MVAPICH

MPI, PGAS and Hybrid MPI+PGAS Library

MVAPICH2-GDR: Pushing the Frontier of HPC and Deep Learning

Talk at Mellanox booth (SC '19)

by

Dhabaleswar K. (DK) Panda

The Ohio State University

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

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

Outline

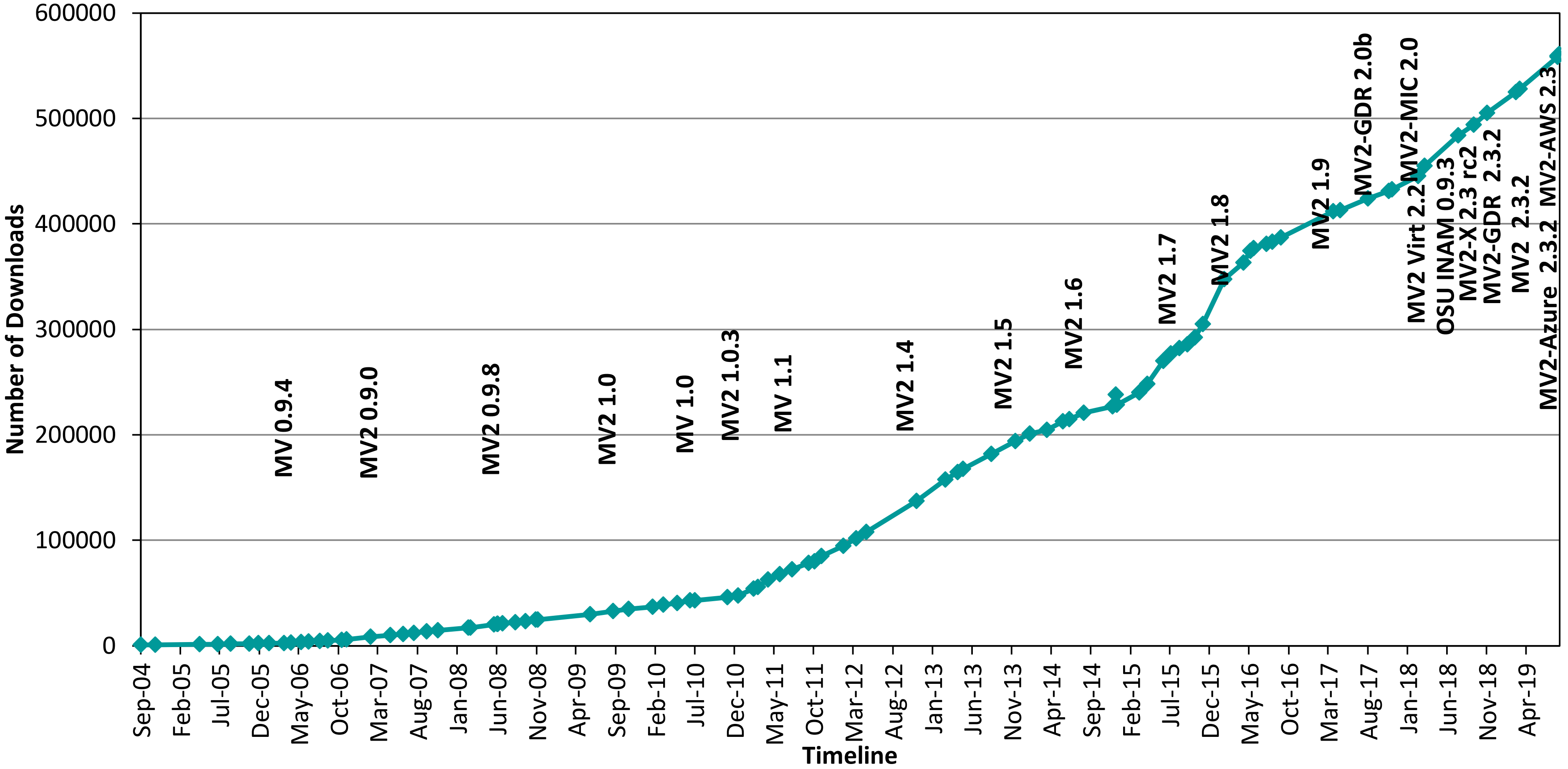
- Overview of the MVAPICH2 Project
- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- What’s new with MVAPICH2-GDR
- High-Performance Deep Learning (HiDL) with MVAPICH2-GDR
- Conclusions

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 (Supercomputing 2002)
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (MVAPICH2-GDR) and MIC (MVAPICH2-MIC), Available since 2014
  - Support for Virtualization (MVAPICH2-Virt), Available since 2015
  - Support for Energy-Awareness (MVAPICH2-EA), Available since 2015
  - Support for InfiniBand Network Analysis and Monitoring (OSU INAM) since 2015
  - **Used by more than 3,050 organizations in 89 countries**
  - **More than 615,000 (> 0.6 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Nov '19 ranking)
    - 3rd, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China
    - 5th, 448, 448 cores (Frontera) at TACC
    - 8th, 391,680 cores (ABCI) in Japan
    - 14th, 570,020 cores (Neurion) in South Korea and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)
  - **http://mvapich.cse.ohio-state.edu**
- Empowering Top500 systems for over a decade

Partner in the 5th ranked TACC Frontera System

MVAPICH2 Release Timeline and Downloads



Architecture of MVAPICH2 Software Family (HPC and DL)

High Performance Parallel Programming Models

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

High Performance and Scalable Communication Runtime

Diverse APIs and Mechanisms

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

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

Transport Protocols: RC, SRD, UD, DC

Modern Features: UMR, ODP, SR-IOV, Multi Rail

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

Transport Mechanisms: Shared Memory, CMA, IVSHMEM, XPMEM

Modern Features: Optane*, NVLink, CAPI*

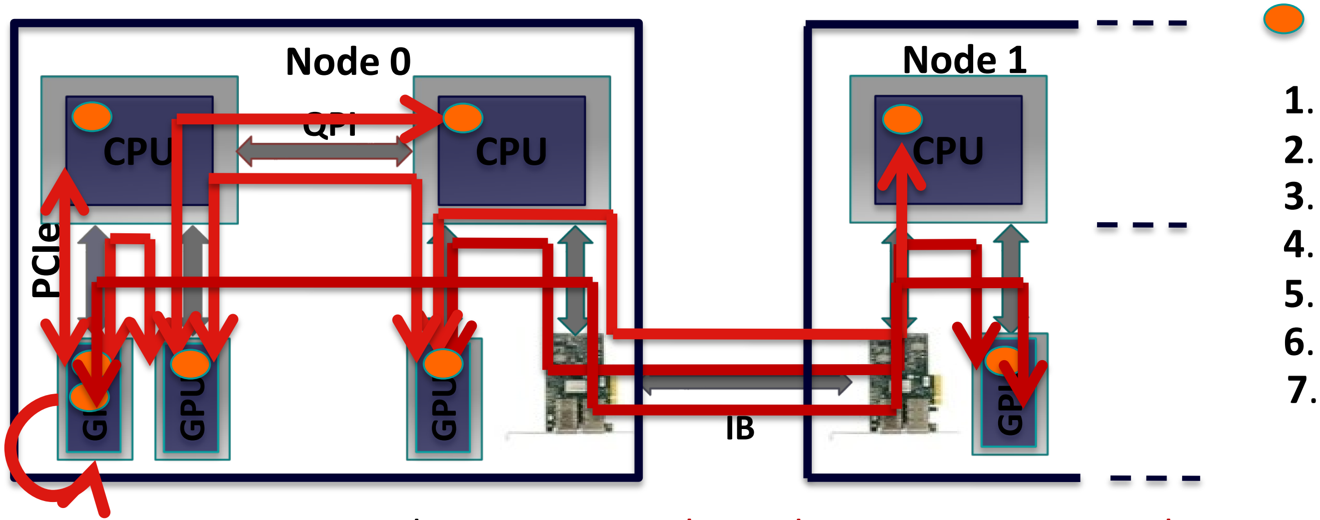
* Upcoming

MVAPICH2 Software Family

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

MVAPICH2-GDR: Optimizing MPI Data Movement on GPU Clusters

- Connected as PCIe devices – Flexibility but Complexity



1. Intra-**GPU**
2. Intra-Socket **GPU**-GPU
3. Inter-Socket **GPU**-GPU
4. Inter-Node **GPU**-GPU
5. Intra-Socket **GPU**-Host
6. Inter-Socket **GPU**-Host
7. Inter-Node **GPU**-Host

8. Inter-Node **GPU**-GPU with IB adapter on remote socket

and more . . .

- For each path different schemes: Shared_mem, IPC, GPUDirect RDMA, pipeline …
- Critical for runtimes to optimize data movement while hiding the complexity

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

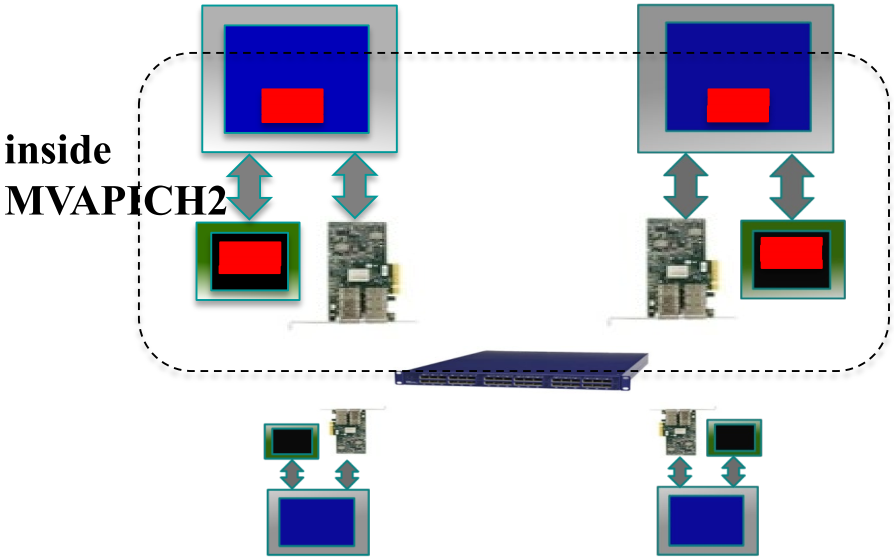
- Standard MPI interfaces used for unified data movement
- Takes advantage of Unified Virtual Addressing (>= CUDA 4.0)
- Overlaps data movement from GPU with RDMA transfers

At Sender:

MPI_Send(s_devbuf, size, …);

At Receiver:

MPI_Recv(r_devbuf, size, …);



High Performance and High Productivity

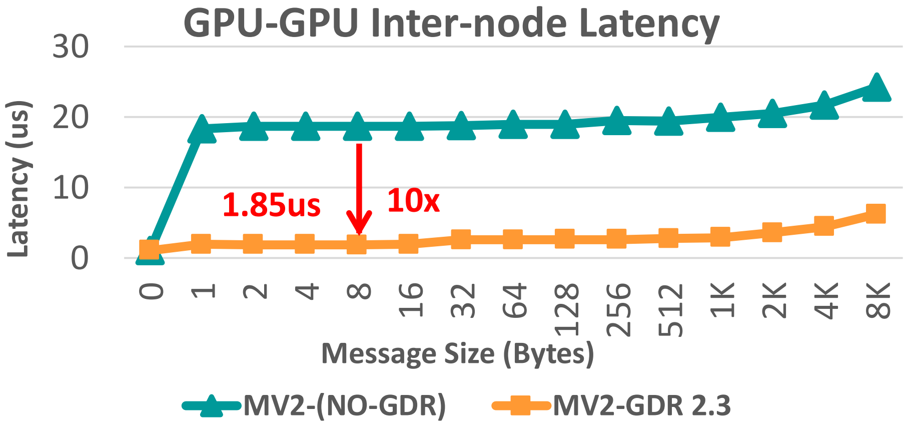
CUDA-Aware MPI: MVAPICH2-GDR 1.8-2.3 Releases

