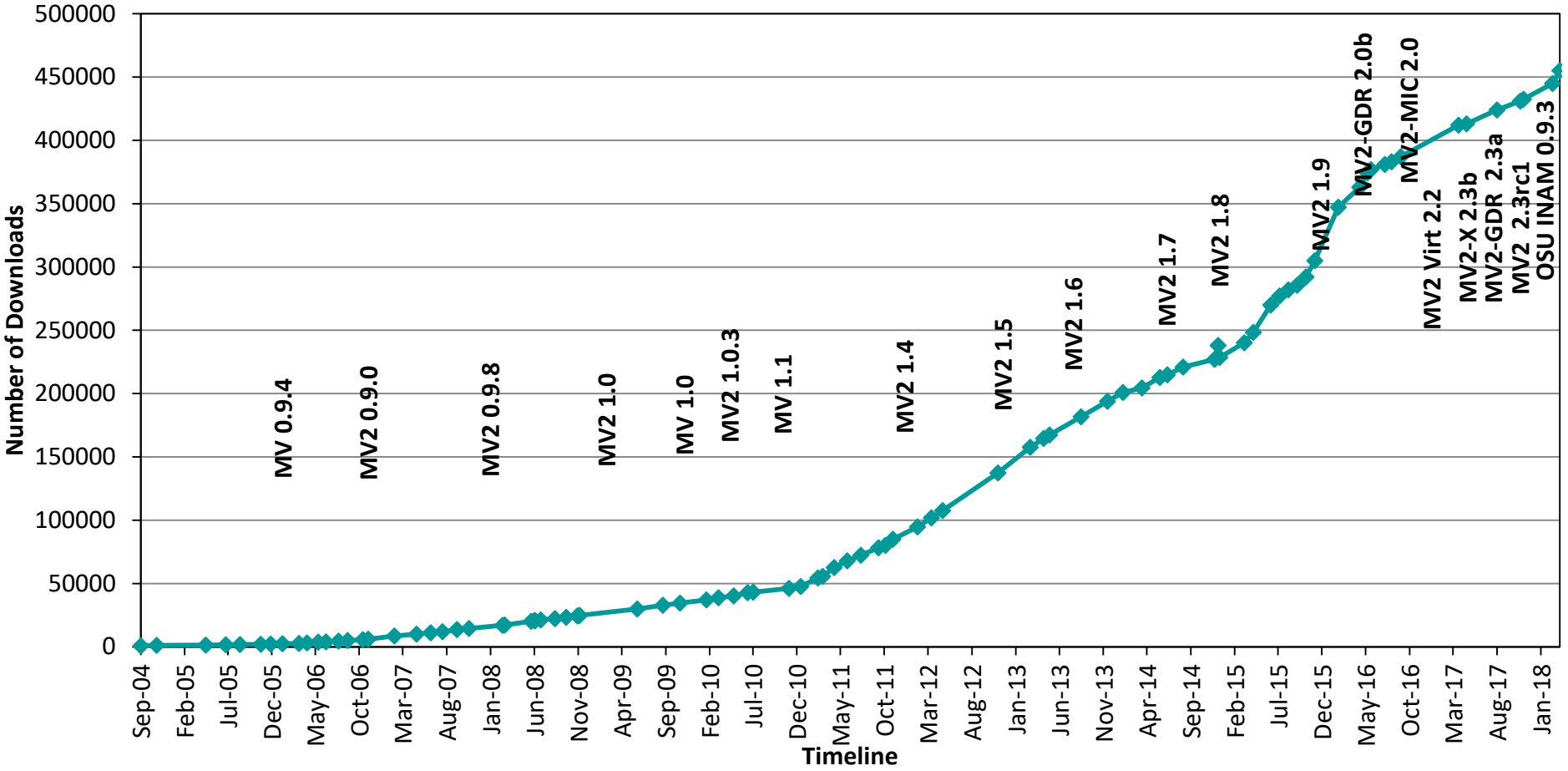
Scalability

Resilience

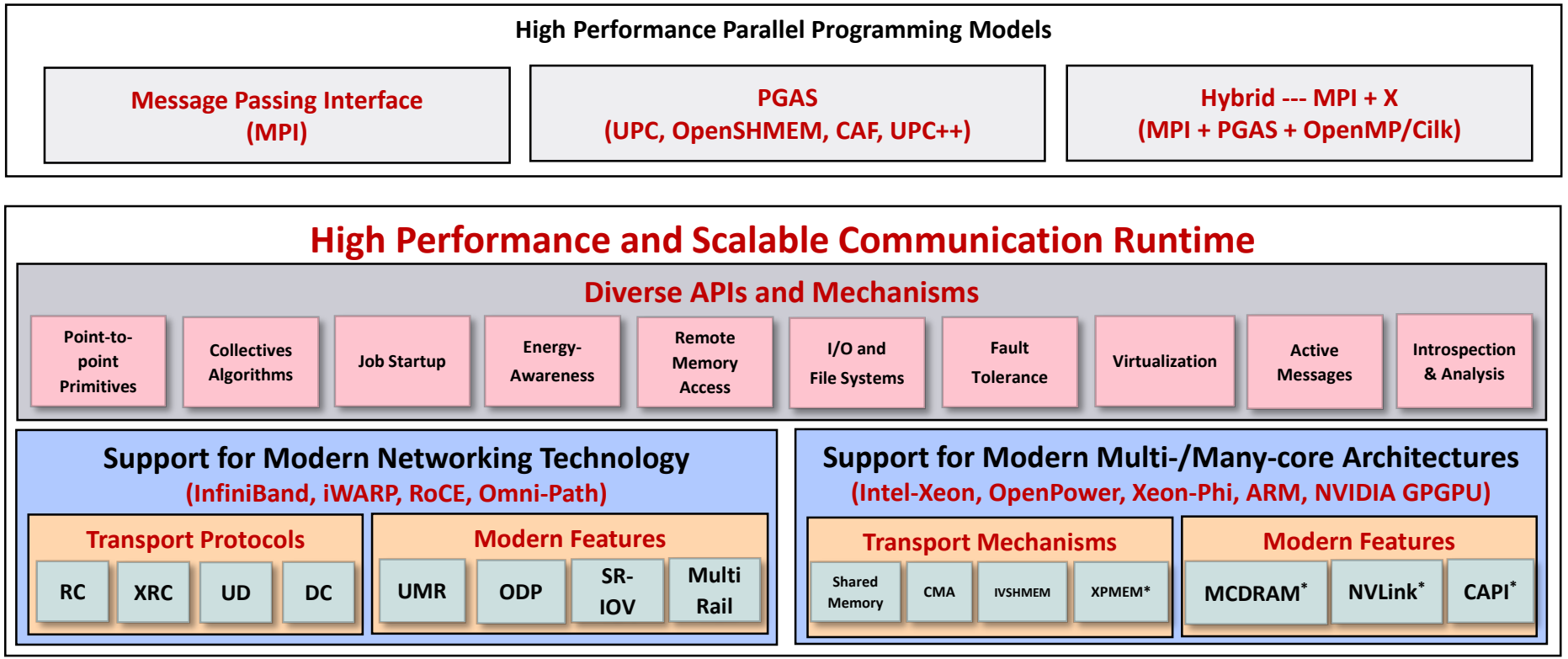
Overview of the MVAPICH2 Project

- 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 462,000 (> 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

MVAPICH2 Software Family

High-Performance Parallel Programming Libraries	
MVAPICH2	Support for InfiniBand, Omni-Path, Ethernet/iWARP, and RoCE
MVAPICH2-X	Advanced MPI features, OSU INAM, PGAS (OpenSHMEM, UPC, UPC++, and CAF), and MPI+PGAS programming models with unified communication runtime
MVAPICH2-GDR	Optimized MPI for clusters with NVIDIA GPUs
MVAPICH2-Virt	High-performance and scalable MPI for hypervisor and container based HPC cloud
MVAPICH2-EA	Energy aware and High-performance MPI
MVAPICH2-MIC	Optimized MPI for clusters with Intel KNC
Microbenchmarks	
OMB	Microbenchmarks suite to evaluate MPI and PGAS (OpenSHMEM, UPC, and UPC++) libraries for CPUs and GPUs
Tools	
OSU INAM	Network monitoring, profiling, and analysis for clusters with MPI and scheduler integration
OEMT	Utility to measure the energy consumption of MPI applications

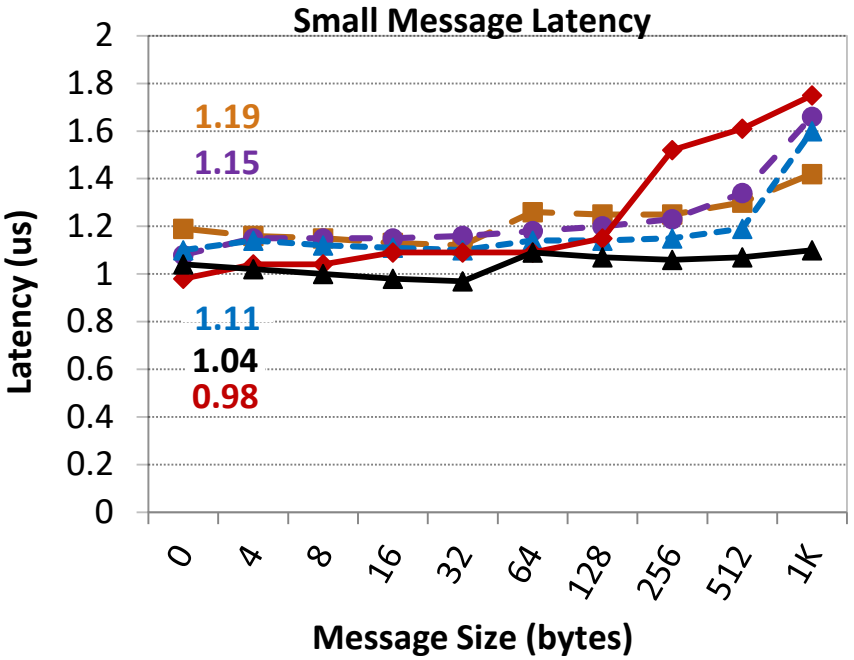
MVAPICH2 2.3rc1

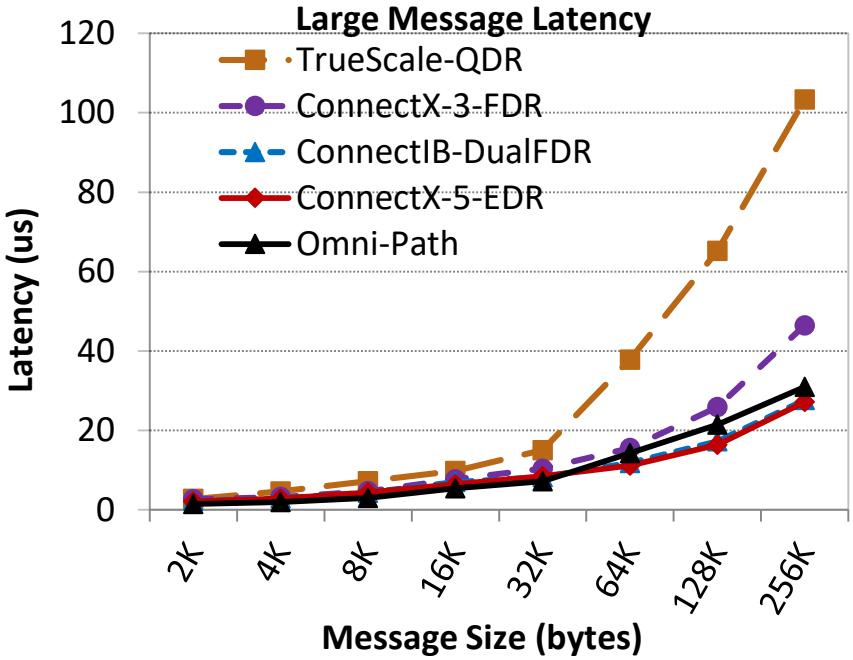
- **Released on 02/19/2018**
- Major Features and Enhancements
  - **Enhanced performance for Allreduce, Reduce_scatter_block, Allgather, Allgatherv through new algorithms**
  - **Enhance support for MPI_T PVARs and CVARs**
  - **Improved job startup time for OFA-IB-CH3, PSM-CH3, and PSM2-CH3**
  - **Support to automatically detect IP address of IB/RoCE interfaces when RDMA_CM is enabled without relying on mv2.conf file**
  - **Enhance HCA detection to handle cases where node has both IB and RoCE HCAs**
  - **Automatically detect and use maximum supported MTU by the HCA**
  - **Added logic to detect heterogeneous CPU/HFI configurations in PSM-CH3 and PSM2-CH3 channels**
  - **Enhanced intra-node and inter-node tuning for PSM-CH3 and PSM2-CH3 channels**
  - **Enhanced HFI selection logic for systems with multiple Omni-Path HFIs**
  - **Enhanced tuning and architecture detection for OpenPOWER, Intel Skylake and Cavium ARM (ThunderX) systems**
  - **Added 'SPREAD', 'BUNCH', and 'SCATTER' binding options for hybrid CPU binding policy**
  - **Rename MV2_THREADS_BINDING_POLICY to MV2_HYBRID_BINDING_POLICY**
  - **Added support for MV2_SHOW_CPU_BINDING to display number of OMP threads**
  - **Update to hwloc version 1.11.9**

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
  - 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

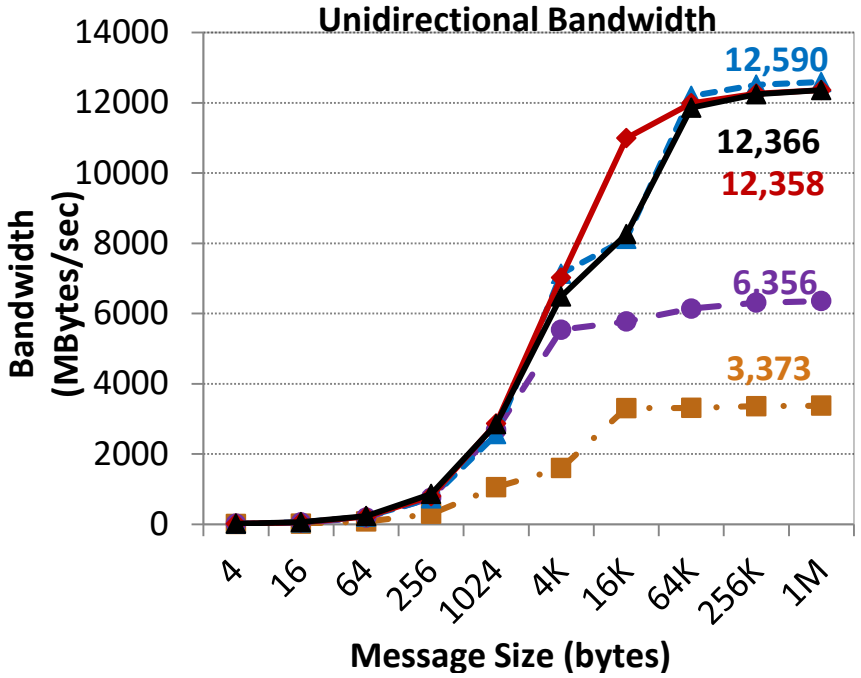
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