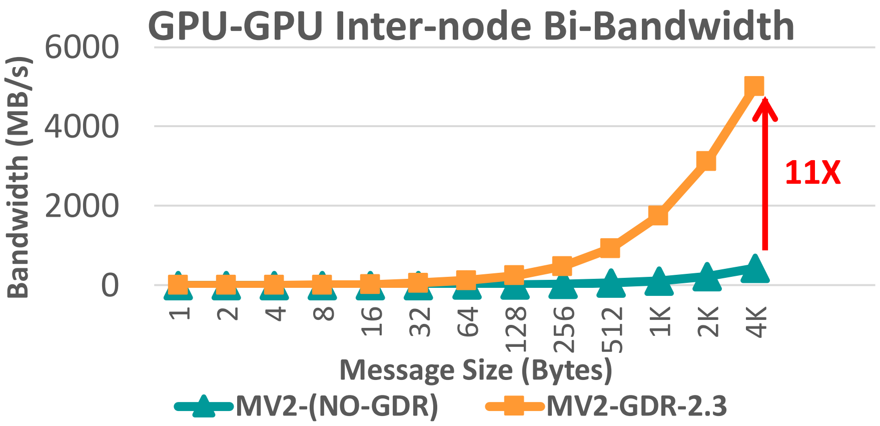
- Support for MPI communication from NVIDIA GPU device memory
- High performance RDMA-based inter-node point-to-point communication (GPU-GPU, GPU-Host and Host-GPU)
- High performance intra-node point-to-point communication for multi-GPU adapters/node (GPU-GPU, GPU-Host and Host-GPU)
- Taking advantage 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

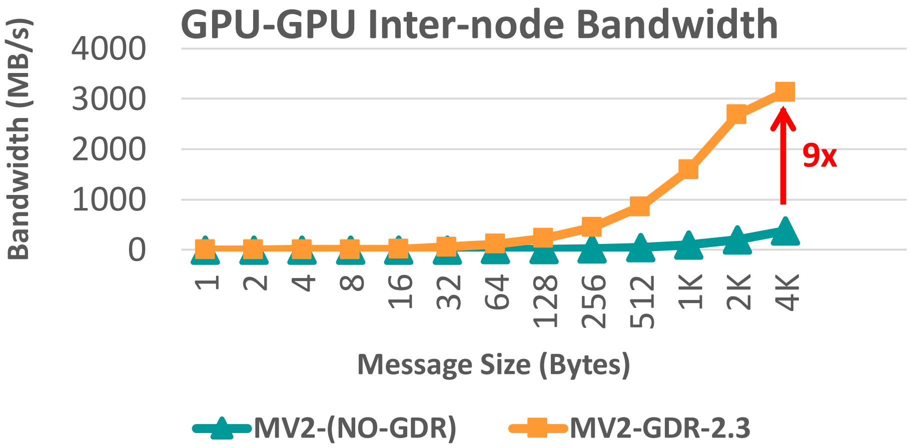
MVAPICH2-GDR 2.3.2

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

Optimized MVAPICH2-GDR Design

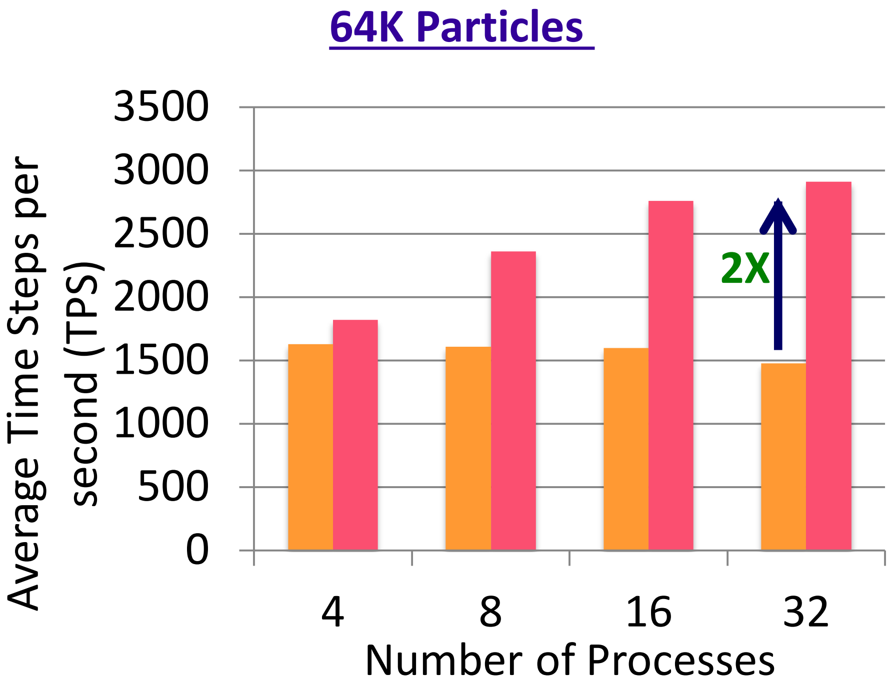


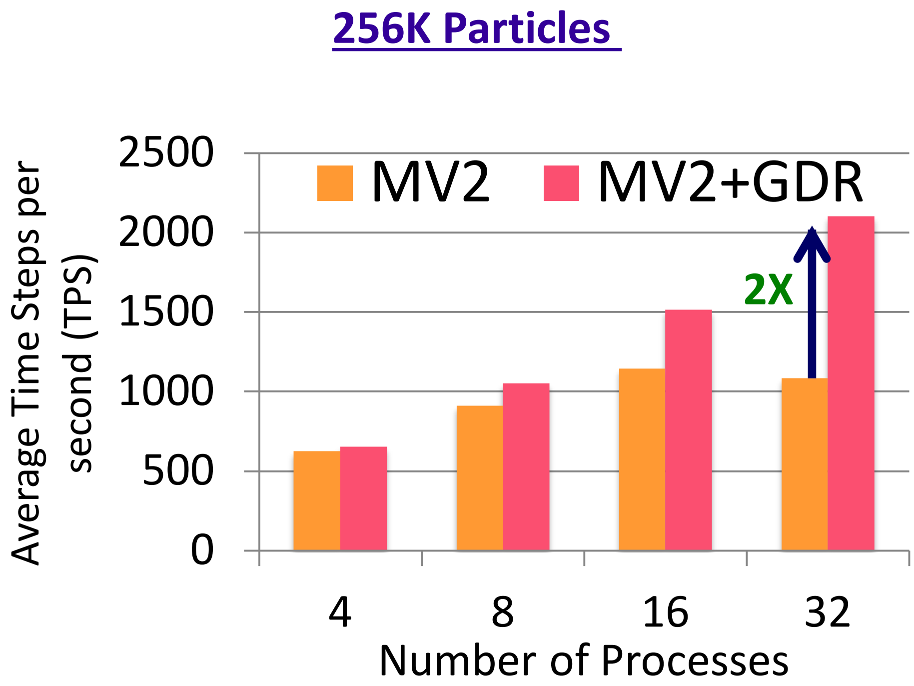




MVAPICH2-GDR-2.3
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

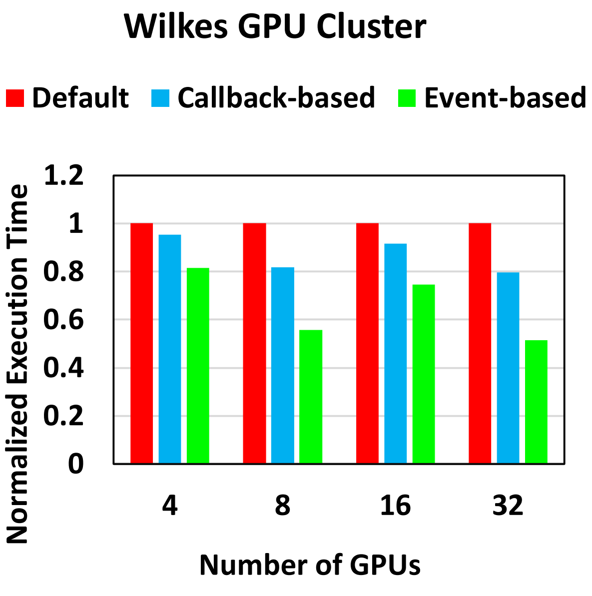
Application-Level Evaluation (HOOMD-blue)

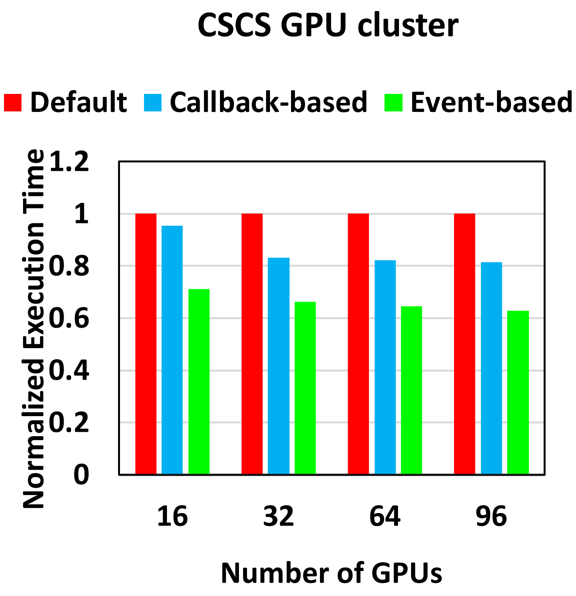


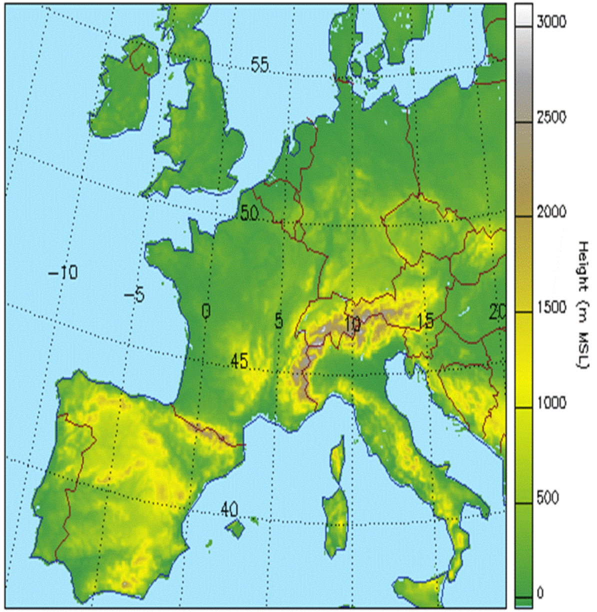


- Platform: Wilkes (Intel Ivy Bridge + NVIDIA Tesla K20c + Mellanox Connect-IB)
- HoomdBlue Version 1.0.5
  - GDRCOPY enabled: MV2_USE_CUDA=1 MV2_IBA_HCA=mlx5_0 MV2_IBA_EAGER_THRESHOLD=32768 MV2_VBUF_TOTAL_SIZE=32768 MV2_USE_GPUDIRECT_LOOPBACK_LIMIT=32768 MV2_USE_GPUDIRECT_GDRCOPY=1 MV2_USE_GPUDIRECT_GDRCOPY_LIMIT=16384

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

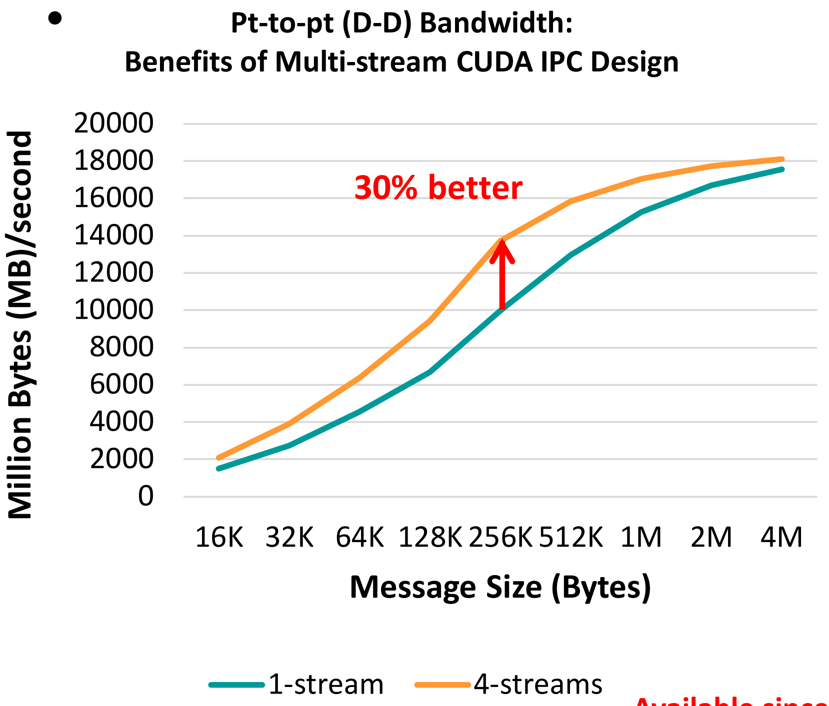
C. Chu, K. Hamidouche, A. Venkatesh, D. Banerjee , H. Subramoni, and D. K. Panda, Exploiting Maximal Overlap for Non-Contiguous Data Movement Processing on Modern GPU-enabled Systems, IPDPS’16

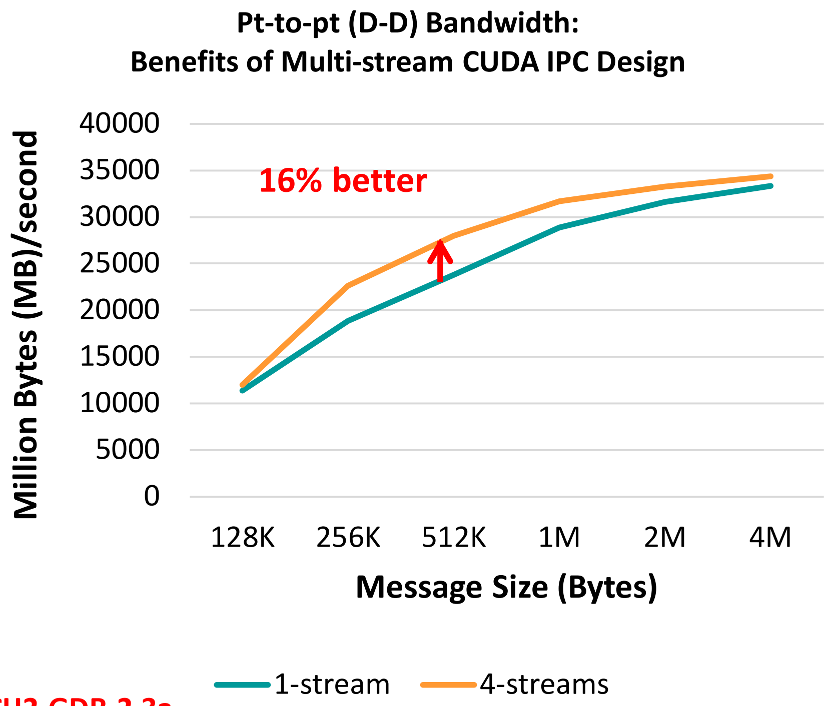
Outline

- Overview of the MVAPICH2 Project
- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- What's new with MVAPICH2-GDR
  - Multi-stream Communication for IPC
  - CMA-based Intra-node Communication Support
  - Support for OpenPower and NVLink with GDRCOPY2
  - Maximal overlap in MPI Datatype Processing
- High-Performance Deep Learning (HiDL) with MVAPICH2-GDR
- Conclusions

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

- Up to **16% higher** Device to Device (D2D) bandwidth on OpenPOWER + NVLink inter-connect
- Up to **30% higher** D2D bandwidth on DGX-1 with NVLink
-

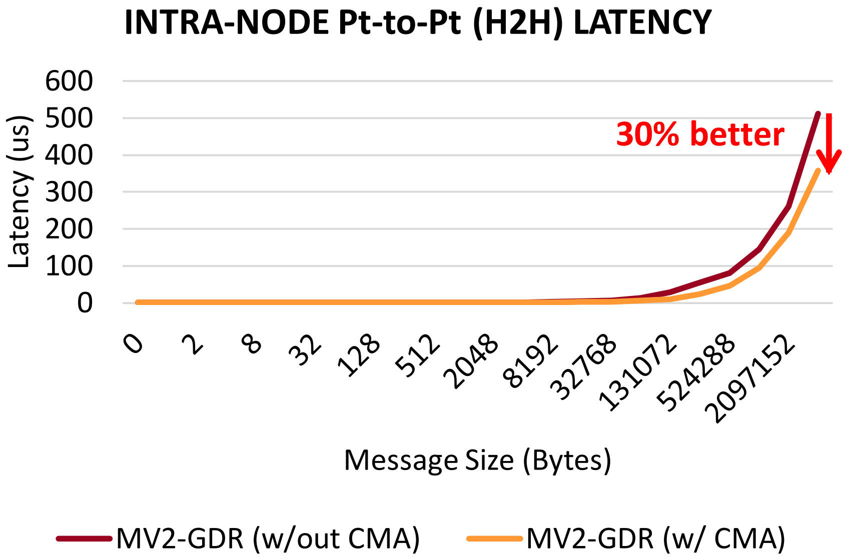


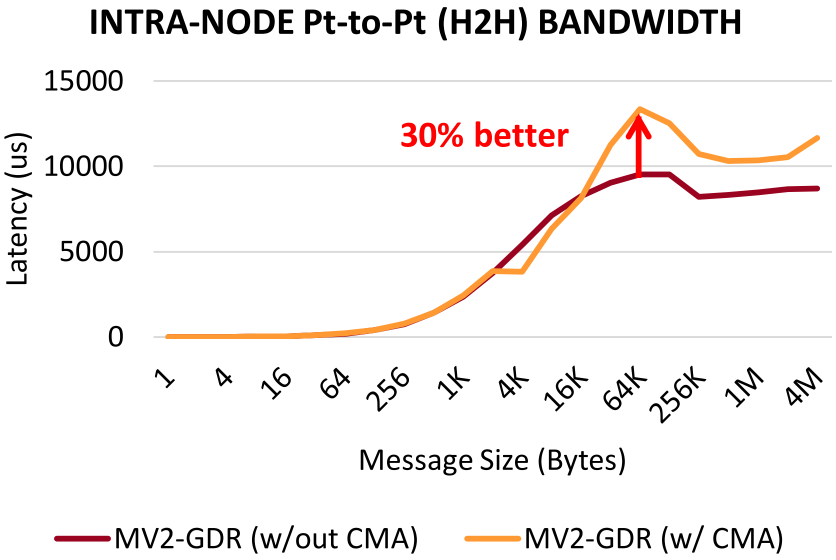


Available since MVAPICH2-GDR-2.3a

CMA-based Intra-node Communication Support

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

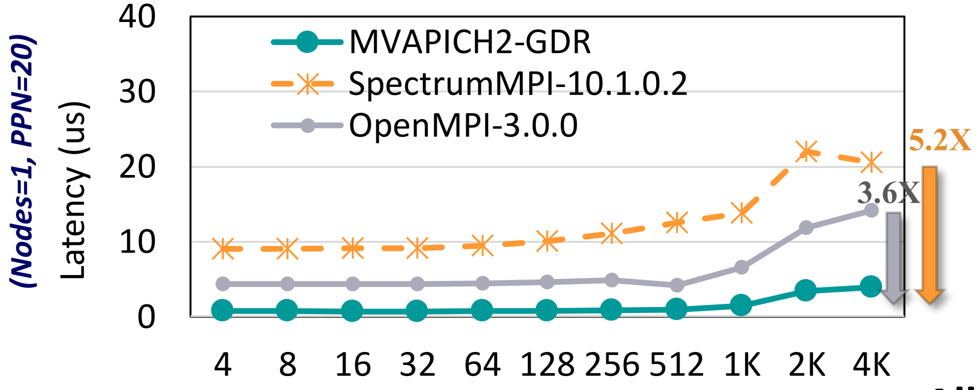


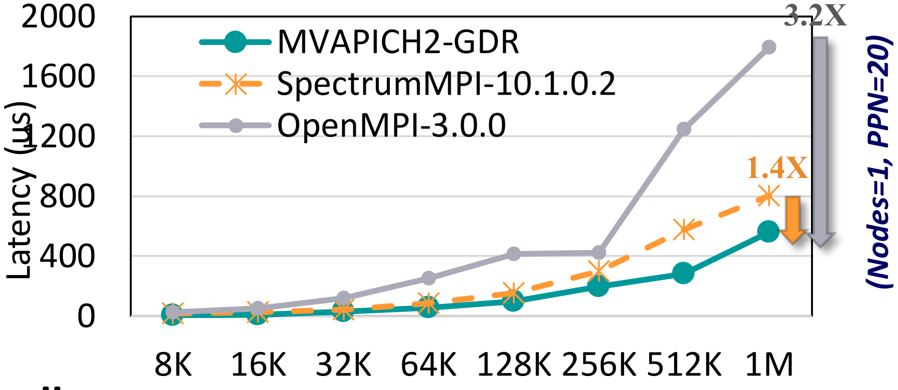


MVAPICH2-GDR-2.3.2
Intel Broadwell (E5-2680 v4 @ 3240 GHz) node – 28 cores
NVIDIA Tesla K-80 GPU, and Mellanox Connect-X4 EDR HCA
CUDA 8.0, Mellanox OFED 4.0 with GPU-Direct-RDMA

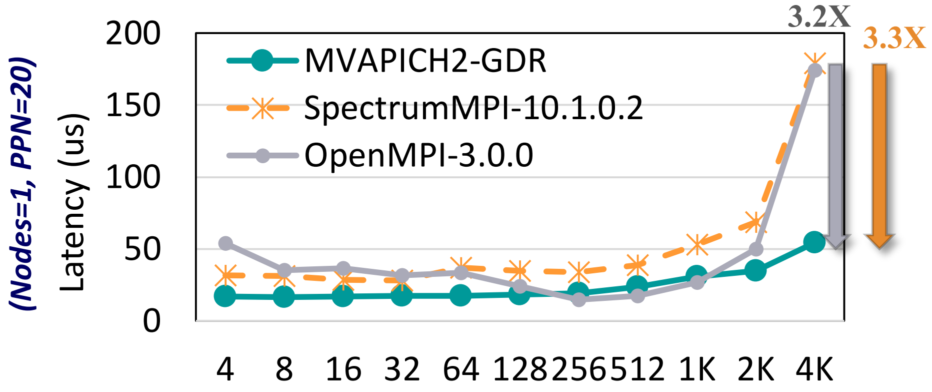
Scalable Host-based Collectives on OpenPOWER (Intra-node Reduce & AlltoAll)

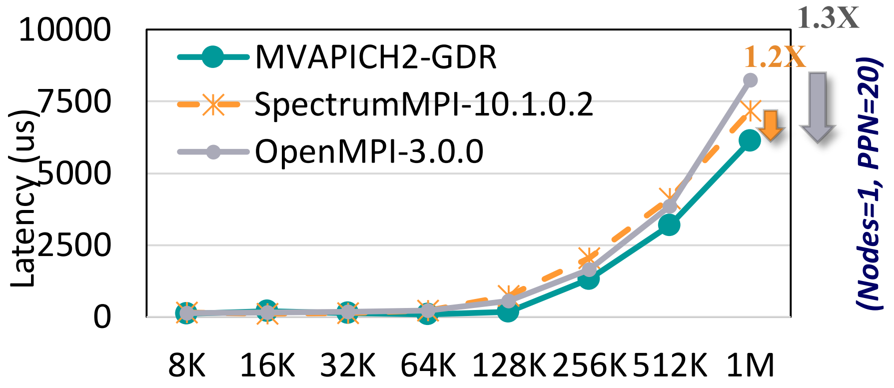
Reduce





Alltoall

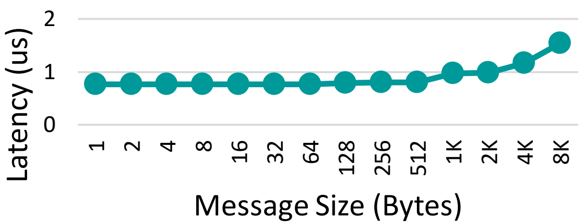




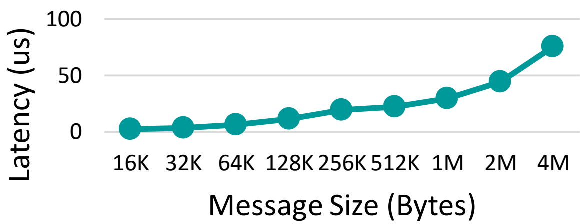
Up to 5X and 3x performance improvement by MVAPICH2 for small and large messages respectively