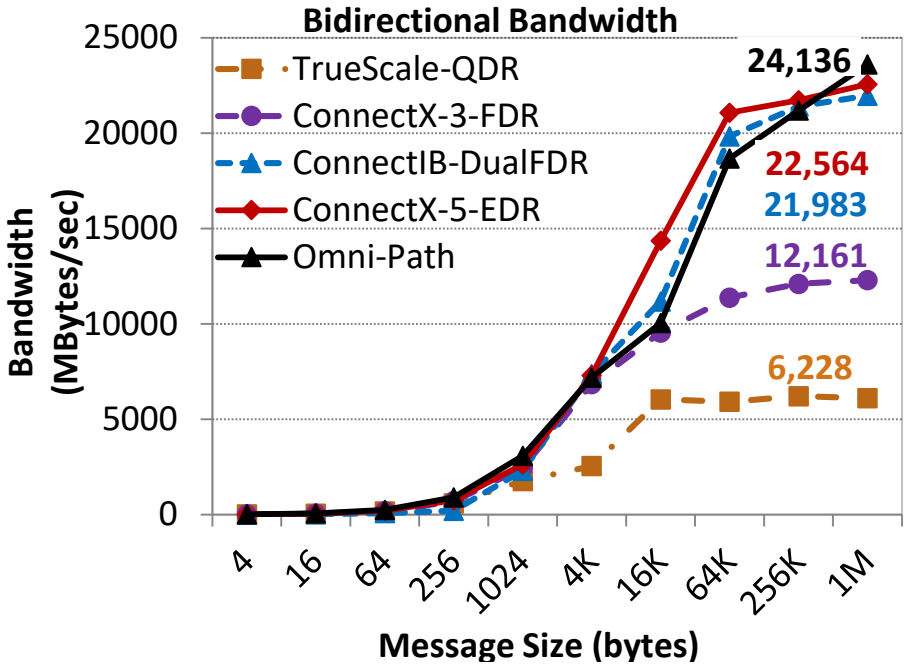
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

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

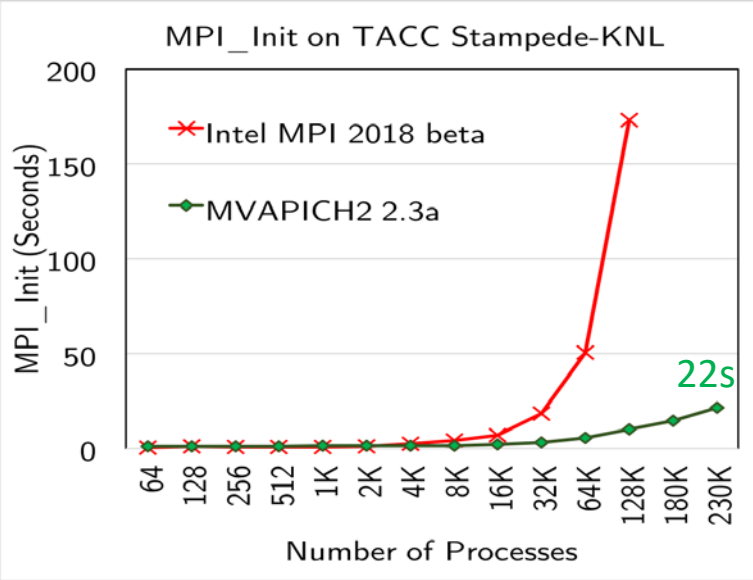
ConnectX-3-FDR - 2.8 GHz Deca-core (IvyBridge) Intel PCI Gen3 with IB switch

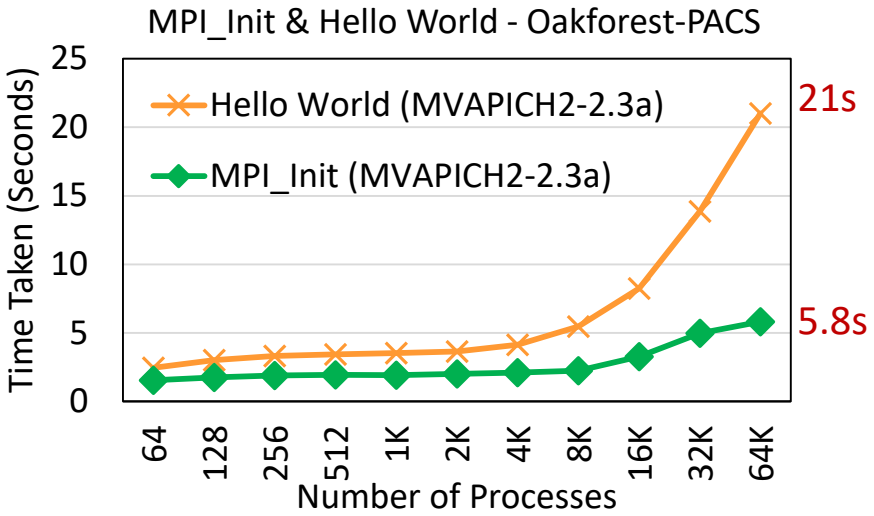
ConnectIB-Dual FDR - 3.1 GHz Deca-core (Haswell) Intel PCI Gen3 with IB switch

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

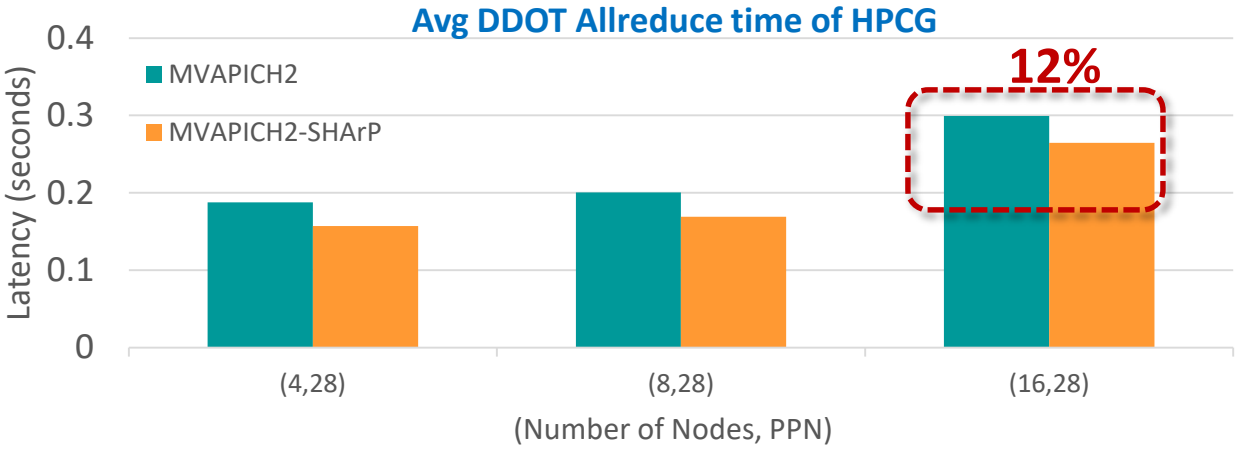


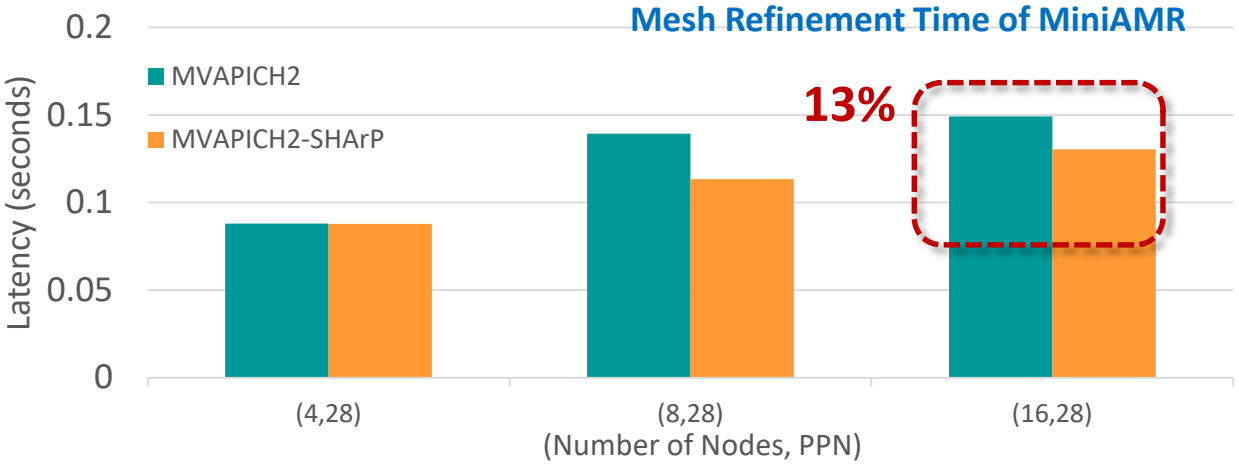


- MPI_Init takes 22 seconds on 229,376 processes on 3,584 KNL nodes (Stampede2 – Full scale)
- 8.8 times faster than Intel MPI at 128K processes (Courtesy: TACC)
- At 64K processes, MPI_Init and Hello World takes 5.8s and 21s respectively (Oakforest-PACS)
- All numbers reported with 64 processes per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

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

Advanced Allreduce Collective Designs Using SHArP

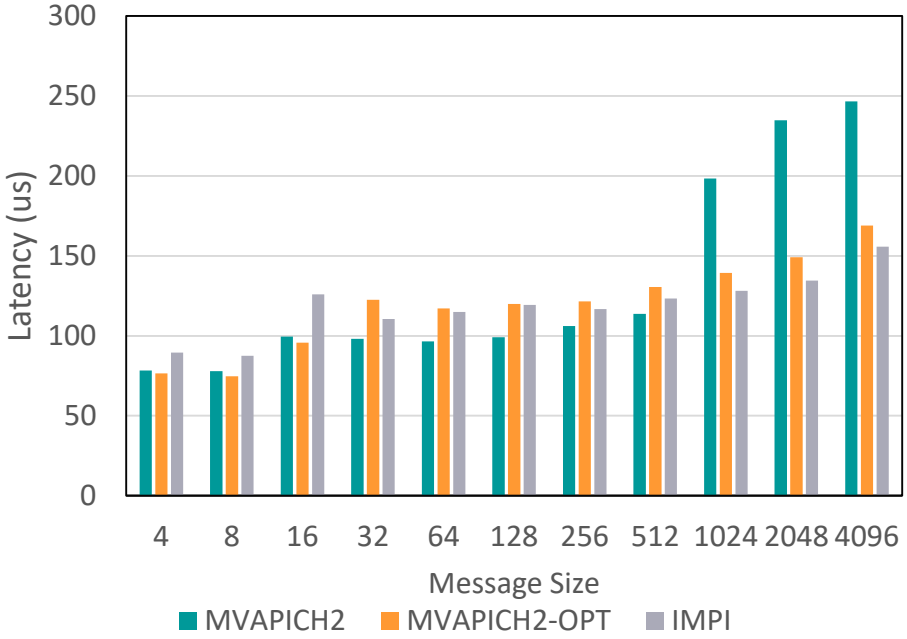


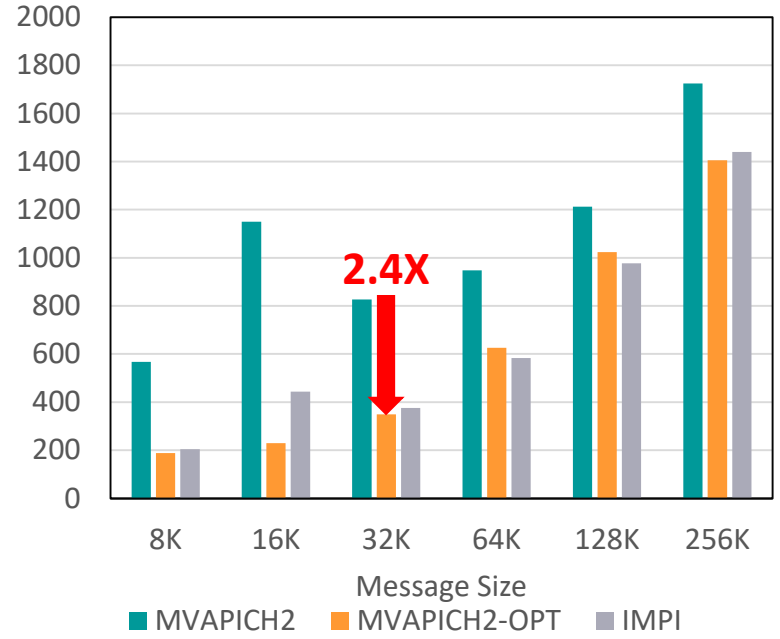


SHArP Support is available since MVAPICH2 2.3a

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)





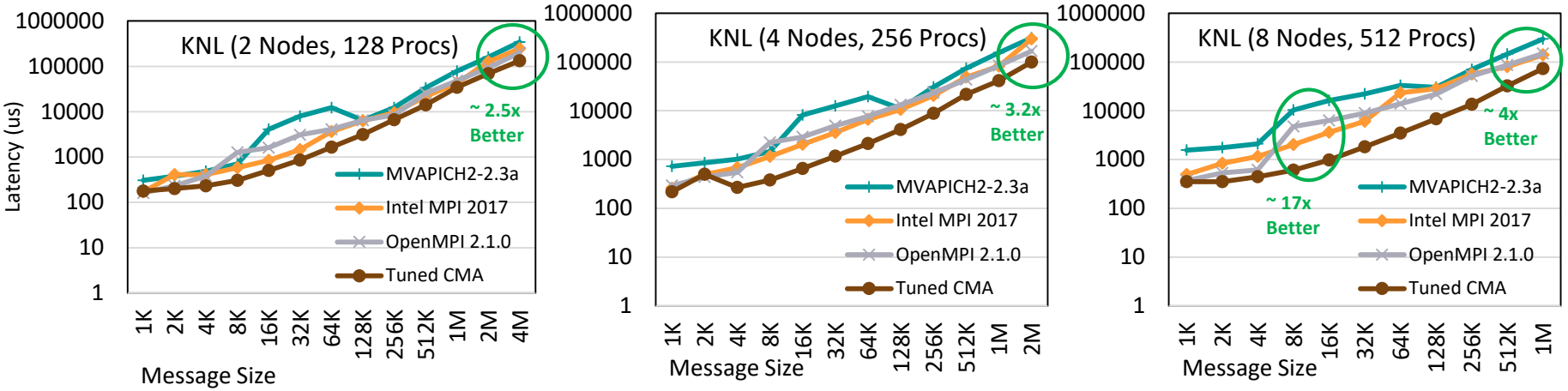
OSU Micro Benchmark 64 PPN

- 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



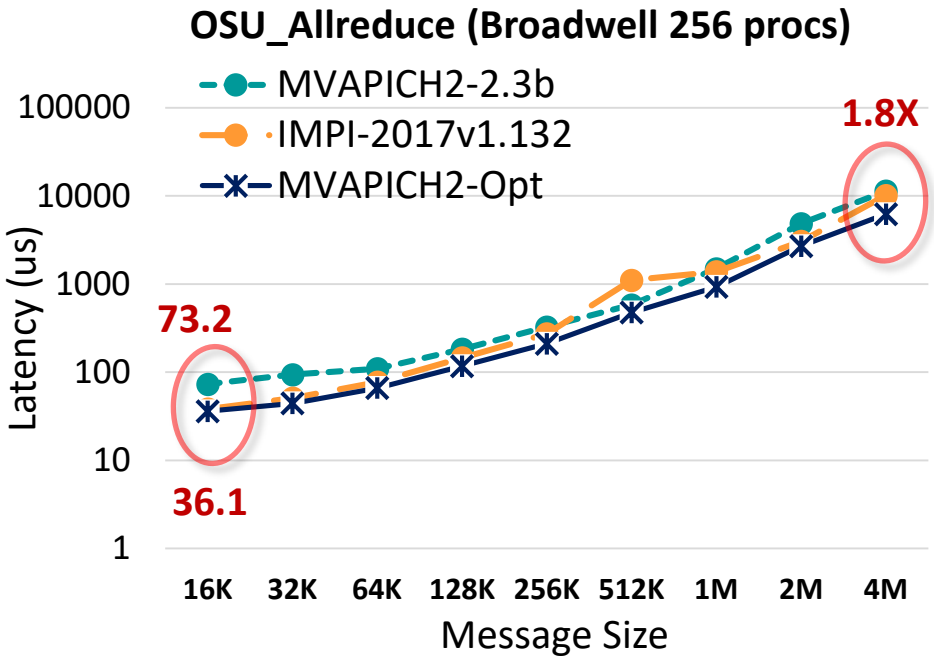
Performance of MPI_Gather on KNL nodes (64PPN)