D-to-D Performance on OpenPOWER w/ GDRCopy (NVLink2 + Volta)

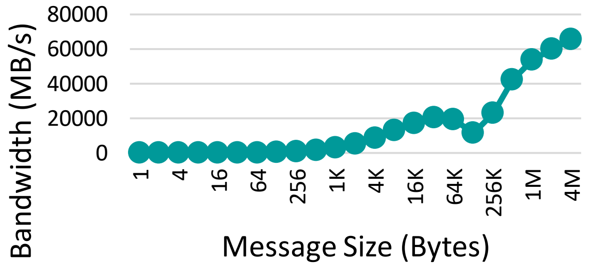
Intra-Node Latency (Small Messages)



Intra-Node Latency (Large Messages)



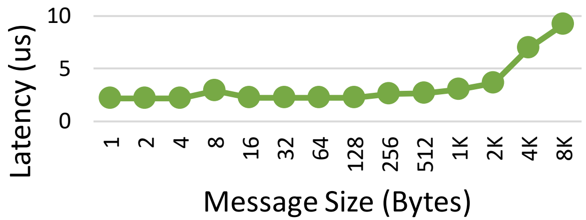
Intra-Node Bandwidth



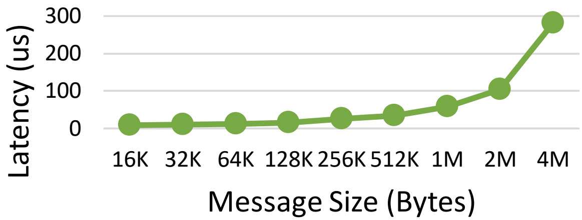
*Intra-node Latency: **0.76 us** (with GDRCopy)*

*Intra-node Bandwidth: **65.48 GB/sec** for 4MB (via NVLINK2)*

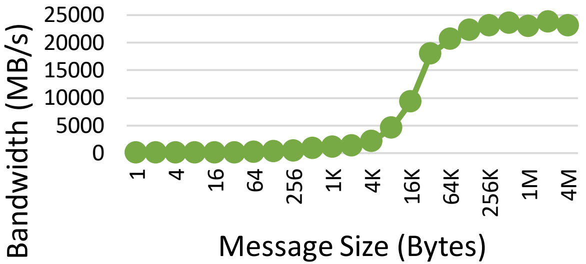
Inter-Node Latency (Small Messages)



Inter-Node Latency (Large Messages)



Inter-Node Bandwidth

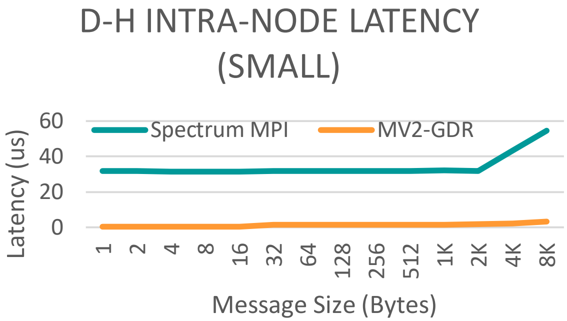


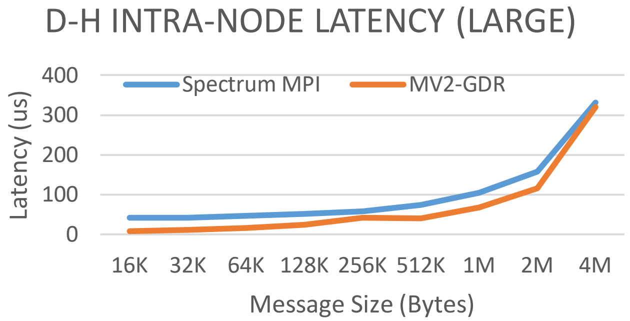
*Inter-node Latency: **2.18 us** (with GDRCopy 2.0)*

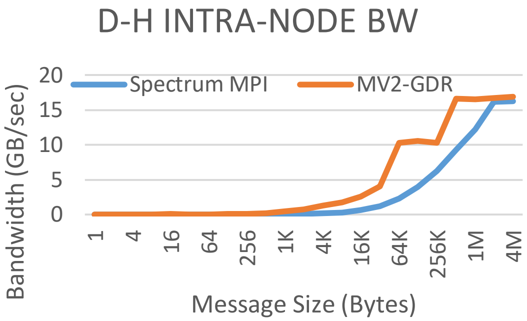
*Inter-node Bandwidth: **23 GB/sec** for 4MB (via 2 Port EDR)*

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

D-to-H & H-to-D Performance on OpenPOWER w/ GDRCopy (NVLink2 + Volta)

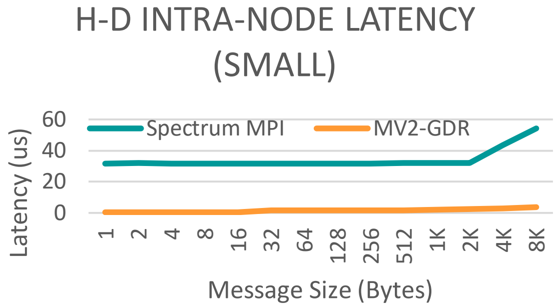


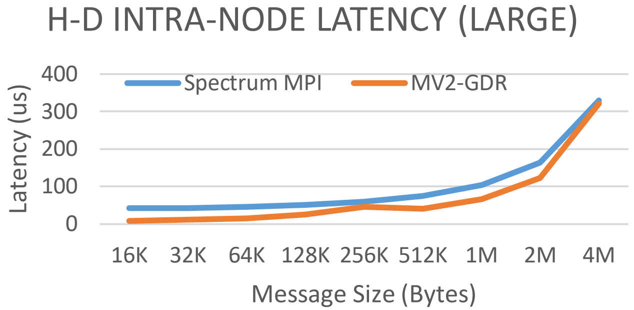


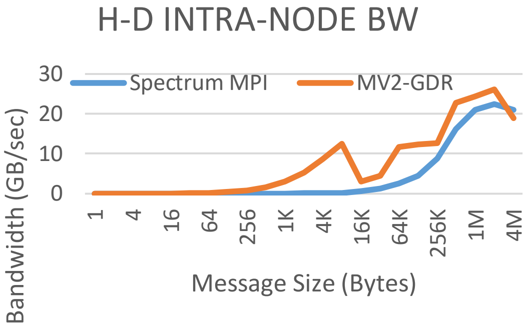


Intra-node D-H Latency: 0.49 us (with GDRCopy)

Intra-node D-H Bandwidth: 16.70 GB/sec for 2MB (via NVLINK2)





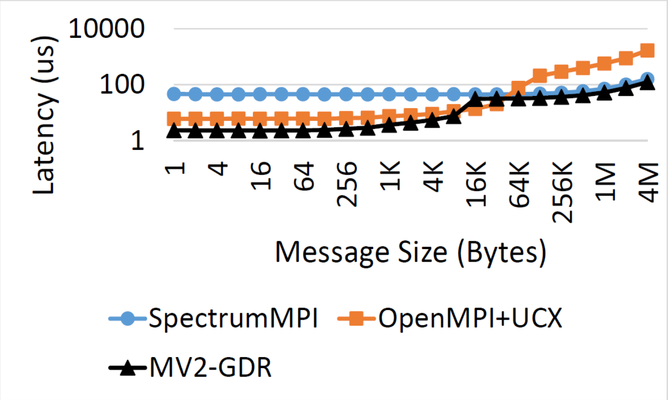


Intra-node H-D Latency: 0.49 us (with GDRCopy 2.0)

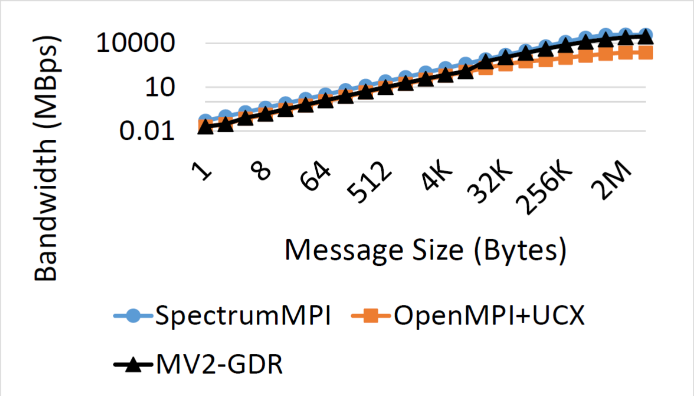
Intra-node H-D Bandwidth: 26.09 GB/sec for 2MB (via NVLINK2)

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

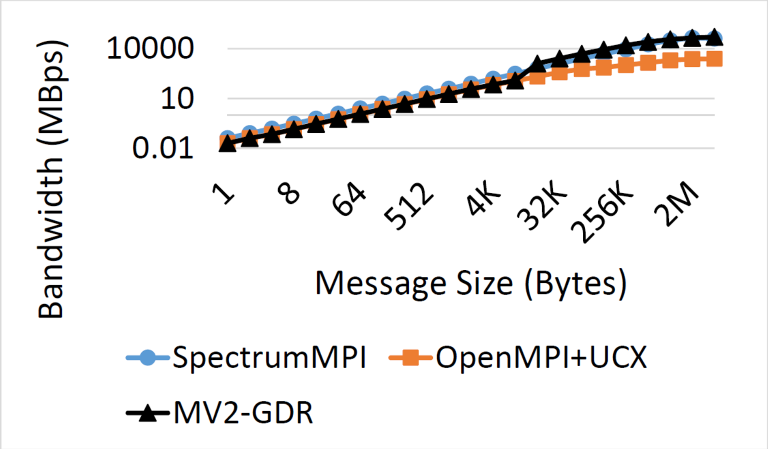
Managed Memory Performance (OpenPOWER Intra-node)



Latency MD MD

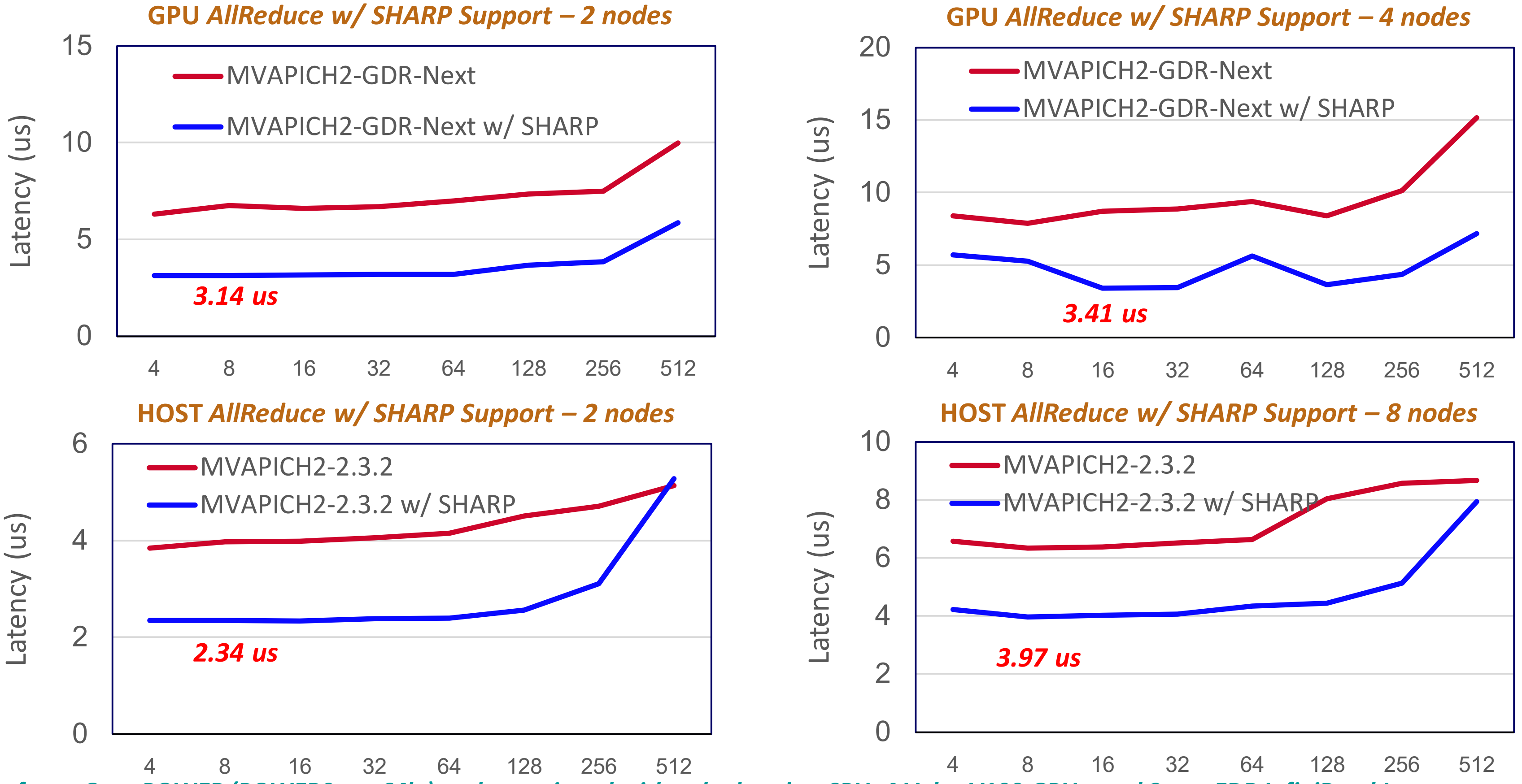


Bandwidth MD MD



Bi-Bandwidth MD MD

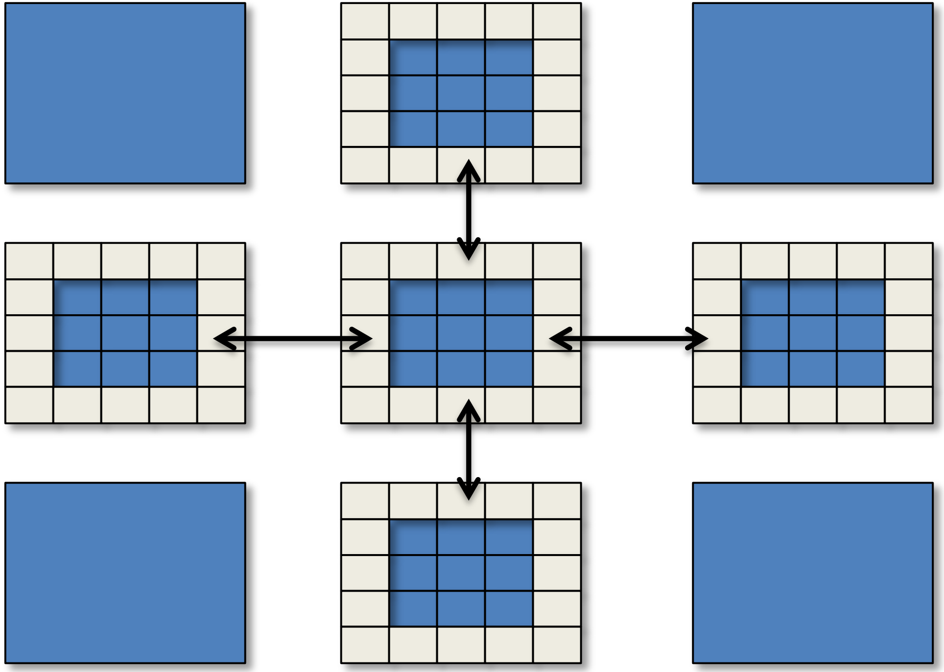
MVAPICH2 with SHARP Support (Preliminary Results)



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

Non-contiguous Data Exchange

Halo data exchange



- Multi-dimensional data
  - Row based organization
  - Contiguous on one dimension
  - Non-contiguous on other dimensions
- Halo data exchange
  - Duplicate the boundary
  - Exchange the boundary in each iteration

MPI Datatype support in MVAPICH2

- Datatypes support in MPI
  - Operate on customized datatypes to improve productivity
  - Enable MPI library to optimize non-contiguous data

At Sender:

MPI_Type_vector (n_blocks, n_elements, stride, old_type, &new_type);
MPI_Type_commit(&new_type);
…
MPI_Send(s_buf, size, new_type, dest, tag, MPI_COMM_WORLD);

- Inside MVAPICH2
  - Use datatype specific CUDA Kernels to pack data in chunks
  - Efficiently move data between nodes using RDMA
  - In progress - currently optimizes *vector* and *hindexed* datatypes
  - Transparent to the user

H. Wang, S. Potluri, D. Bureddy, C. Rosales and D. K. Panda, GPU-aware MPI on RDMA-Enabled Clusters: Design, Implementation and Evaluation, IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 10, pp. 2595-2605 , Oct 2014.

MPI Datatype Processing (Computation Optimization)

- Comprehensive support
  - Targeted kernels for regular datatypes - vector, subarray, indexed_block
  - Generic kernels for all other irregular datatypes
- Separate non-blocking stream for kernels launched by MPI library
  - Avoids stream conflicts with application kernels
- Flexible set of parameters for users to tune kernels
  - Vector
    - MV2_CUDA_KERNEL_VECTOR_TIDBLK_SIZE
    - MV2_CUDA_KERNEL_VECTOR_YSIZE
  - Subarray
    - MV2_CUDA_KERNEL_SUBARR_TIDBLK_SIZE
    - MV2_CUDA_KERNEL_SUBARR_XDIM
    - MV2_CUDA_KERNEL_SUBARR_YDIM
    - MV2_CUDA_KERNEL_SUBARR_ZDIM
  - Indexed_block
    - MV2_CUDA_KERNEL_IDXBLK_XDIM

MPI Datatype Processing (Communication Optimization)

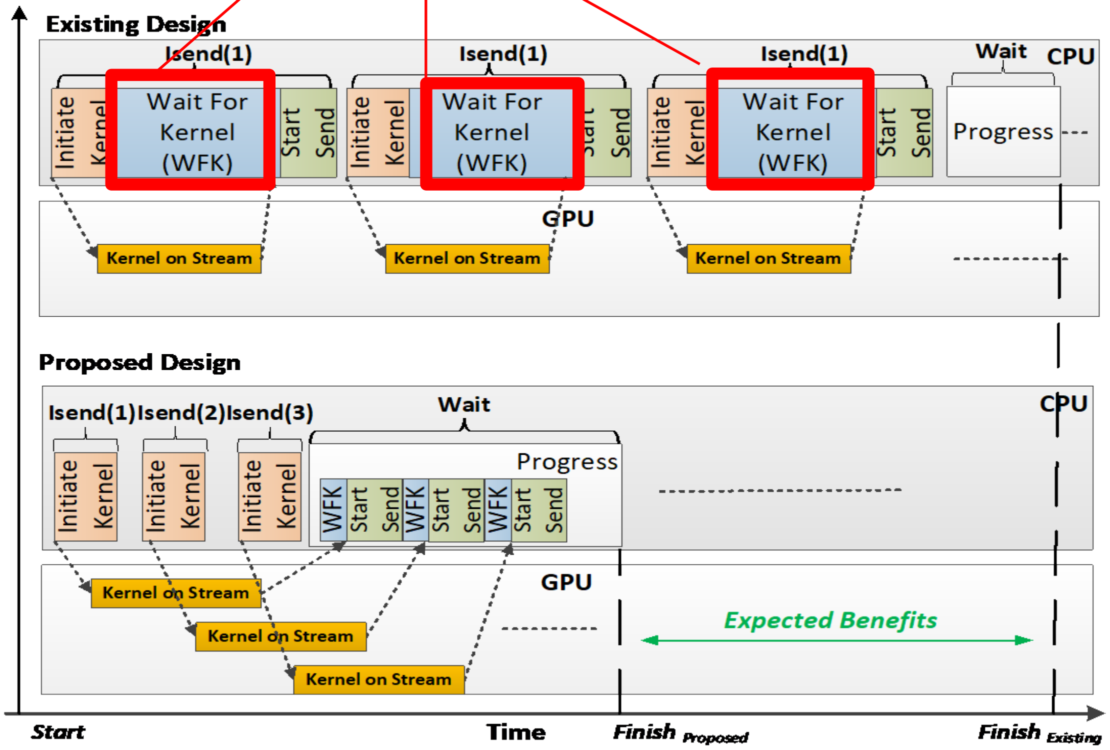
Common Scenario

MPI_Isend (A,.. Datatype,…)
MPI_Isend (B,.. Datatype,…)
MPI_Isend (C,.. Datatype,…)
MPI_Isend (D,.. Datatype,…)
…

MPI_Waitall (…);

*A, B…contain non-contiguous MPI Datatype

Waste of computing resources on CPU and GPU



Application: COMB

Run Scripts pushed to COMB Github repo: https://github.com/LLNL/Comb/pull/2

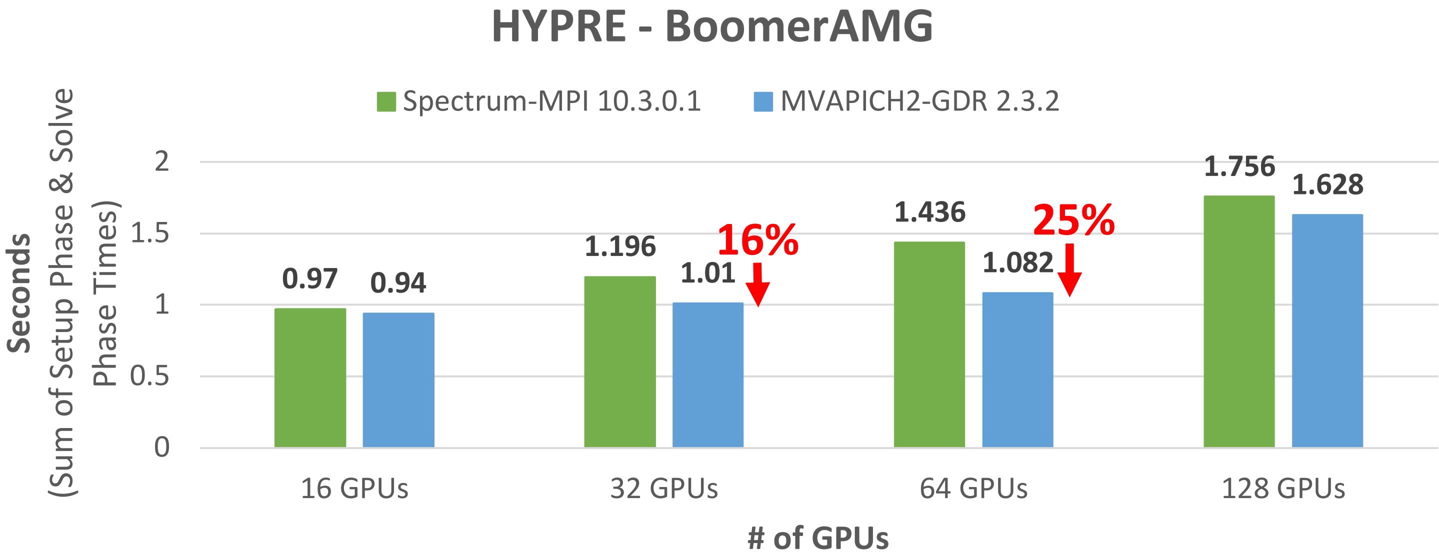
16 GPUs on POWER9 system (test Comm mpi Mesh cuda Device Buffers mpi_type)									
	pre-comm	post-recv	post-send	wait-recv	wait-send	post-comm	start-up	test-comm	bench-comm
Spectrum MPI 10.3	0.0001	0.0000	1.6021	1.7204	0.0112	0.0001	0.0004	7.7383	**83.6229**
MVAPICH2-GDR 2.3.2	0.0001	0.0000	0.0862	0.0871	0.0018	0.0001	0.0009	0.3558	**4.4396**
MVAPICH2-GDR 2.3.3 (Upcoming)	0.0001	0.0000	0.0030	0.0032	0.0001	0.0001	0.0009	0.0133	**0.1602**