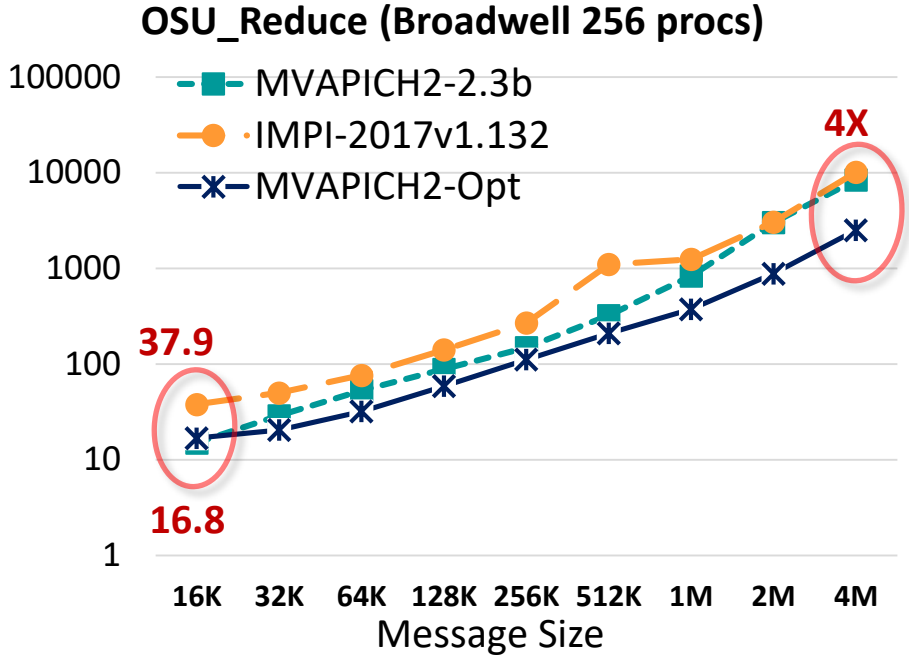
- Significant improvement over existing implementation for Scatter/Gather with 1MB messages (up to 4x on KNL, 2x on Broadwell, 14x on OpenPower)
- New two-level algorithms for better scalability
- Improved performance for other collectives (Bcast, Allgather, and Alltoall)

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

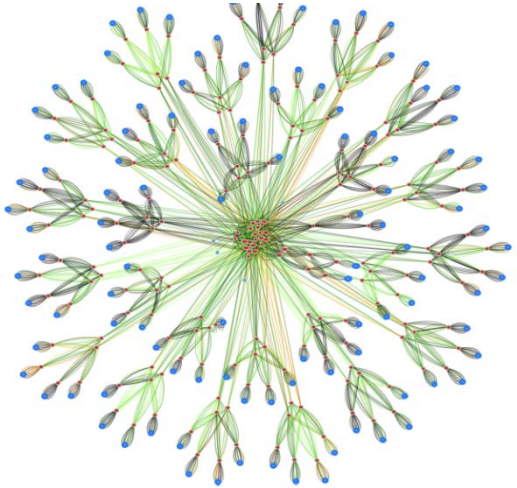
J. Hashmi, S. Chakraborty, M. Bayatpour, H. Subramoni, and D. Panda, Designing Efficient Shared Address Space Reduction Collectives for Multi-/Many-cores, International Parallel & Distributed Processing Symposium (IPDPS '18), May 2018.

Will be available in future

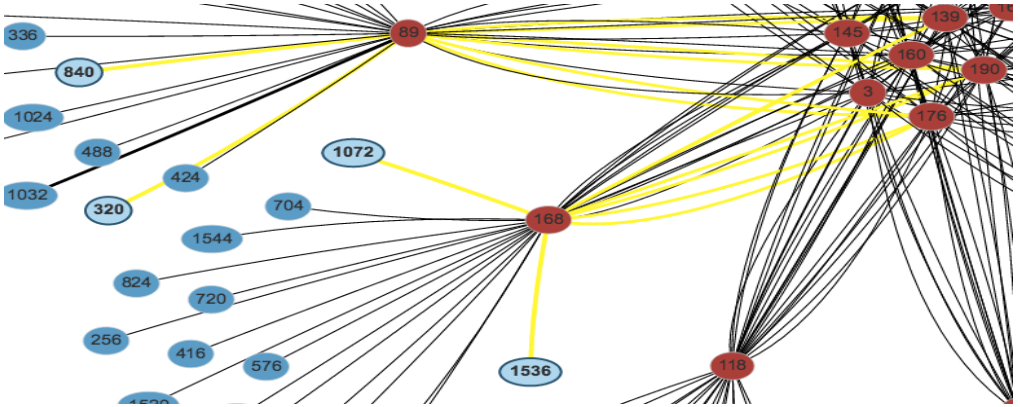
Overview of OSU INAM

- 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

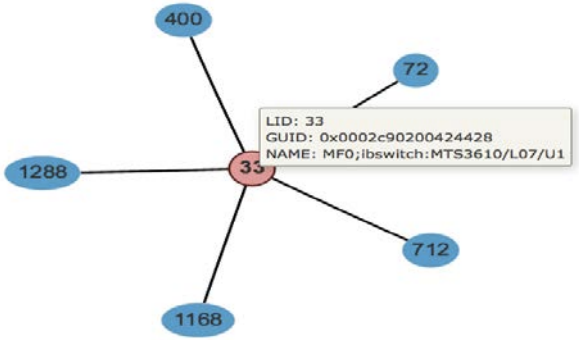


Finding Routes Between Nodes

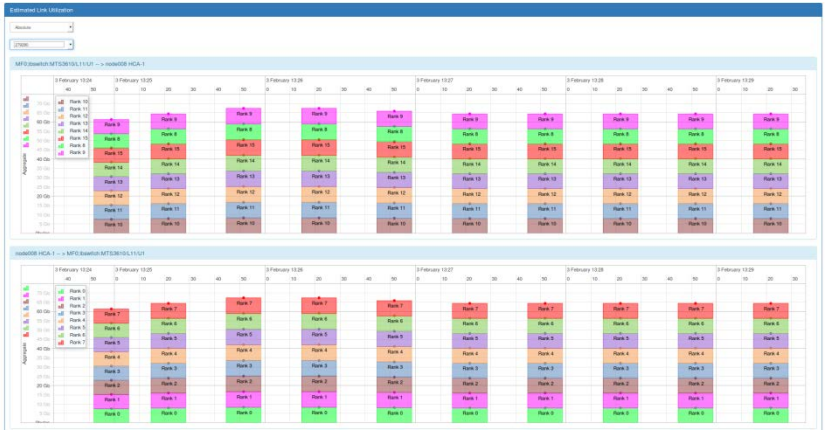
(1,879 nodes, 212 switches, 4,377 network links)

- 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

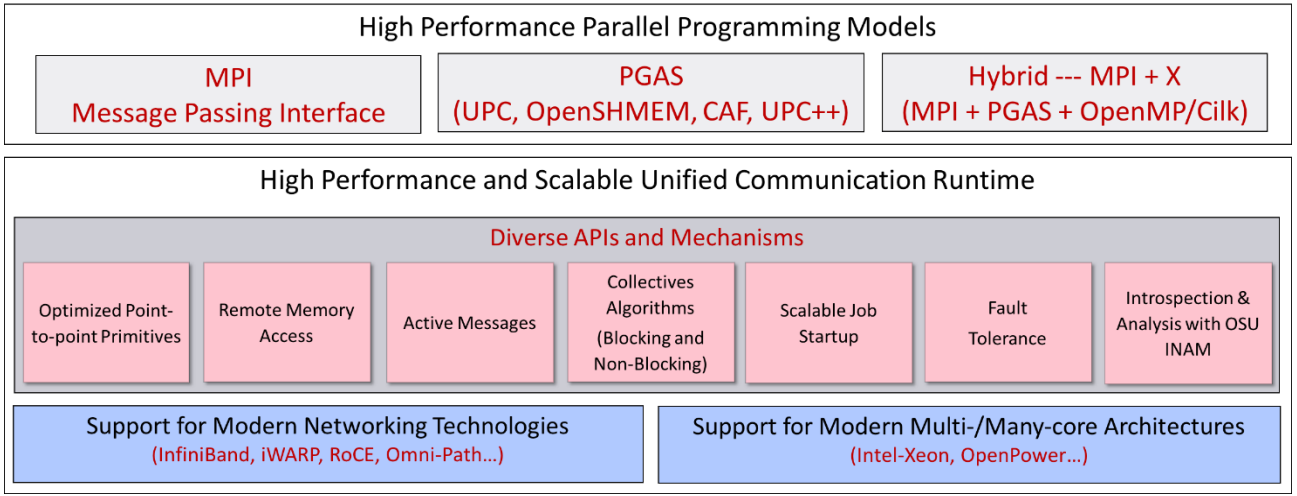
- Job level view
  - Show different network metrics (load, error, etc.) for any live job
  - Play back historical data for completed jobs to identify bottlenecks
- Node level view - details per process or per node
  - CPU utilization for each rank/node
  - Bytes sent/received for MPI operations (pt-to-pt,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

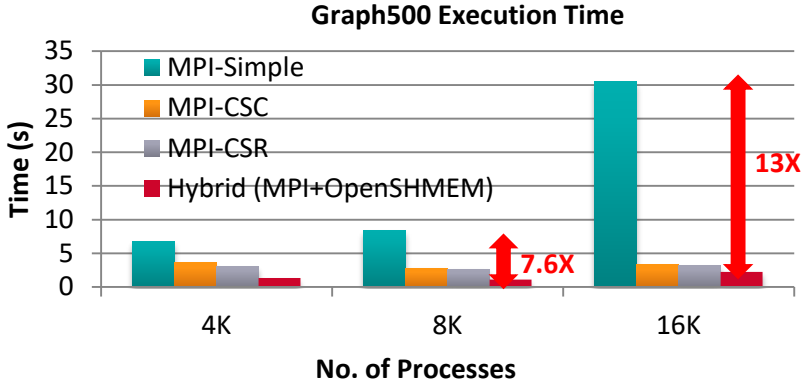
- Scalability for million to billion processors
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, UPC++, …)
- Integrated Support for GPGPUs
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- Application Scalability and Best Practices

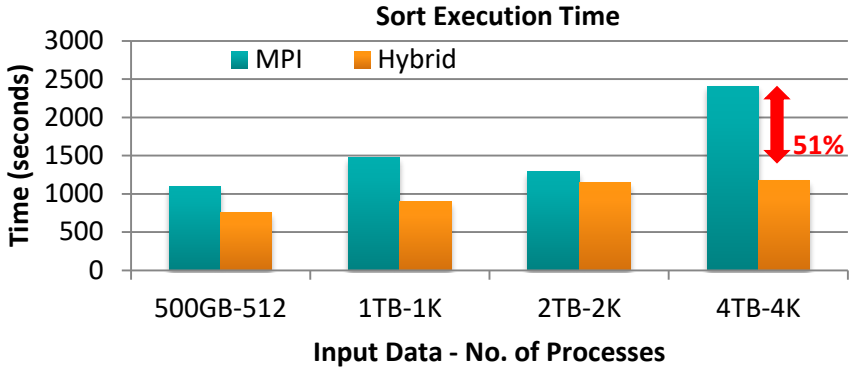
MVAPICH2-X for Hybrid MPI + PGAS Applications



- 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

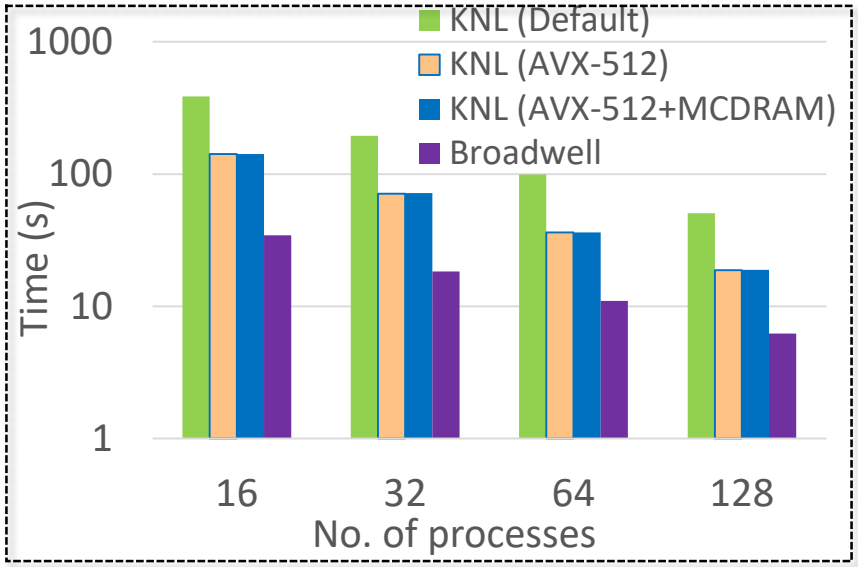
J. Jose, K. Kandalla, S. Potluri, J. Zhang and D. K. Panda, Optimizing Collective Communication in OpenSHMEM, Int'l Conference on Partitioned Global Address Space Programming Models (PGAS '13), October 2013.