18x

27x

- **Improvements due to enhanced support for GPU-kernel based packing/unpacking routines**

Application: HYPRE - BoomerAMG



RUN MVAPICH2-GDR 2.3.2:

export MV2_USE_CUDA=1 MV2_USE_GDRCOPY=0 MV2_USE_RDMA_CM=0

export MV2_USE_GPUDIRECT_LOOPBACK=0 MV2_HYBRID_BINDING_POLICY=spread MV2_IBA_HCA=mlx5_0:mlx5_3

OMP_NUM_THREADS=20 lrun -n 128 -N 32 mpibind ./ij -P 8 4 4 -n 50 50 50 -pmis -Pmx 8 -keepT 1 -rlx 18

RUN Spectrum-MPI 10.3.0.1:

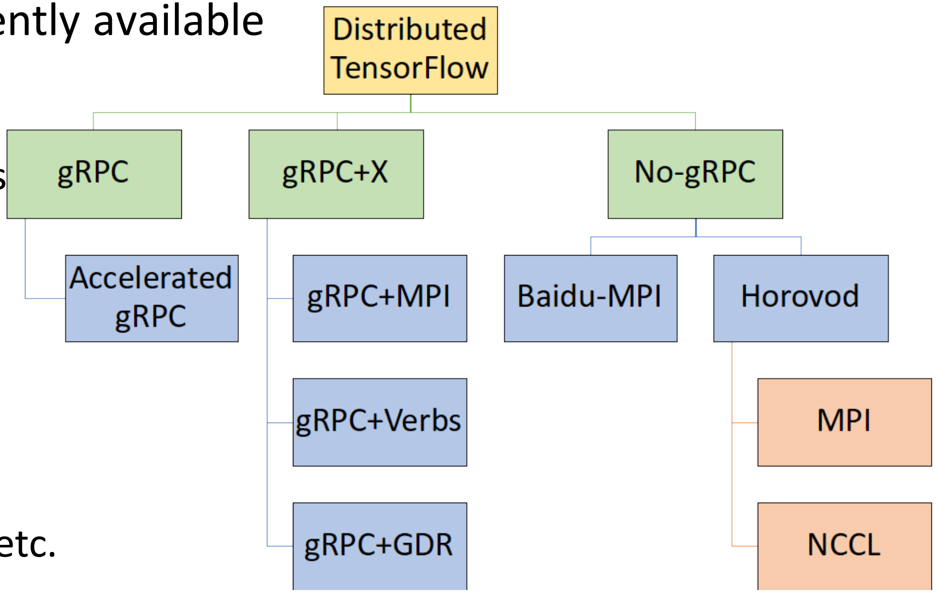
OMP_NUM_THREADS=20 lrun -n 128 -N 32 --smpiargs "-gpu --disable_gdr" mpibind ./ij -P 8 4 4 -n 50 50 50 -pmis -Pmx 8 -keepT 1 -rlx 18

Outline

- Overview of the MVAPICH2 Project
- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- What's new with MVAPICH2-GDR
- High-Performance Deep Learning (HiDL) with MVAPICH2-GDR
  - Benefits of CUDA-Aware MPI with TensorFlow
  - Optimized Collectives for Deep Learning
  - Out-of-core DNN Training
- Conclusions

Data Parallel Training with TensorFlow (TF)

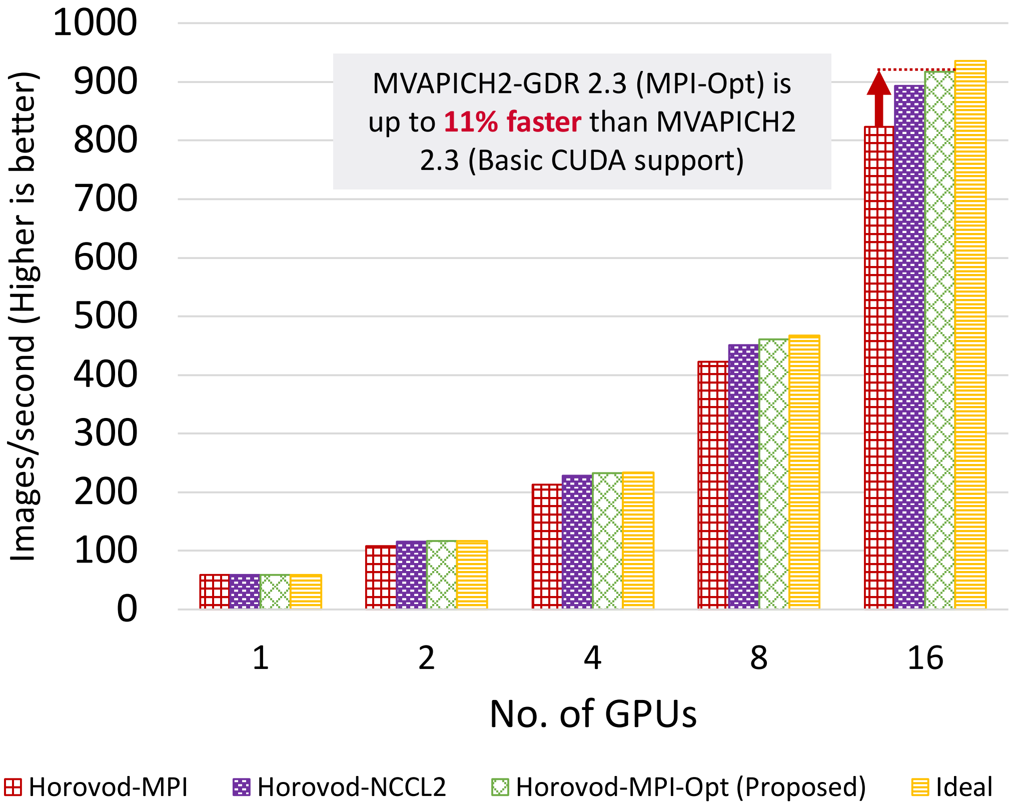
- Need to understand several options currently available
- gRPC (official support)
  - Open-source – can be enhanced by others
  - Accelerated gRPC (add RDMA to gRPC)
- gRPC+X
  - Use gRPC for bootstrap and rendezvous
  - ***Actual communication is in "X"***
  - X→ MPI, Verbs, GPUDirect RDMA (GDR), etc.
- No-gRPC
  - Baidu – the first one to use MPI Collectives for TF
  - Horovod – Use NCCL, or MPI, or any other future library (e.g. IBM DDL support recently added)



A. A. Awan, J. Bedorf, C.-H. Chu, H. Subramoni and D. K. Panda, "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs, and Performance Evaluation", CCGrid '19. https://arxiv.org/abs/1810.11112

Exploiting CUDA-Aware MPI for TensorFlow (Horovod)

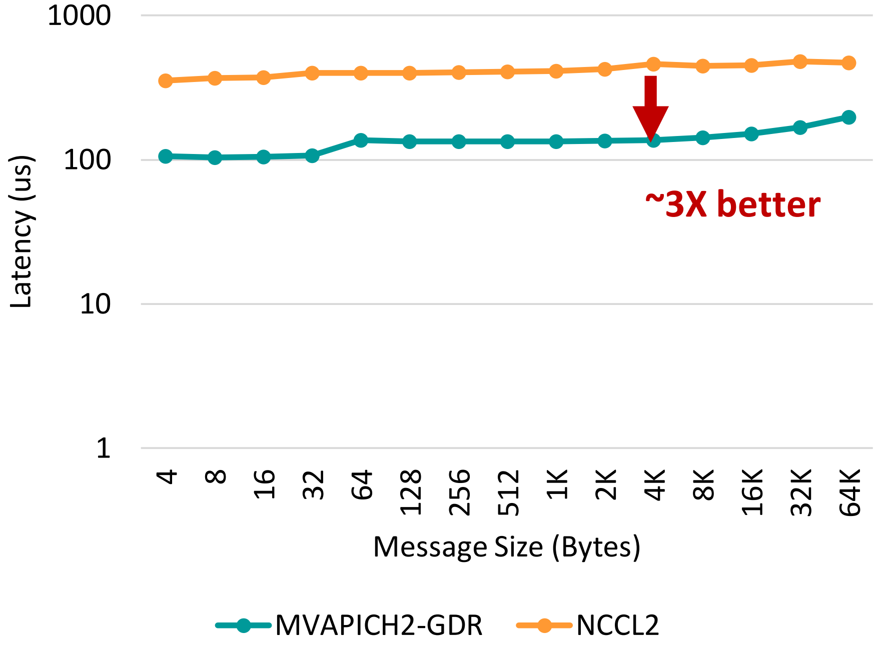
- MVAPICH2-GDR offers excellent performance via advanced designs for MPI_Allreduce.
- Up to **11% better** performance on the RI2 cluster (16 GPUs)
- Near-ideal – **98% scaling efficiency**

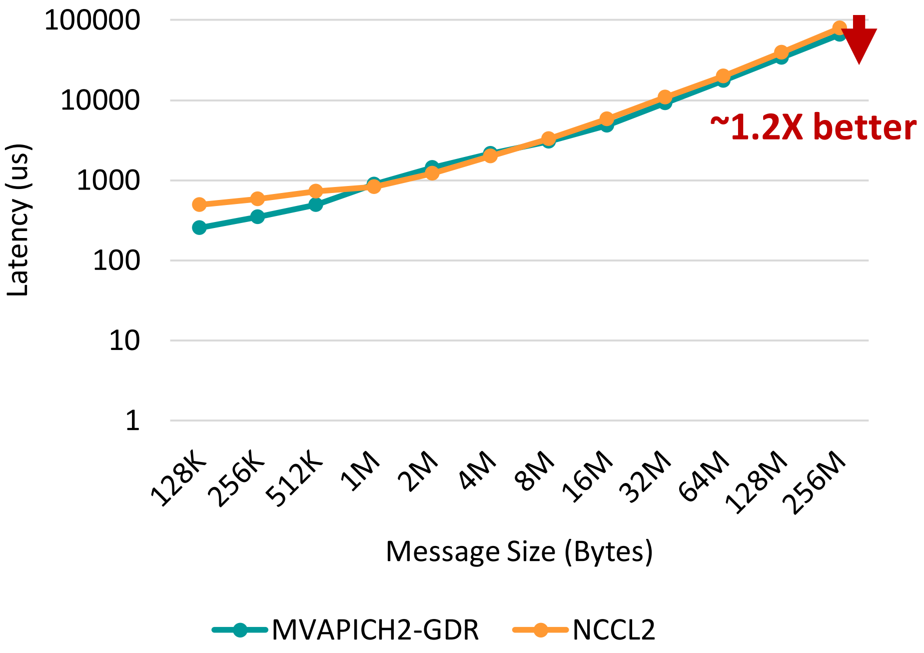


A. A. Awan et al., "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs, and Performance Evaluation", CCGrid '19, https://arxiv.org/abs/1810.11112

MVAPICH2-GDR vs. NCCL2: Allreduce Operation

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

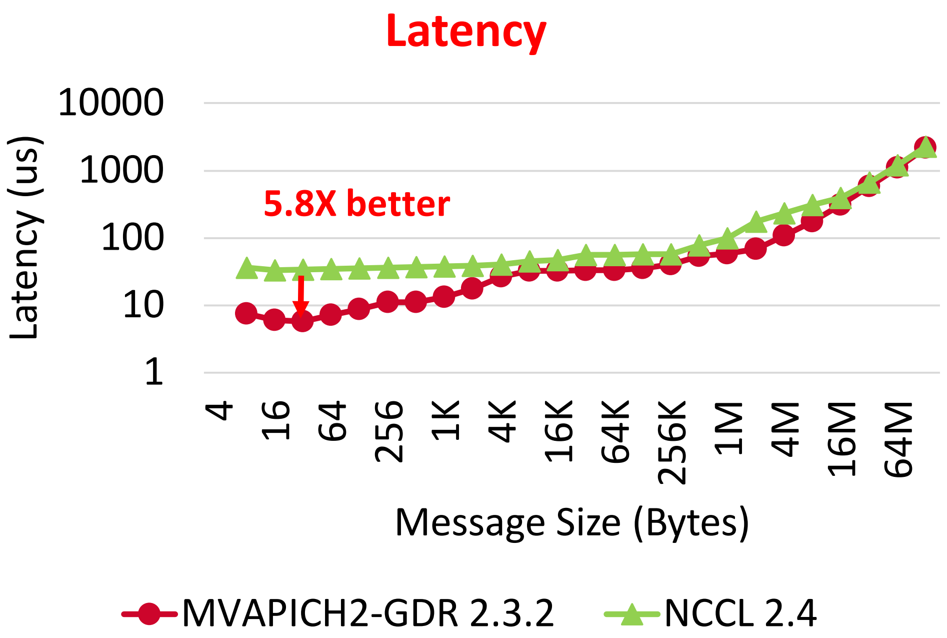


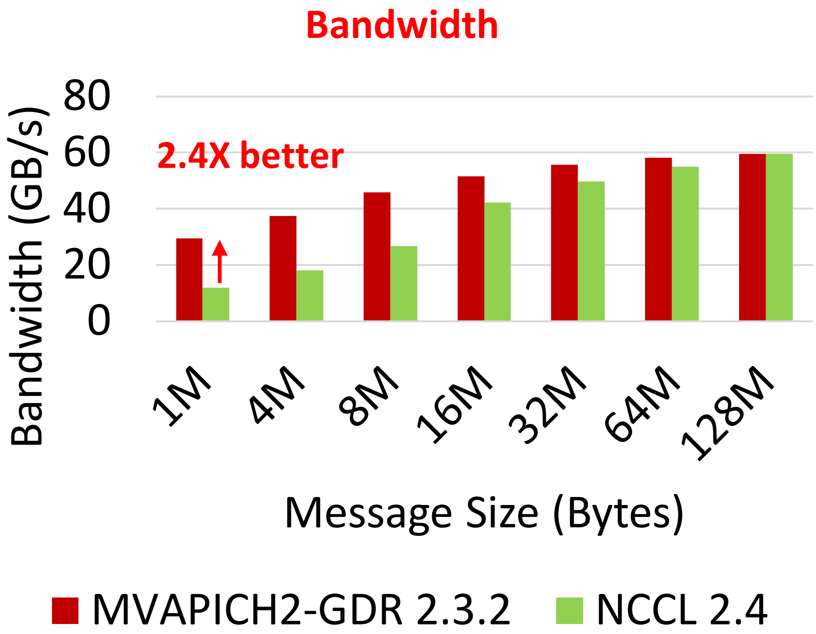


Platform: Intel Xeon (Broadwell) nodes equipped with a dual-socket CPU, 1 K-80 GPUs, and EDR InfiniBand Inter-connect

MVAPICH2-GDR vs. NCCL2: Allreduce Optimization (DGX-2)

- **Optimized designs in upcoming MVAPICH2-GDR offer better performance for most cases**
- **MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) on a DGX-2 machine**

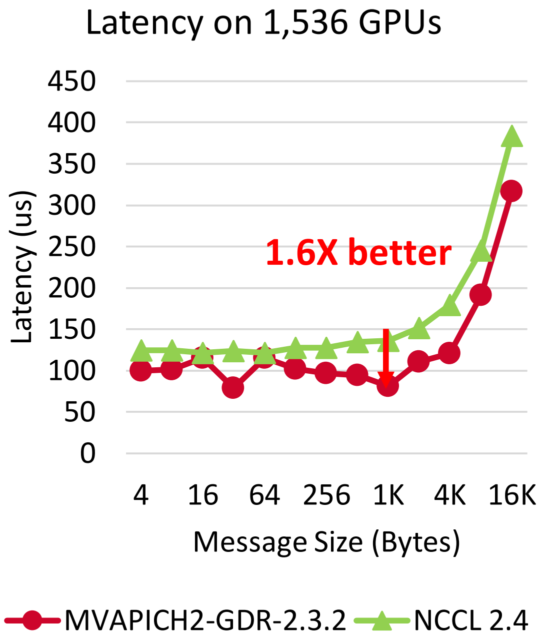


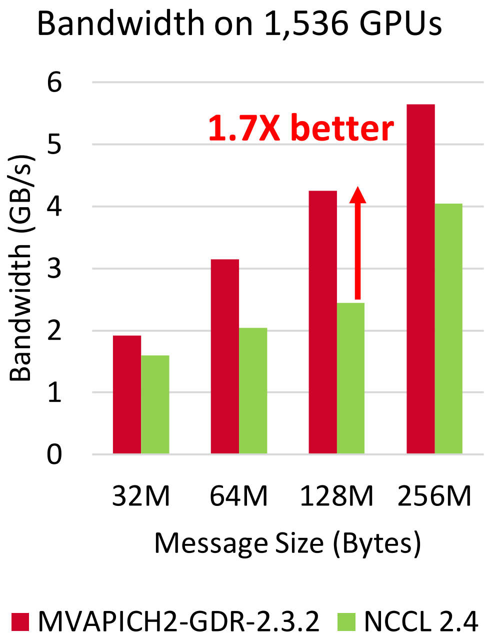


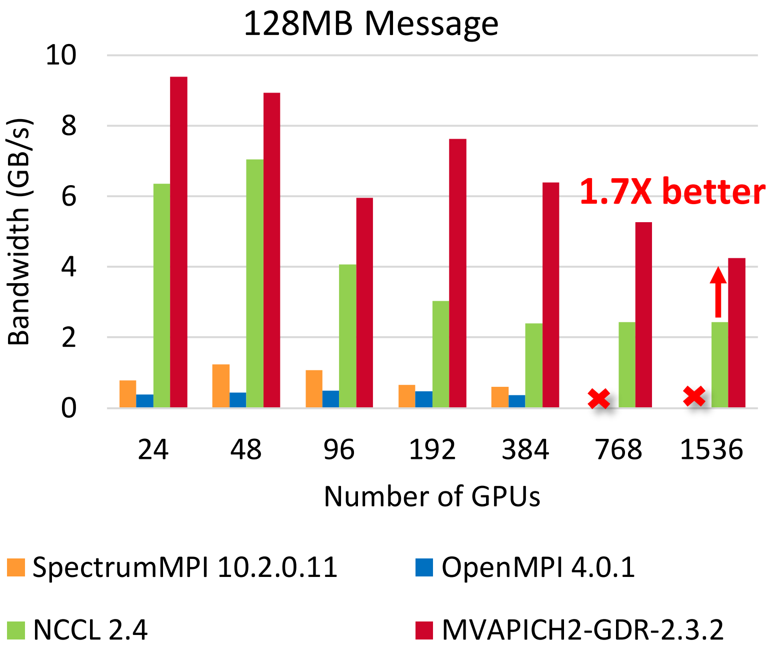
Platform: Nvidia DGX-2 system @ PSC (16 Nvidia Volta GPUs connected with NVSwitch), CUDA 9.2

MVAPICH2-GDR: MPI_Allreduce (Device Buffers) on Summit

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



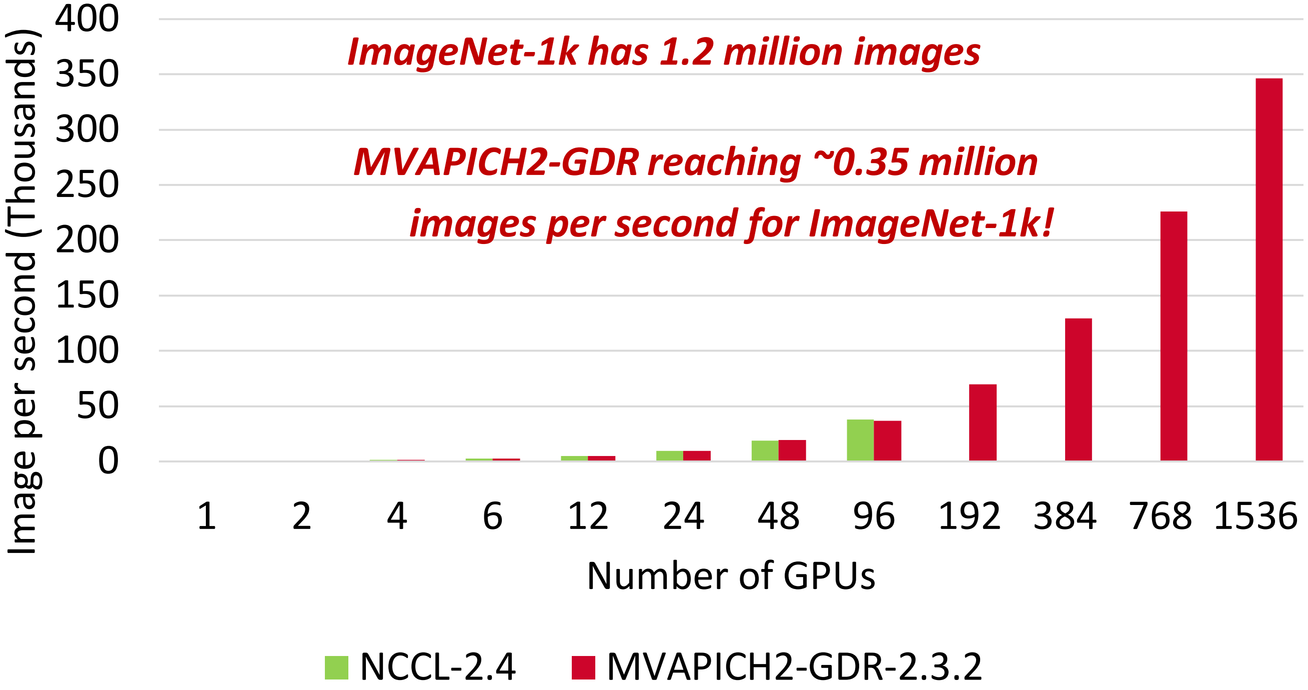




Platform: Dual-socket IBM POWER9 CPU, 6 NVIDIA Volta V100 GPUs, and 2-port InfiniBand EDR Interconnect