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

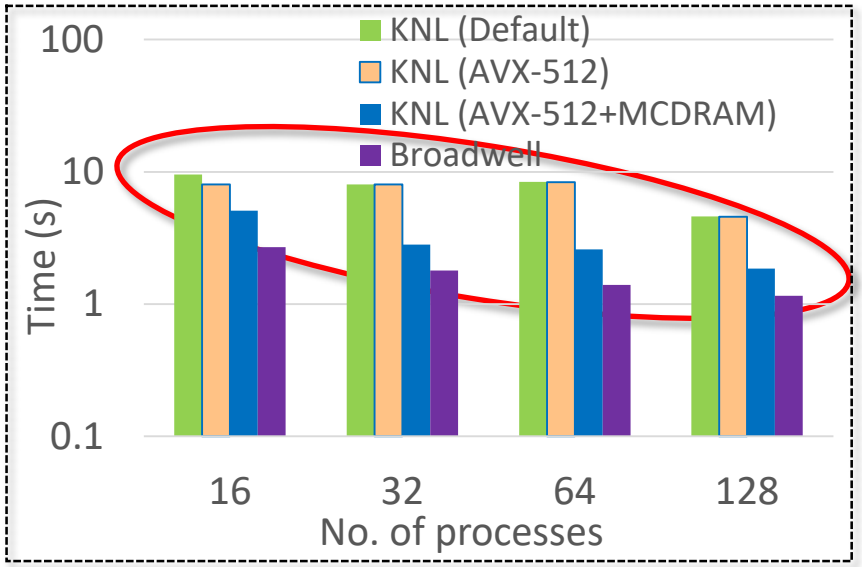
J. Jose, K. Kandalla, M. Luo and D. K. Panda, Supporting Hybrid MPI and OpenSHMEM over InfiniBand: Design and Performance Evaluation, Int'l Conference on Parallel Processing (ICPP '12), September 2012

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
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- Application Scalability and Best Practices

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

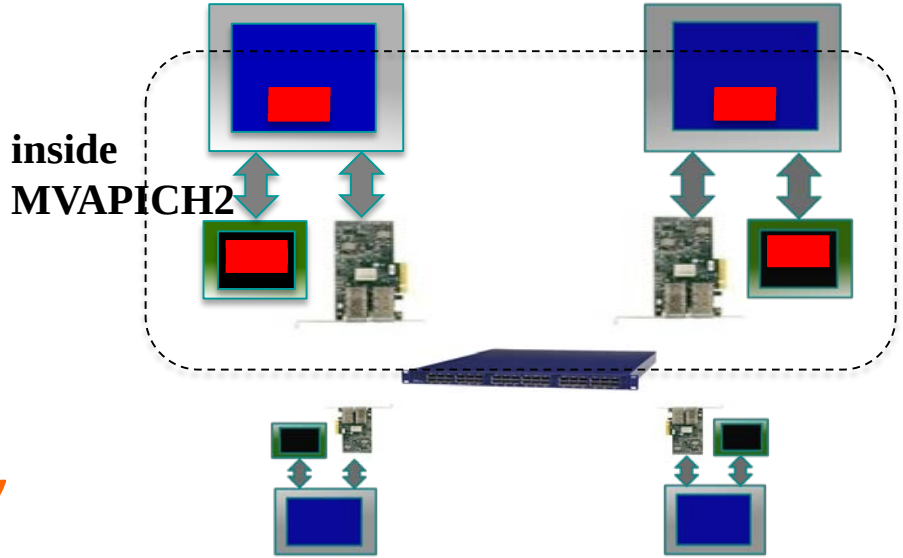
- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

At Sender:

```
MPI_Send(s_devbuf, size, …);
```

At Receiver:

```
MPI_Recv(r_devbuf, size, …);
```

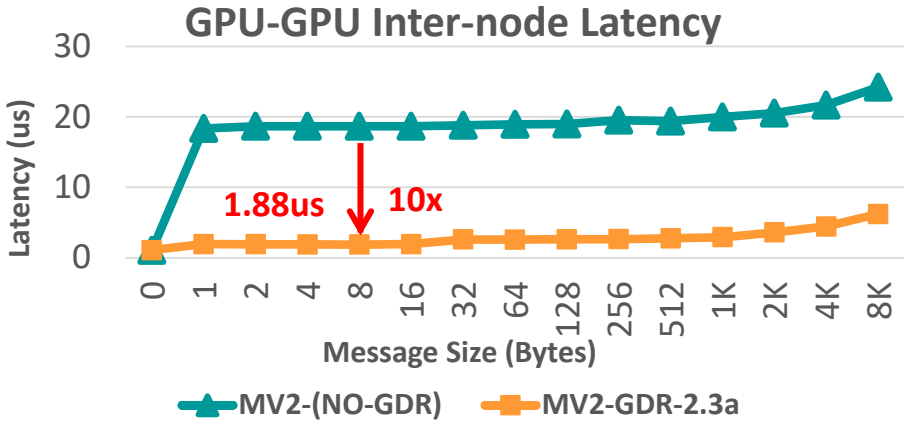


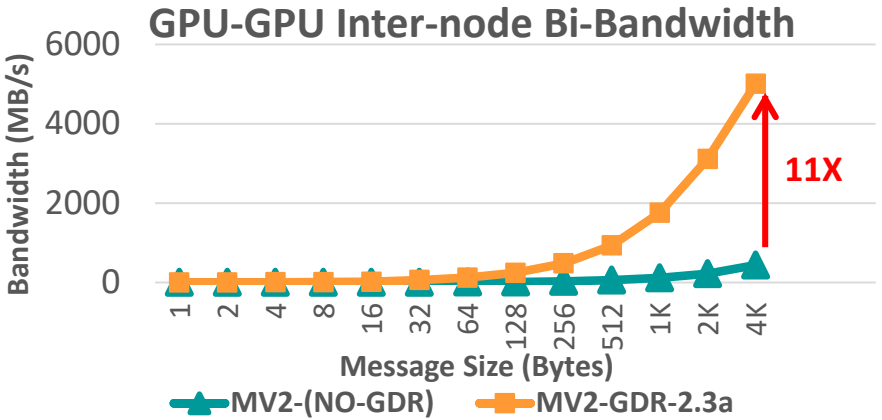
High Performance and High Productivity

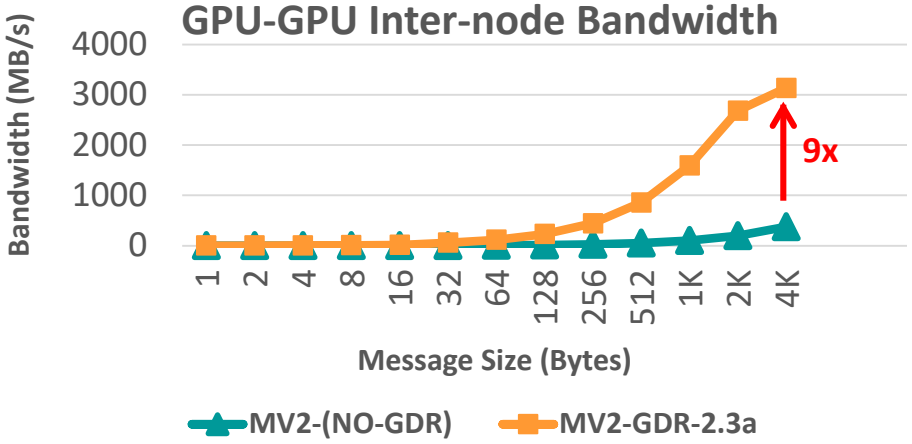
CUDA-Aware MPI: MVAPICH2-GDR 1.8-2.3 Releases

- 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

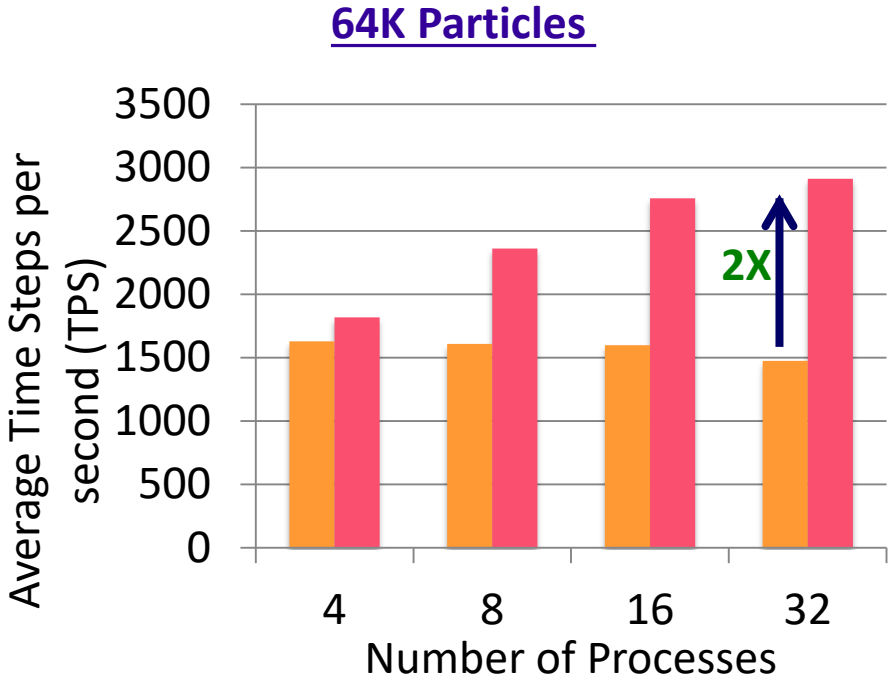


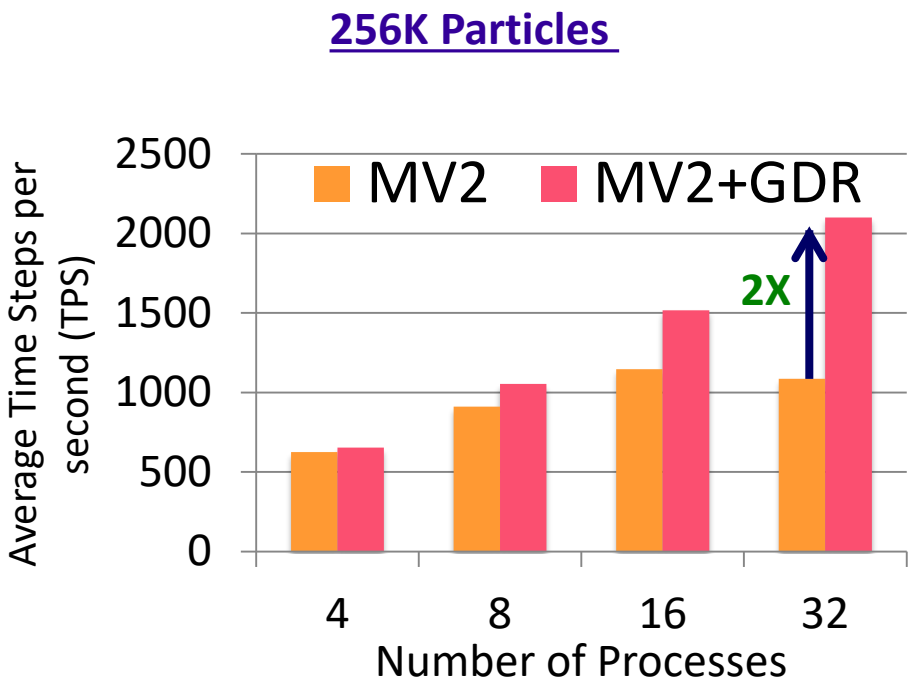




MVAPICH2-GDR-2.3a
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

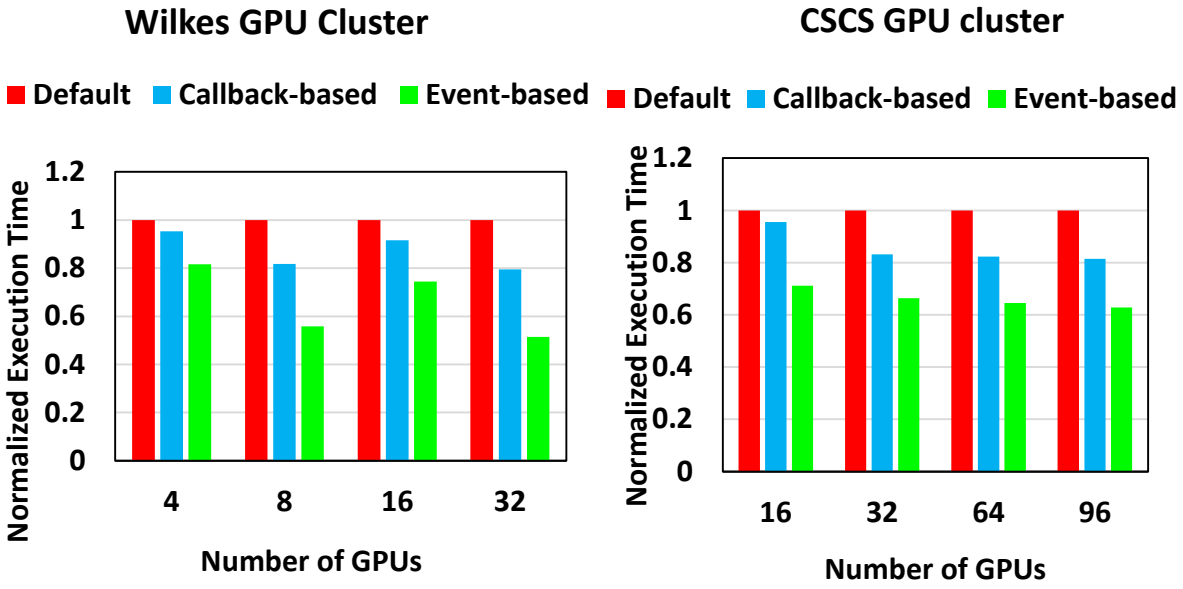
Application-Level Evaluation (HOOMD-blue)

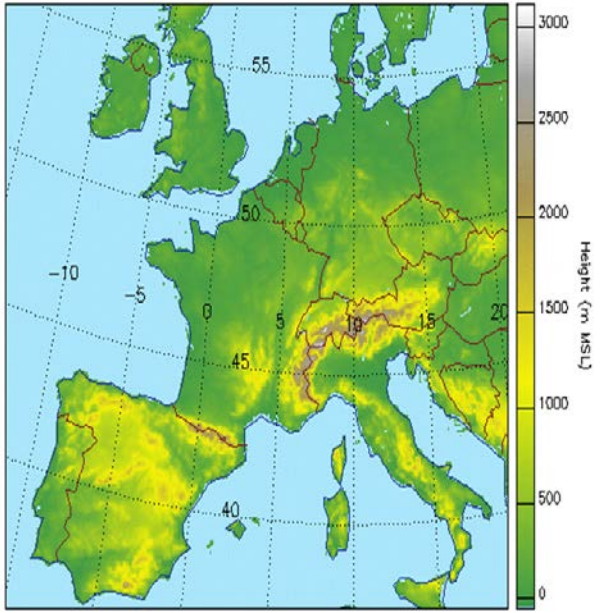




- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- HoomdBlue Version 1.0.5
  - GDRCOPY enabled: MV2_USE_CUDA=1 MV2_IBA_HCA=mlx5_0 MV2_IBA_EAGER_THRESHOLD=32768 MV2_VBUF_TOTAL_SIZE=32768 MV2_USE_GPUDIRECT_LOOPBACK_LIMIT=32768 MV2_USE_GPUDIRECT_GDRCOPY=1 MV2_USE_GPUDIRECT_GDRCOPY_LIMIT=16384

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/

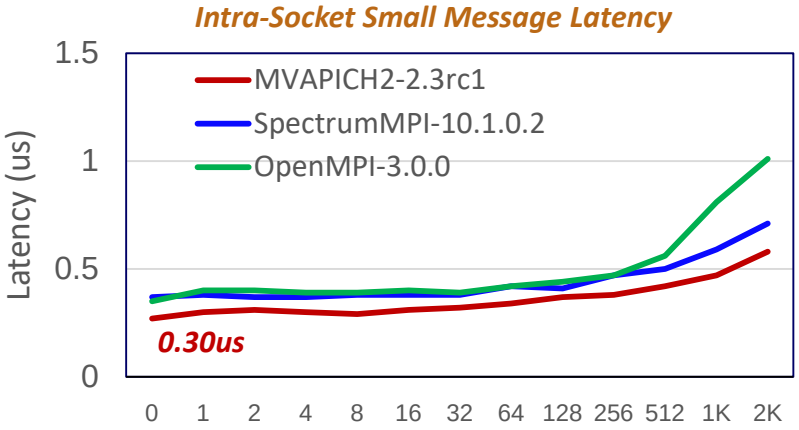
On-going collaboration with CSCS and MeteoSwiss (Switzerland) in co-designing MV2-GDR and Cosmo Application

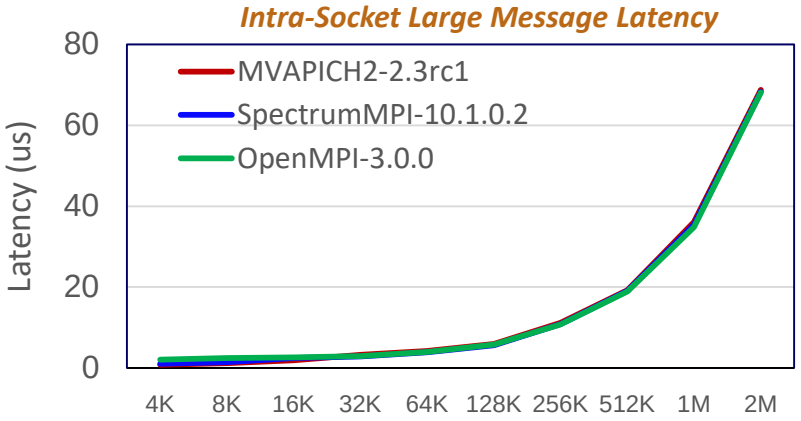
C. Chu, K. Hamidouche, A. Venkatesh, D. Banerjee , H. Subramoni, and D. K. Panda, Exploiting Maximal Overlap for Non-Contiguous Data Movement Processing on Modern GPU-enabled Systems, IPDPS'16

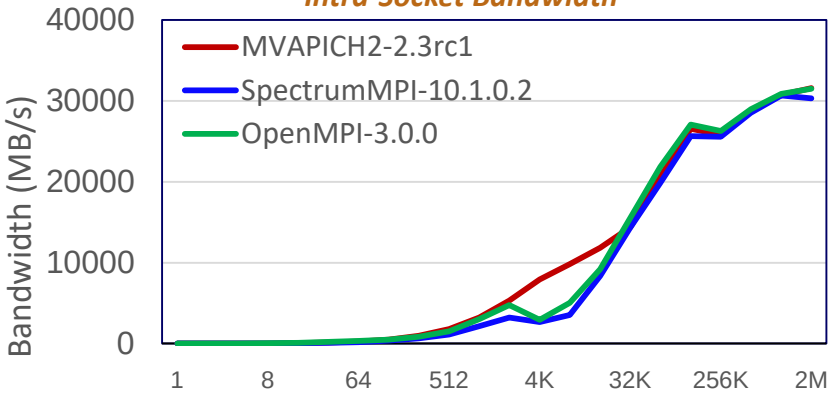
Overview of A Few Challenges being Addressed by the MVAPICH2 Project for Exascale

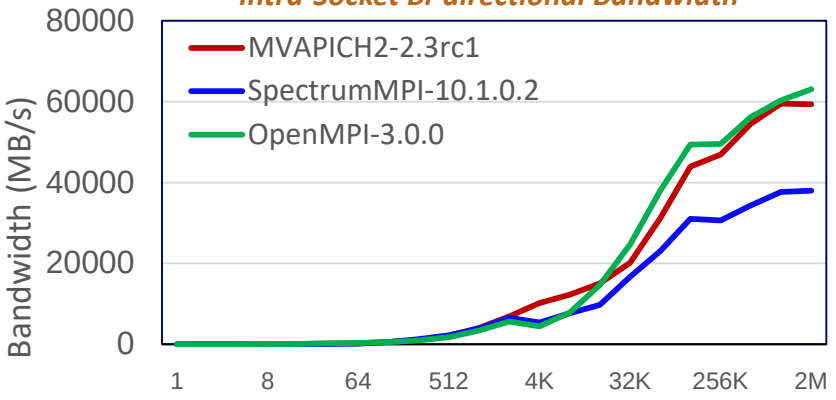
- Scalability for million to billion processors
- Unified Runtime for Hybrid MPI+PGAS programming (MPI + OpenSHMEM, MPI + UPC, CAF, UPC++, …)
- Integrated Support for GPGPUs
- Optimized MVAPICH2 for OpenPower (with/ NVLink) and ARM
- Application Scalability and Best Practices

Intra-node Point-to-Point Performance on OpenPower



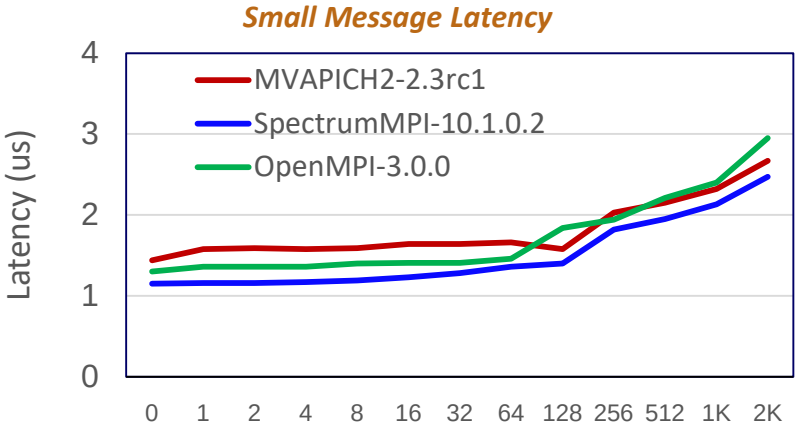


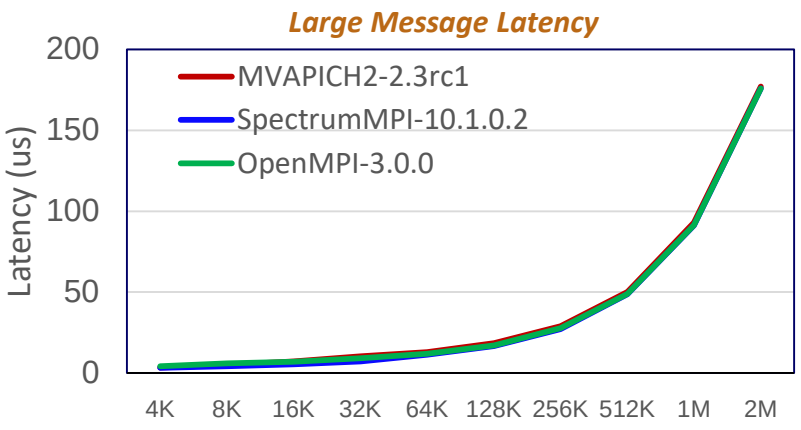


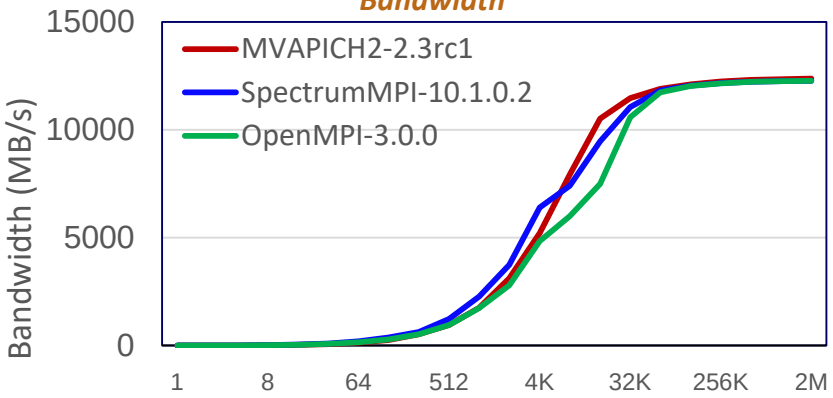


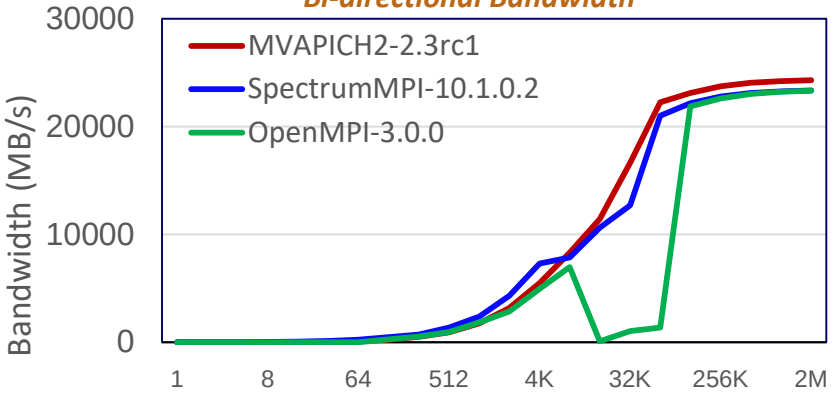
Platform: Two nodes of OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA

Inter-node Point-to-Point Performance on OpenPower



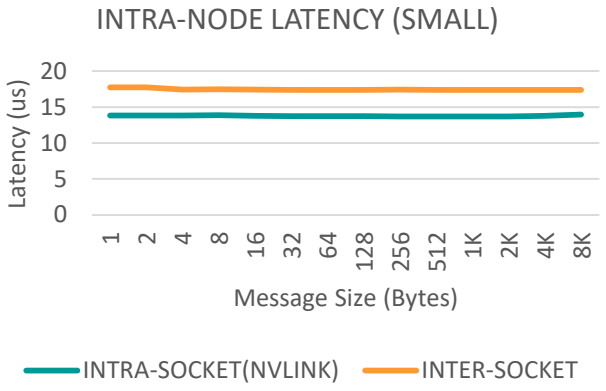


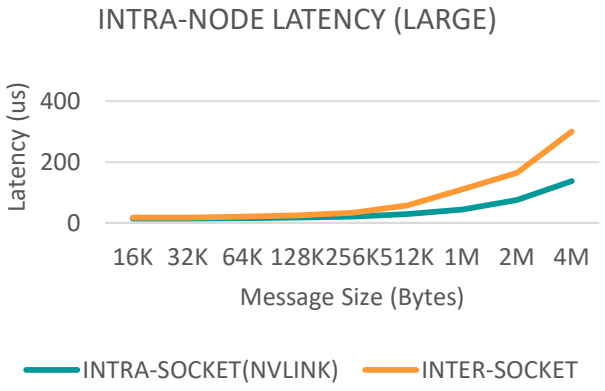


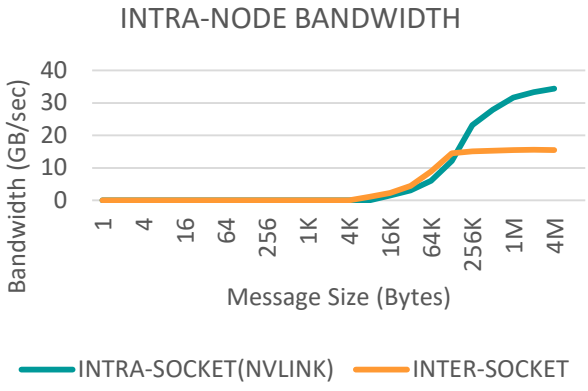


Platform: Two nodes of OpenPOWER (Power8-ppc64le) CPU using Mellanox EDR (MT4115) HCA

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

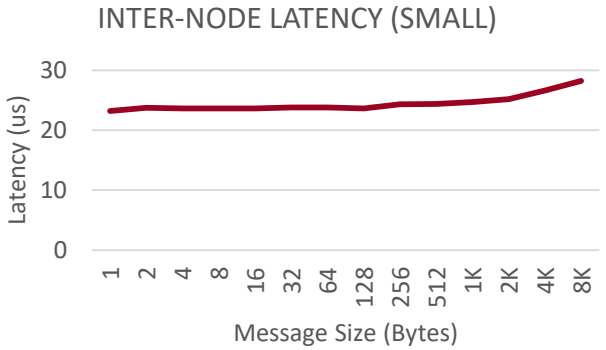


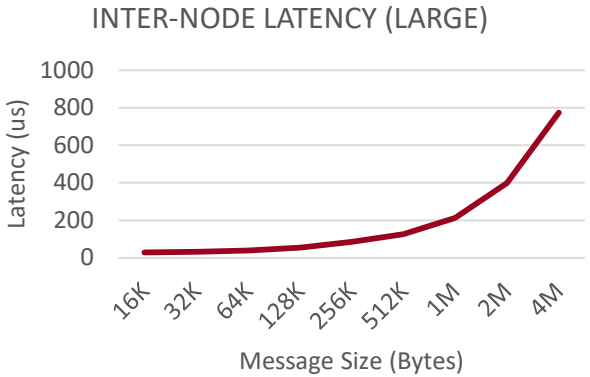


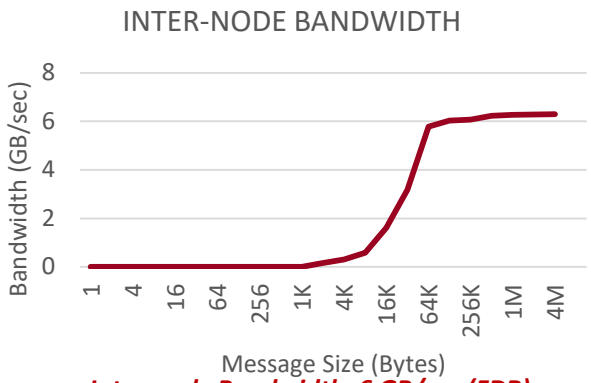


Intra-node Latency: 13.8 us (without GPUDirectRDMA)

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







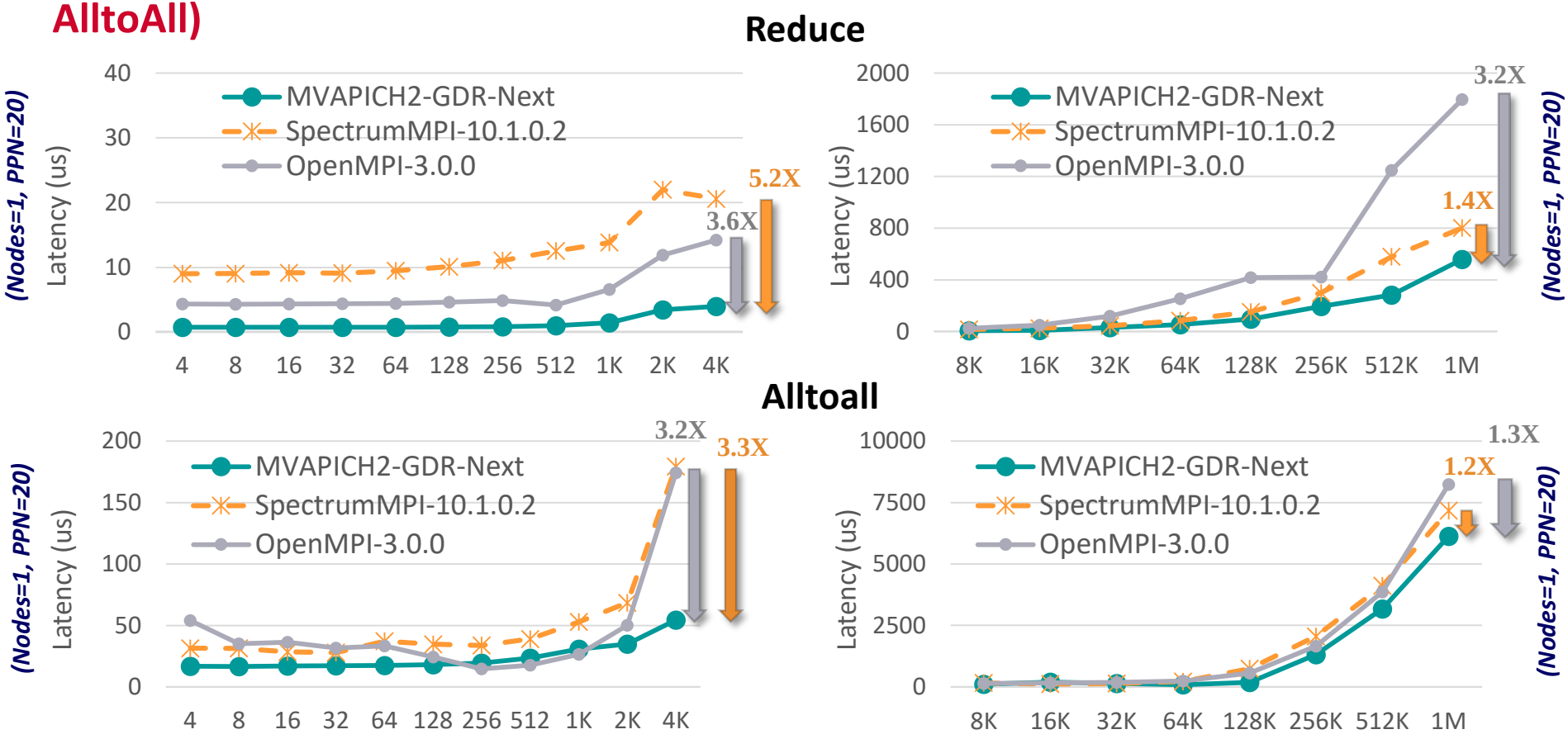
Inter-node Latency: 23 us (without GPUDirectRDMA)