Distributed Training with TensorFlow and MVAPICH2-GDR on Summit

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

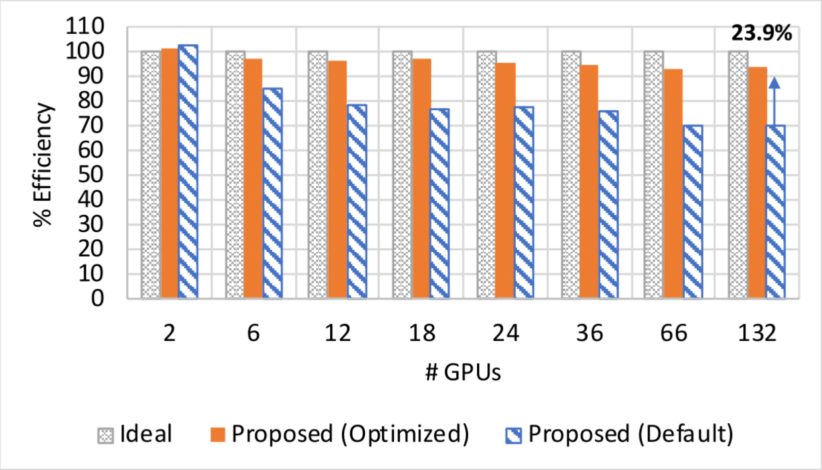


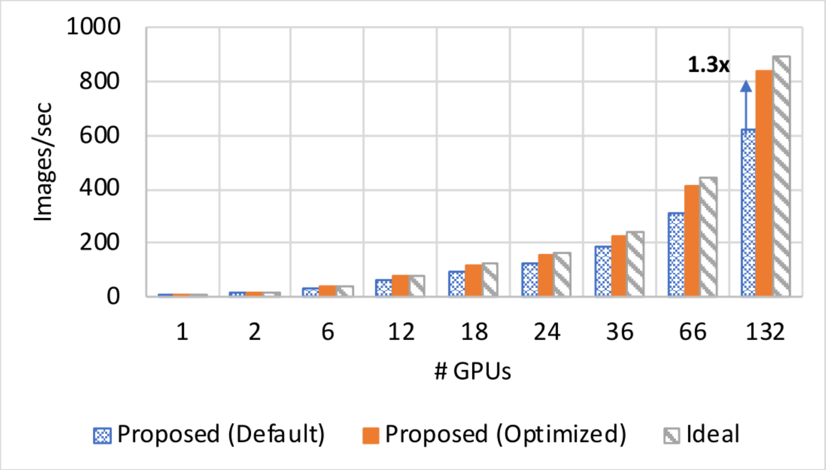
*We observed errors for NCCL2 beyond 96 GPUs

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

New Benchmark for Image Segmentation on Summit

- Near-linear scaling may be achieved by **tuning Horovod/MPI**
  - Optimizing MPI/Horovod towards large message sizes for high-resolution images
- Develop a generic Image Segmentation benchmark
- Tuned DeepLabV3+ model using the benchmark and Horovod, up to **1.3X** better than default





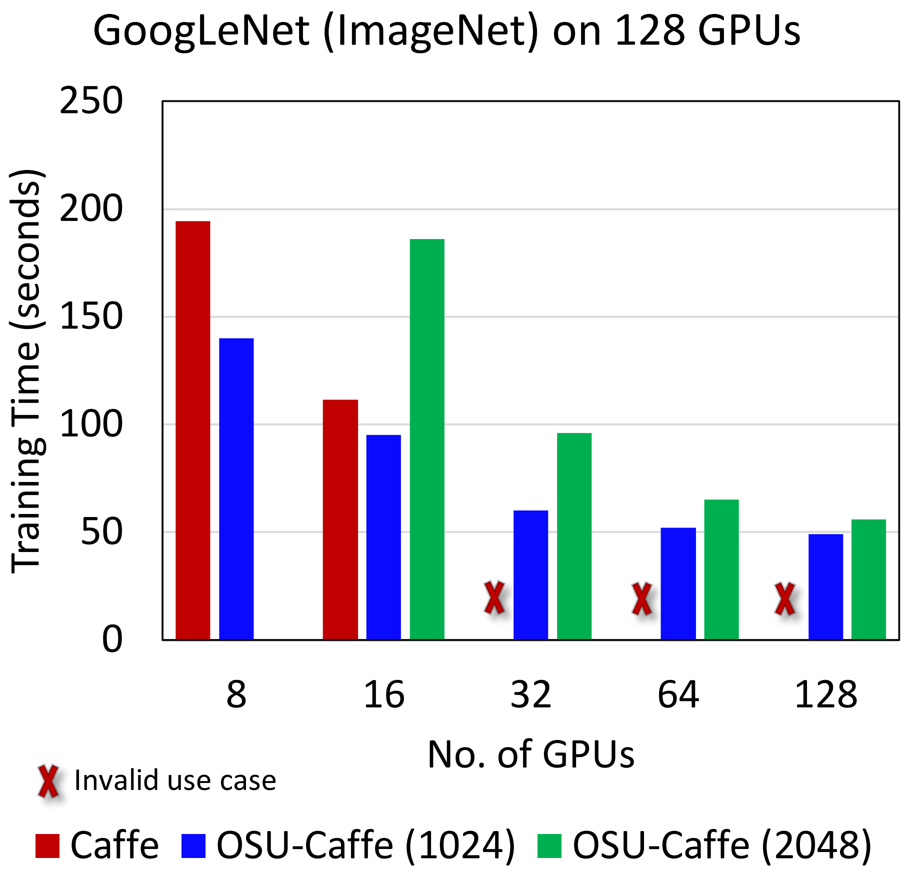
*Anthony et al., "Scaling Semantic Image Segmentation using Tensorflow and MVAPICH2-GDR on HPC Systems" (Submission under review)

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

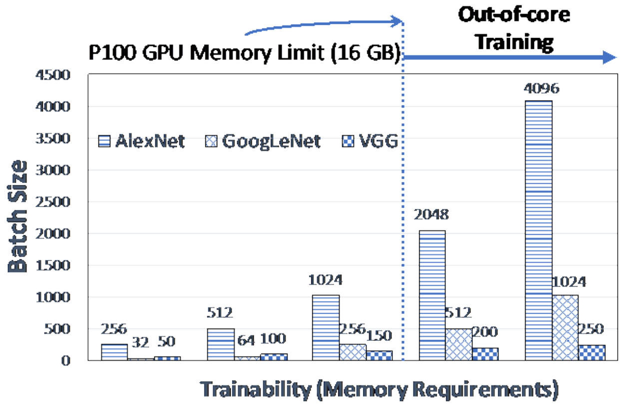
OSU-Caffe publicly available from

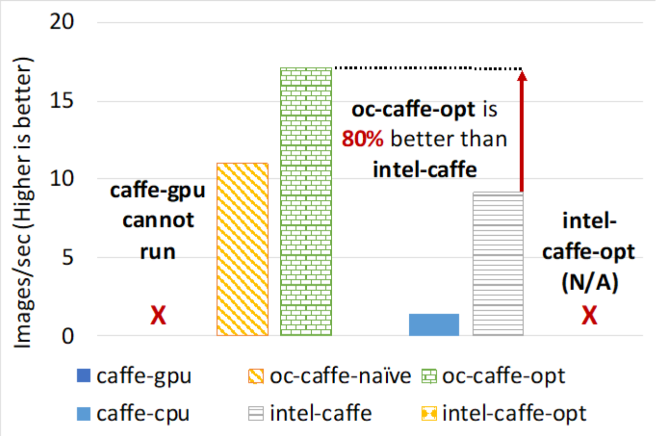
http://hidl.cse.ohio-state.edu/



Scalability and Large (Out-of-core) Models?

- Large DNNs cannot be trained on GPUs due to memory limitation!
  - ResNet-50 for Image Recognition but current frameworks can only go up to a small batch size of 45
  - Next generation models: Neural Machine Translation (NMT)
    - Ridiculously large (billions of parameters),
    - **Will require even more memory!**
  - Can we exploit new software features in CUDA 8/9 and hardware mechanisms in Pascal/Volta GPUs?
- General intuition is that managed allocations “will be” slow!
  - 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-Opt: up to **80% better** than Intel-optimized CPU Caffe for ResNet-50 training on the Volta V100 GPU with CUDA9 and CUDNN7

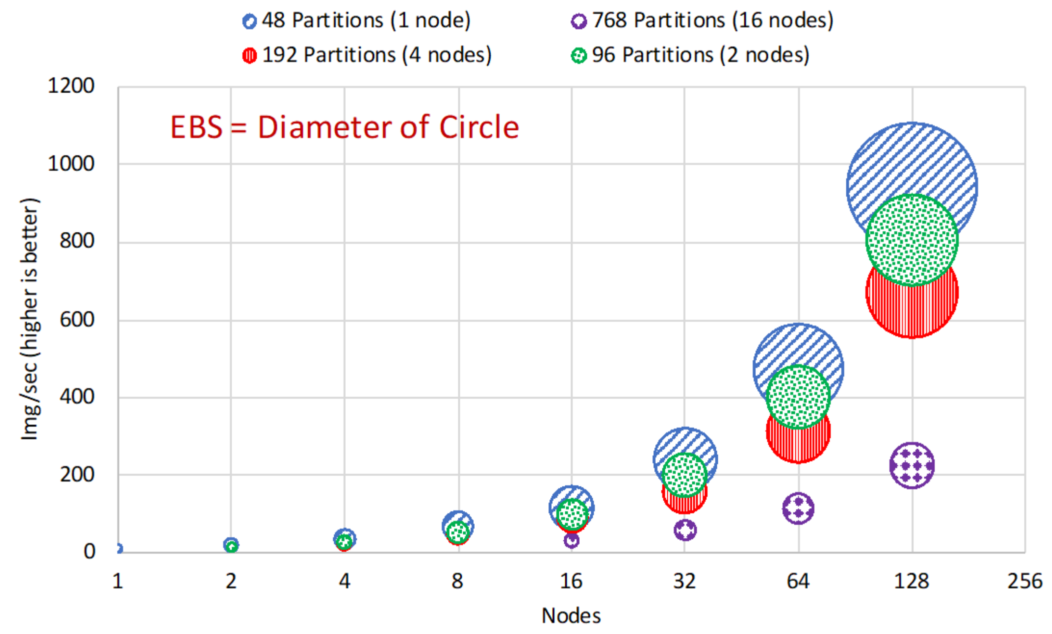




A. A. Awan, C.-H. Chu, H. Subramoni, X. Lu, and D. K. Panda, OC-DNN: Exploiting Advanced Unified Memory Capabilities in CUDA 9 and Volta GPUs for Out-of-Core DNN Training, HiPC ’18

HyPar-Flow (HF): Hybrid Parallelism for TensorFlow

- CPU based results
  - AMD EPYC
  - Intel Xeon
- Excellent speedups for
  - VGG-19
  - ResNet-110
  - ResNet-1000 (1k layers)
- Able to train "future" models
  - E.g. ResNet-5000 (a synthetic 5000-layer model we benchmarked)



110x speedup on 128 Intel Xeon Skylake nodes (TACC Stampede2 Cluster)

*Awan et al., "HyPar-Flow: Exploiting MPI and Keras for Hybrid Parallel Training of TensorFlow models", arXiv '19. https://arxiv.org/pdf/1911.05146.pdf

Outline

- Overview of the MVAPICH2 Project
- MVAPICH2-GPU with GPUDirect-RDMA (GDR)
- What's new with MVAPICH2-GDR
- High-Performance Deep Learning (HiDL) with MVAPICH2-GDR
- Conclusions

Conclusions

- MVAPICH2-GDR Library provides optimized MPI communication on InfiniBand and RoCE clusters with GPUs
- Supports both X86 and OpenPower with NVLink
- Takes advantage of CUDA features like IPC and GPUDirect RDMA families
- Allows flexible solutions for streaming applications with GPUs
- Provides optimized solutions (scale-up and scale-out) for High-Performance Deep Learning

Commercial Support for MVAPICH2, HiBD, and HiDL Libraries

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

X-ScaleSolutions

Multiple Events at SC ‘19

- Presentations at OSU and X-Scale Booth (#2094)
  - Members of the MVAPICH, HiBD and HiDL members
  - External speakers
- Presentations at SC main program (Tutorials, Workshops, BoFs, Posters, and Doctoral Showcase)
- Presentation at many other booths (Mellanox, Intel, Microsoft, and AWS) and satellite events
- Complete details available at

  http://mvapich.cse.ohio-state.edu/conference/752/talks/

Funding Acknowledgments

Funding Support by

Equipment Support by

Personnel Acknowledgments

Current Students (Graduate)

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

Current Research Scientist

- H. Subramoni

Current Students (Undergraduate)

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

Current Post-doc

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

Current Research Specialist

- J. Smith

Past Students

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

Past Research Scientist

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

Past Programmers

- D. Bureddy
- J. Perkins

Past Research Specialist

- M. Arnold

Past Post-Docs

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

Thank You!

panda@cse.ohio-state.edu

Follow us on

https://twitter.com/mvapich

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/