Available in MVAPICH2-GDR 2.3a

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

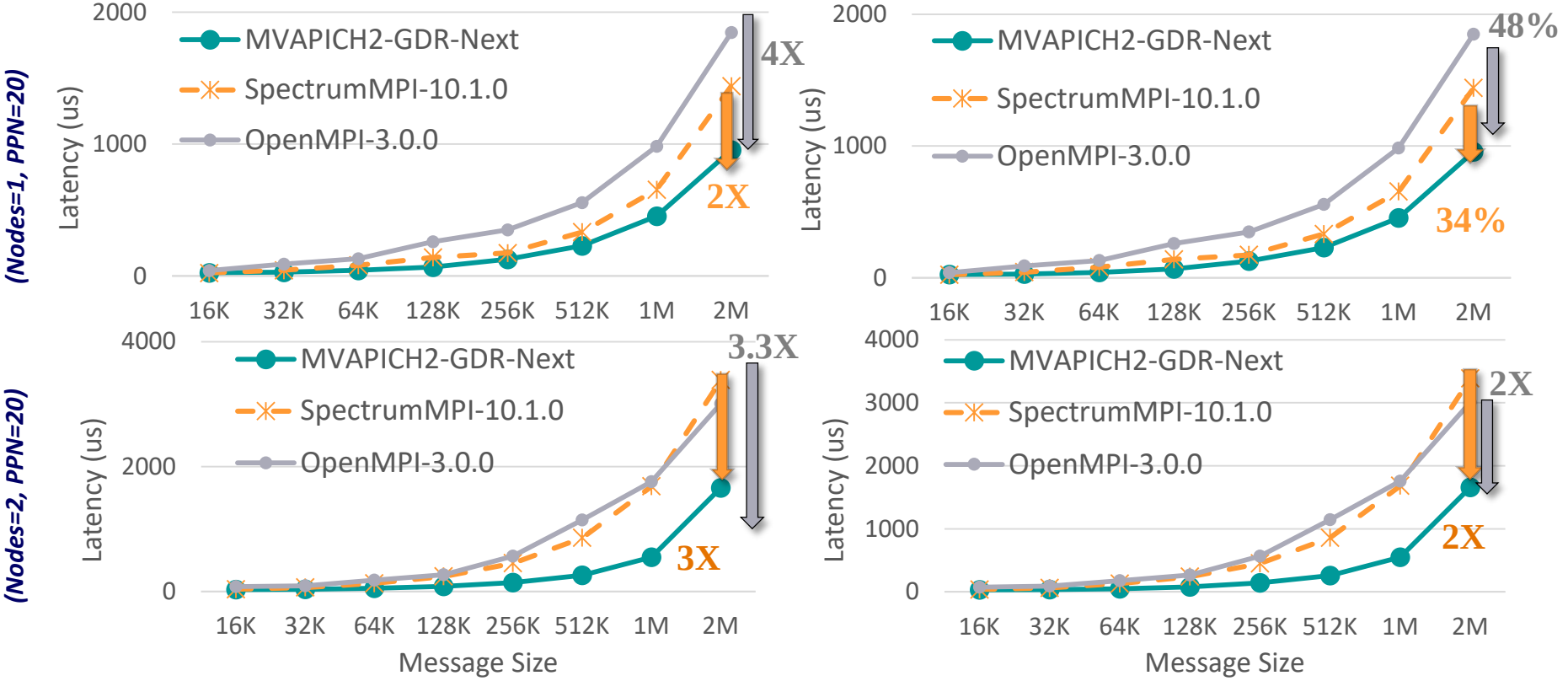
Platform: OpenPOWER (ppc64le) nodes equipped with a dual-socket CPU, 4 Pascal P100-SXM GPUs, and 4X-FDR InfiniBand Inter-connect

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



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

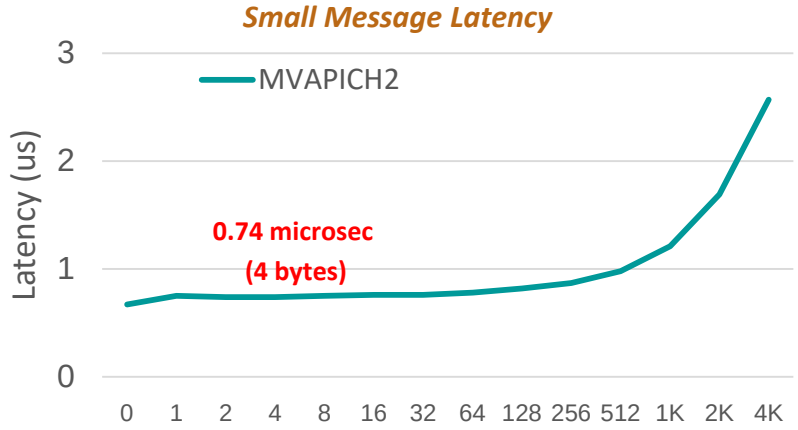
Optimized All-Reduce with XPMEM on OpenPOWER

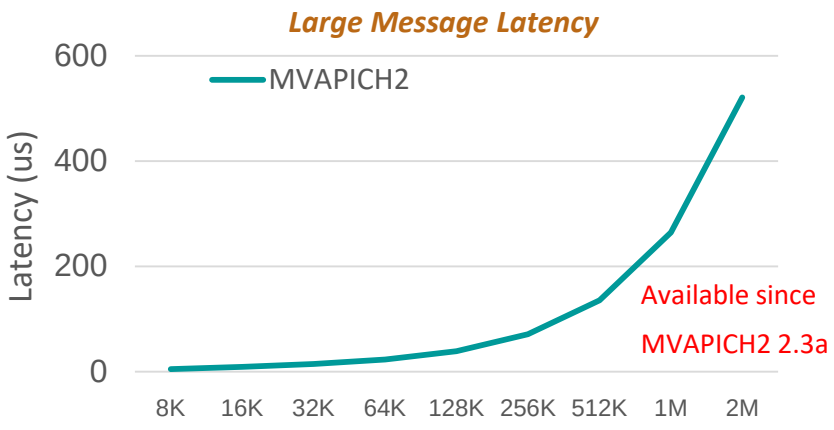


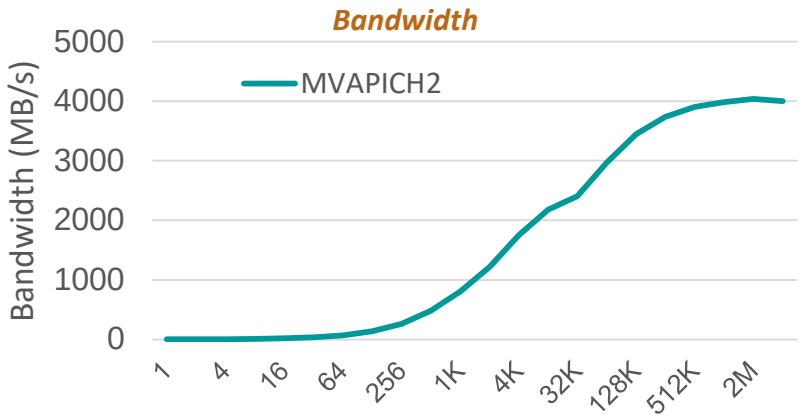
- **Optimized MPI All-Reduce Design in MVAPICH2**
  - ***Up to 2X*** performance improvement over Spectrum MPI and 4X over OpenMPI for intra-node

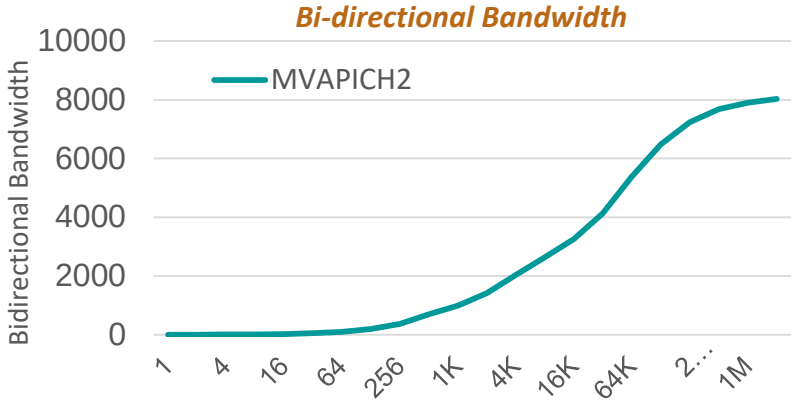
Optimized Runtime Parameters: MV2_CPU_BINDING_POLICY=hybrid MV2_HYBRID_BINDING_POLICY=bunch

Intra-node Point-to-point Performance on ARMv8







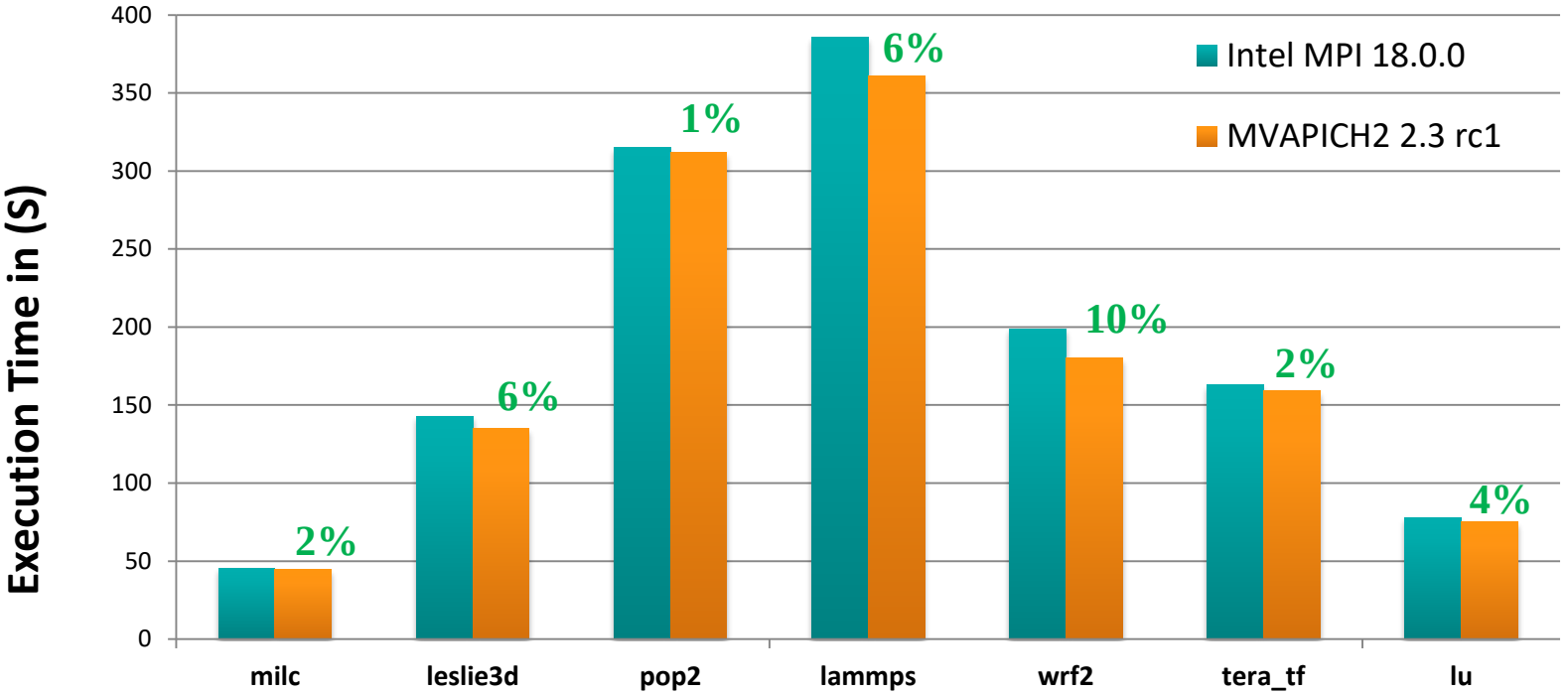


Platform: ARMv8 (aarch64) MIPS processor with 96 cores dual-socket CPU. Each socket contains 48 cores.

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

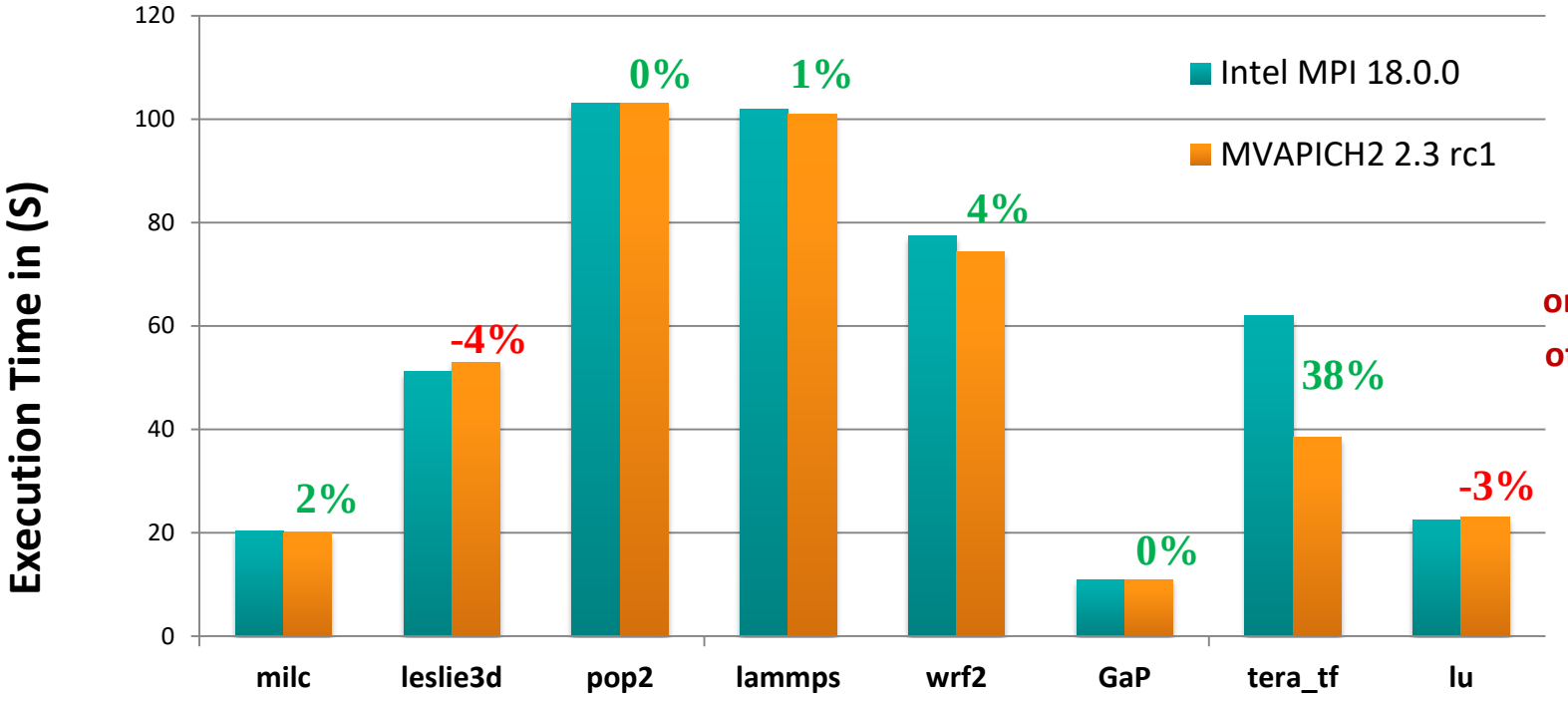
Performance of SPEC MPI 2007 Benchmarks (KNL + Omni-Path)



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

Mvapich2 outperforms Intel MPI by up to 10%

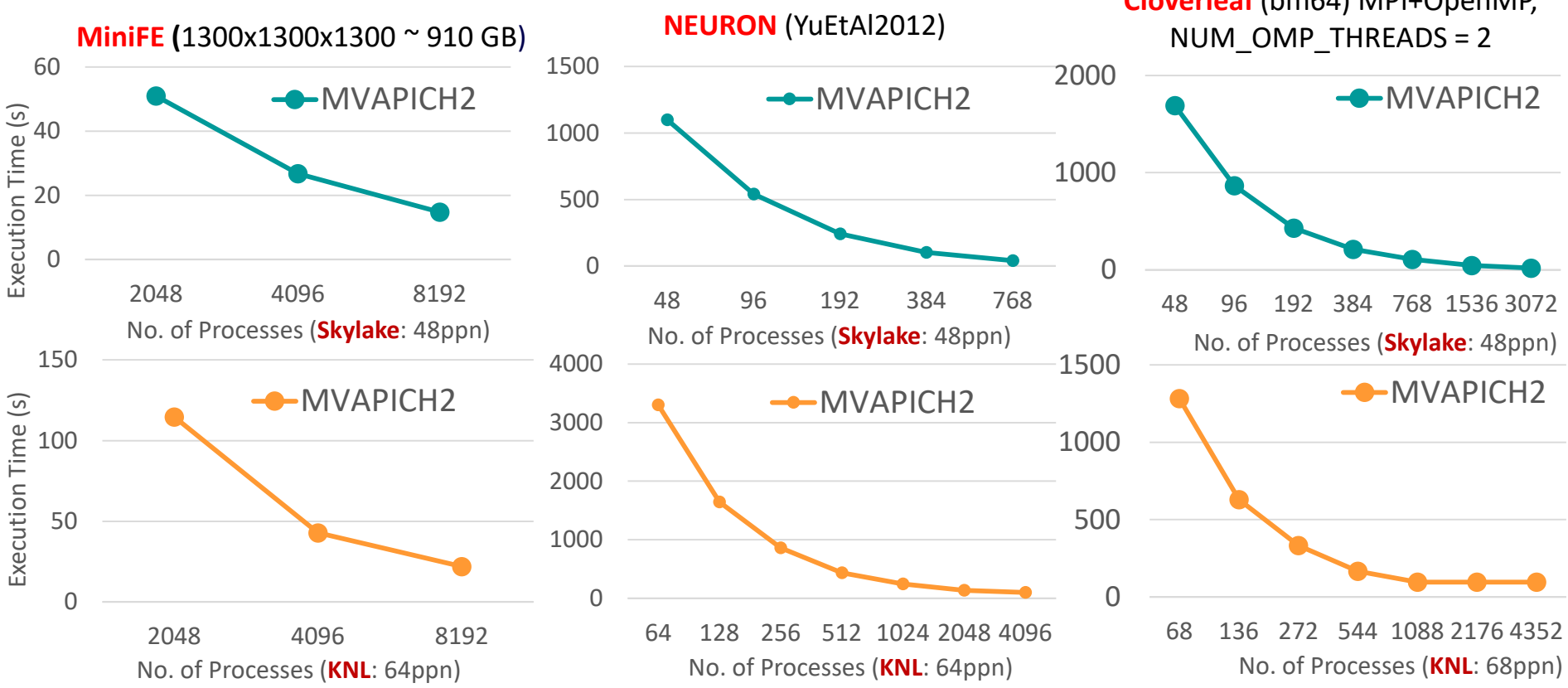
Performance of SPEC MPI 2007 Benchmarks (Skylake + Omni-Path)



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

MVAPICH2 outperforms Intel MPI by up to 38%

Application Scalability on Skylake and KNL



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*

Compilation of Best Practices

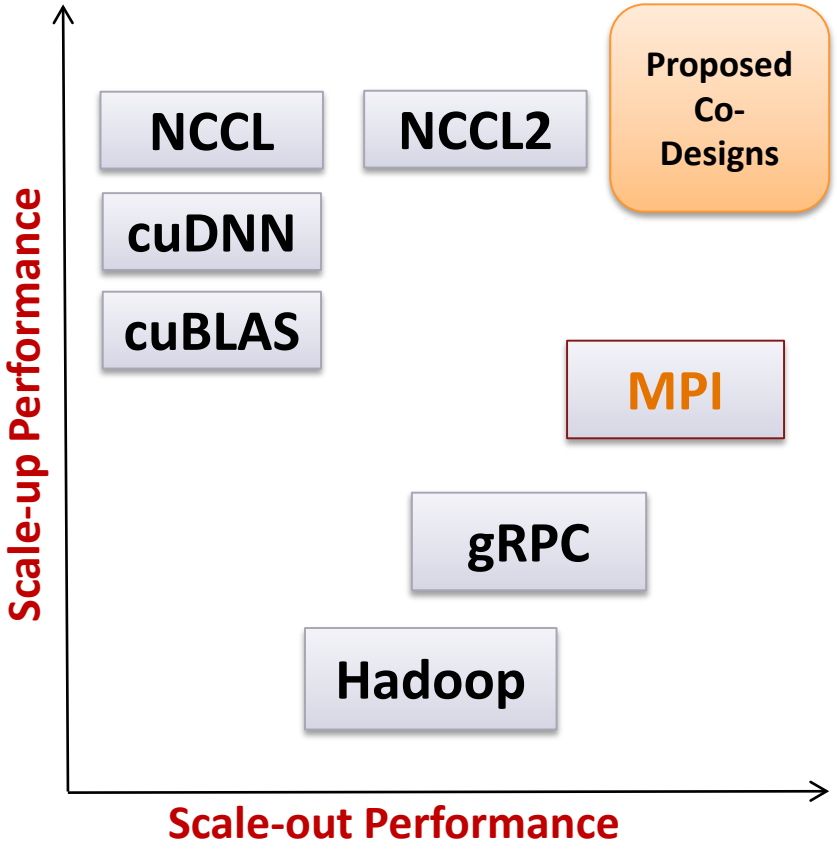
- MPI runtime has many parameters
- Tuning a set of parameters can help you to extract higher performance
- Compiled a list of such contributions through the MVAPICH Website
  - http://mvapich.cse.ohio-state.edu/best_practices/
- Initial list of applications
  - Amber
  - HoomDBlue
  - HPCG
  - Lulesh
  - 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 and Deep Learning

- Traditional HPC
  - Message Passing Interface (MPI), including MPI + OpenMP
  - Support for PGAS and MPI + PGAS (OpenSHMEM, UPC)
  - Exploiting Accelerators
- Deep Learning
  - MPI-level Challenges
  - MVAPICH2-GDR Support
  - OSU-Caffe
  - Out-of-core Processing

Deep Learning: New Challenges for MPI Runtimes

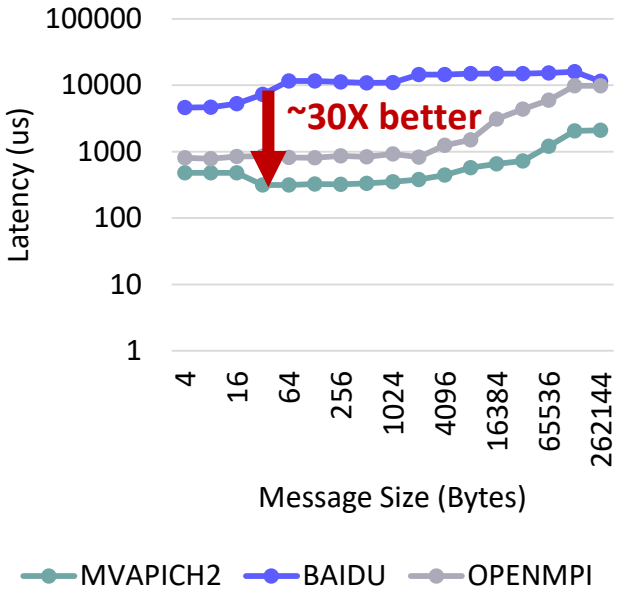
- Deep Learning frameworks are a different game altogether
  - Unusually large message sizes (order of megabytes)
  - Most communication based on GPU buffers
- Existing State-of-the-art
  - cuDNN, cuBLAS, NCCL --> **scale-up** performance
  - NCCL2, CUDA-Aware MPI --> **scale-out** performance
    -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

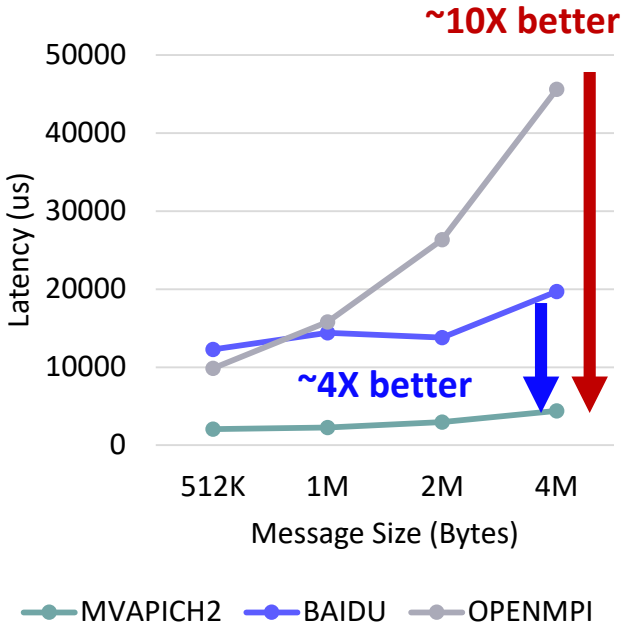


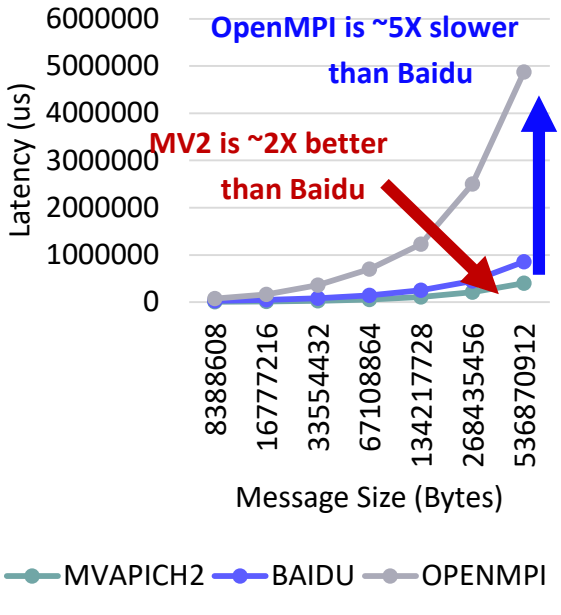
A. A. Awan, K. Hamidouche, J. M. Hashmi, and D. K. Panda, S-Caffe: Co-designing MPI Runtimes and Caffe for Scalable Deep Learning on Modern GPU Clusters. In *Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (PPoPP '17)

MVAPICH2: Allreduce Comparison with Baidu and OpenMPI

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



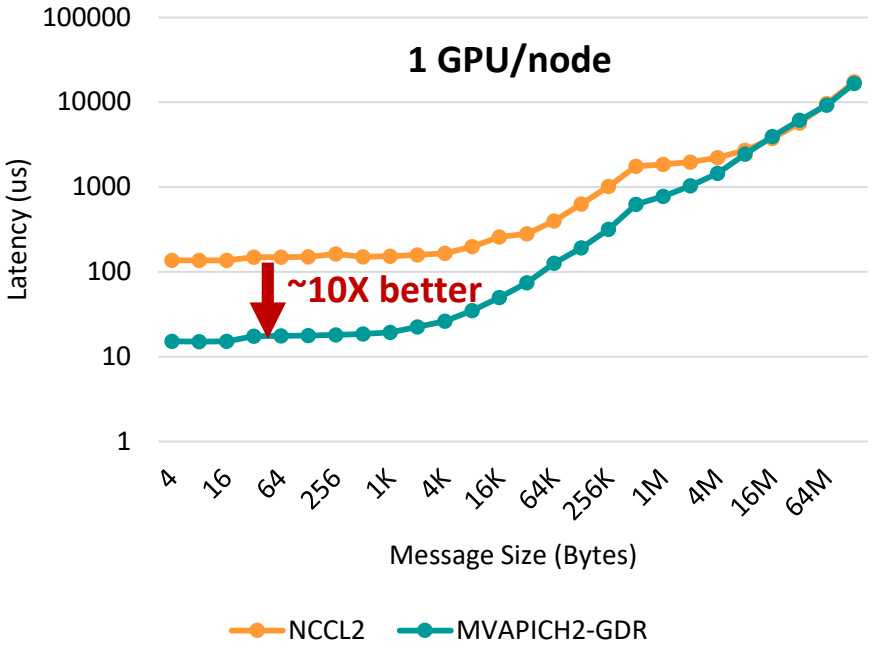


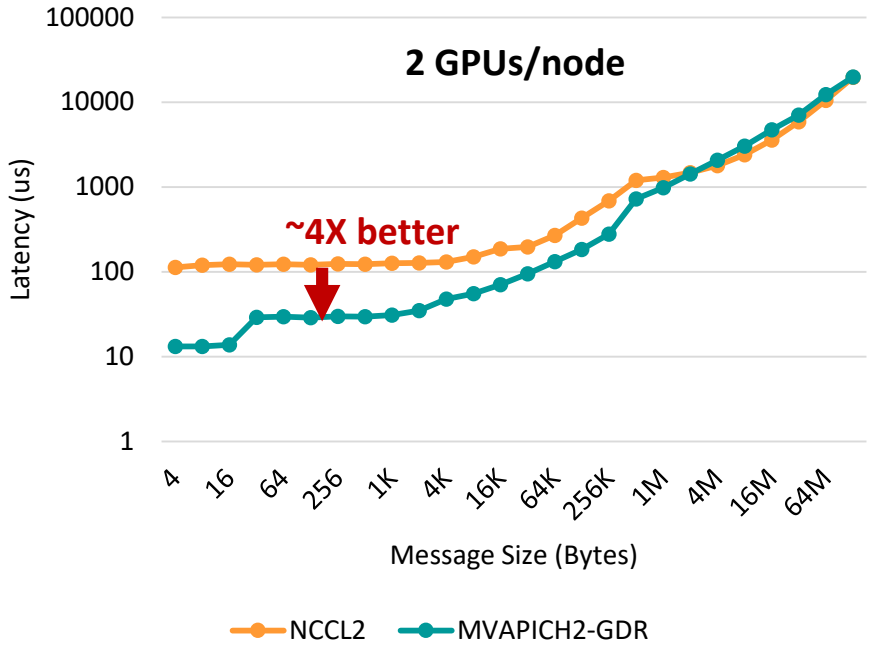


**Available with MVAPICH2-GDR 2.3a*

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



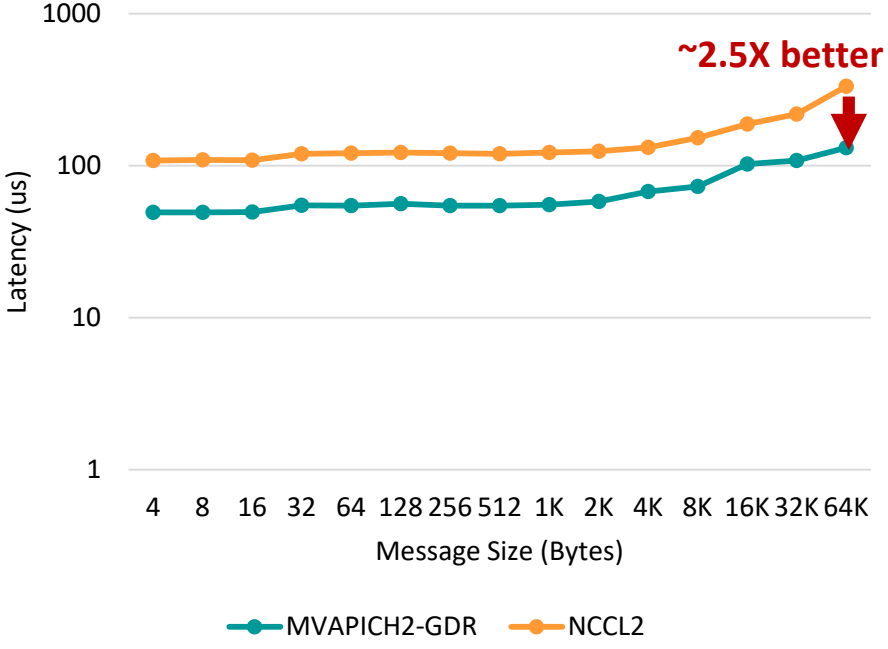


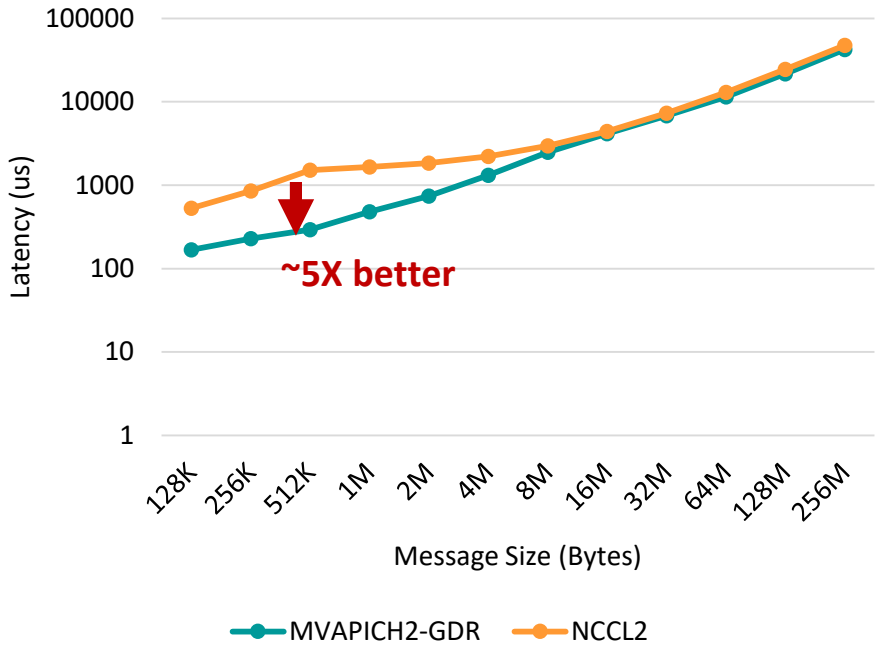
*Will be available with upcoming MVAPICH2-GDR 2.3b

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



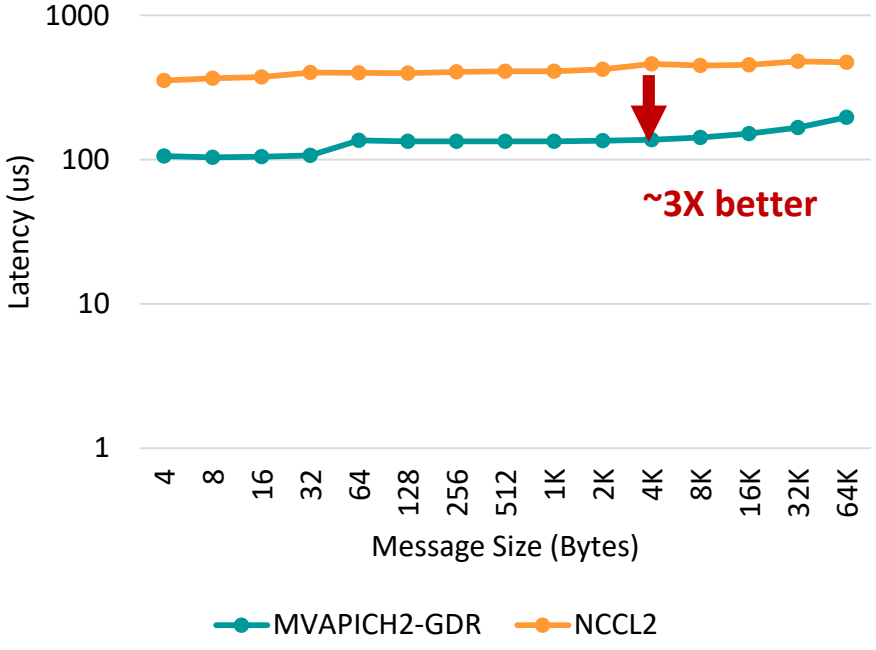


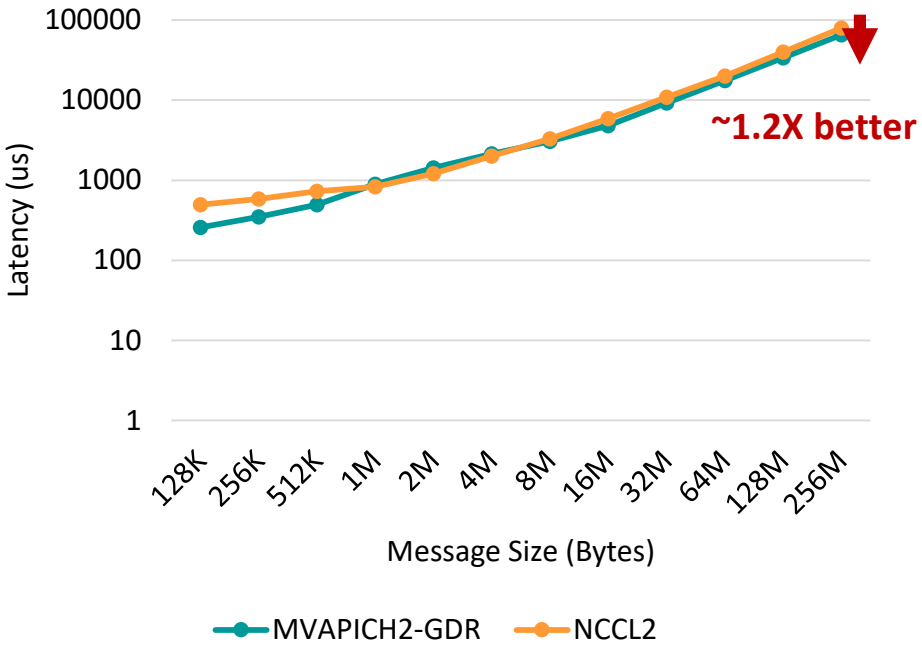
*Will be available with upcoming MVAPICH2-GDR 2.3b

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

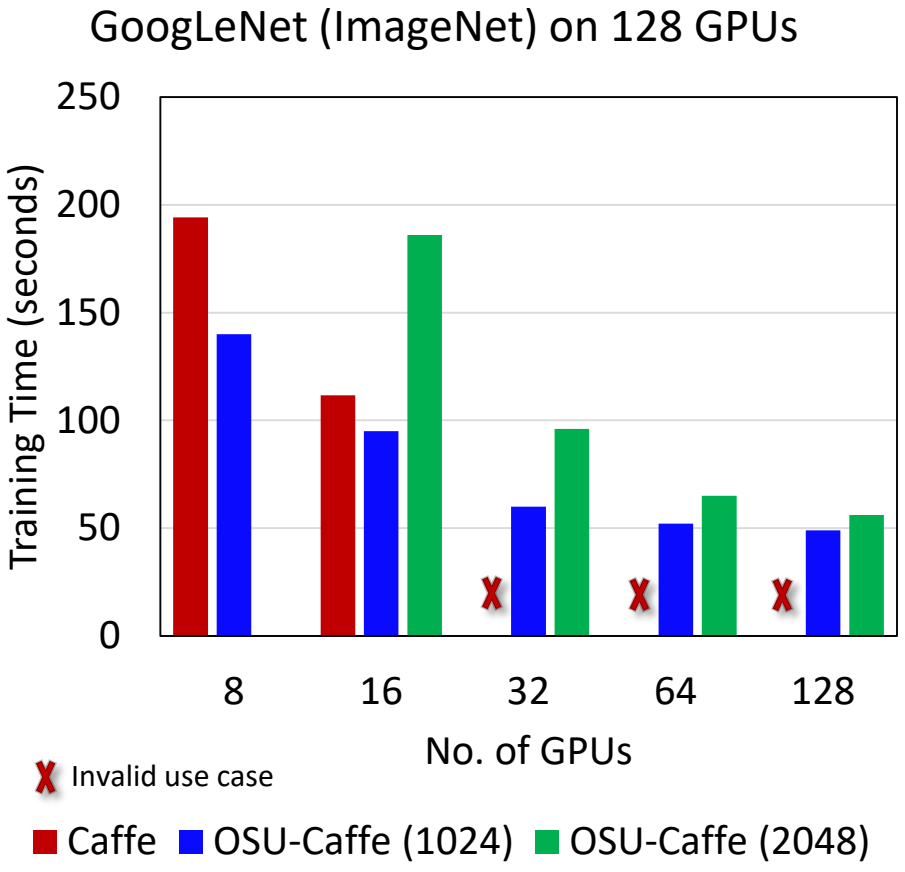
OSU-Caffe: Scalable Deep Learning

- 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



High Productivity and High Performance Out-of-Core DNN Training

- Large Size Deep Neural Networks (DNNs) cannot be trained on GPUs due to memory limitation!
  - ResNet-50 - state-of-the-art DNN architecture for Image Recognition (Trainable with a small batch size of 45)
  - Next generation models like Neural Machine Translation (NMT) are emerging that require even more memory
- Can we design Out-of-core DNN training support using new features in CUDA 8/9 and hardware mechanisms in Pascal/Volta GPUs?
- The proposed framework called OC-Caffe (Out-of-Core Caffe) shows the potential of managed memory designs that can provide performance with negligible/no overhead.
  - OC-Caffe eliminates 3,000 lines of code for a high-productivity design by exploiting Unified Memory features

Existing Design

```
class ConvolutionLayer
{
public:
   void cpu_data()
   void cpu_diff()
   void gpu_data()
   void gpu_diff()

   void mutable_cpu_data()
   void mutable_cpu_diff()
   void mutable_gpu_data()
   void mutable_gpu_diff()

   void Forward_cpu()
   void Forward_gpu()
   void forward_cpu_gemm()
   void forward_gpu_gemm()
   void forward_cpu_bias()
   void forward_gpu_bias()

   void Backward_cpu()
   void Backward_gpu()
   void backward_cpu_gemm()
   void backward_gpu_gemm()
   void backward_cpu_bias()
   void backward_gpu_bias()
}
```

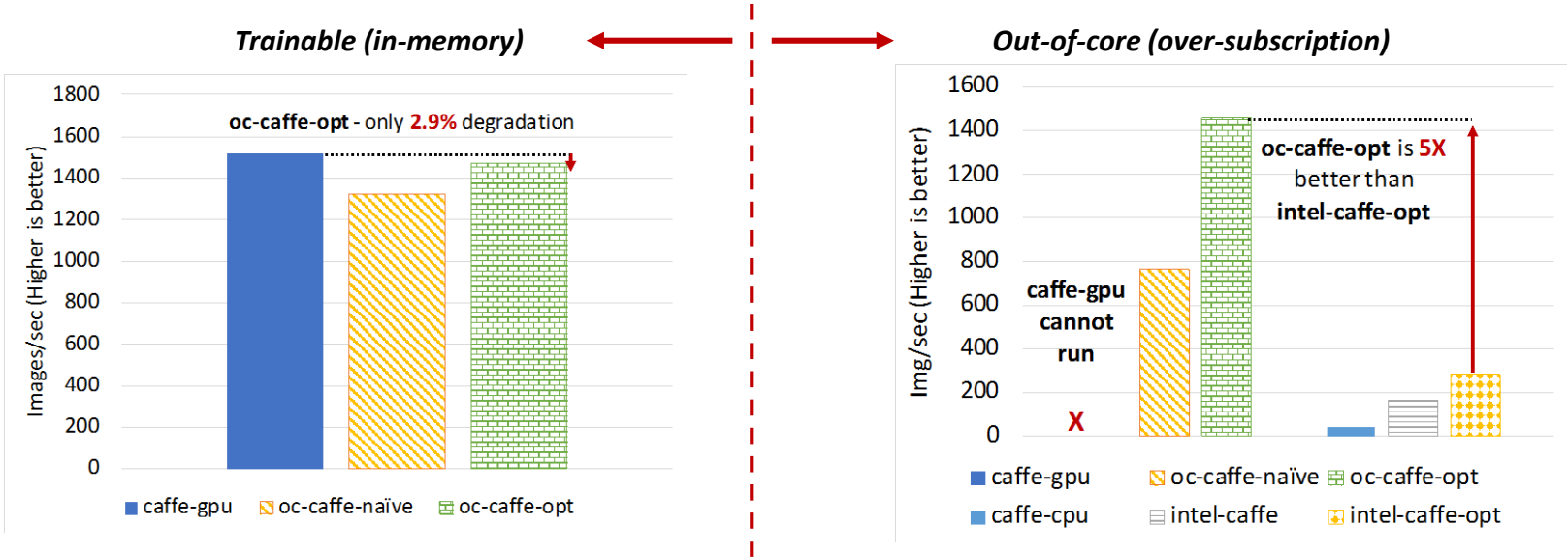
```
class ConvolutionLayer
{
public:
   void data()
   void diff()

   void mutable_data()
   void mutable_diff()

   void Forward()
   void forward_gemm()
   void forward_bias()

   void Backward()
   void backward_gemm()
   void backward_bias()
}
```

Proposed High-Productivity Design based on Managed Memory Allocation and Data Movement

Submission Under Review

Performance Trends for OC-Caffe

- Comparable performance to Caffe-Default for “in-memory” batch sizes
- OC-Caffe-Opt: up to **5X improvement** over Intel-MKL-optimized CPU-based AlexNet training on Volta V100 GPU with CUDA9 and CUDNN7



OC-Caffe will be released by the HiDL Team@OSU

hidl.cse.ohio-state.edu

Submission Under Review

MVAPICH2 – Plans for Exascale

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

Three More Presentations from the OSU Group

- Tuesday (04/10/18) at 11:30 am

  DLoBD: An Emerging Paradigm of Deep Learning over Big Data Stacks on RDMA-enabled Clusters

- Wednesday (04/11/18) at 11:30 am

  Building Efficient Clouds for HPC, Big Data, and Neuroscience Applications over SR-IOV-enabled InfiniBand Clusters

- Thursday (04/12/18) at 04:00 pm

  High-Performance Big Data Analytics with RDMA over NVM and NVMe-SSD

Funding Acknowledgments

Funding Support by

Equipment Support by

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

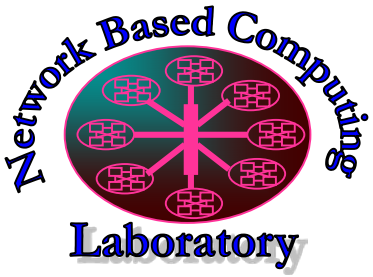
- K. Hamidouche
- S. Sur

Past Programmers

- D. Bureddy
- J. Perkins

Thank You!

subramon@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 Deep Learning Project
http://hidl.cse.ohio-state.edu/