MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

HiDL
High-Performance Deep Learning

HiBD
High-Performance Big Data

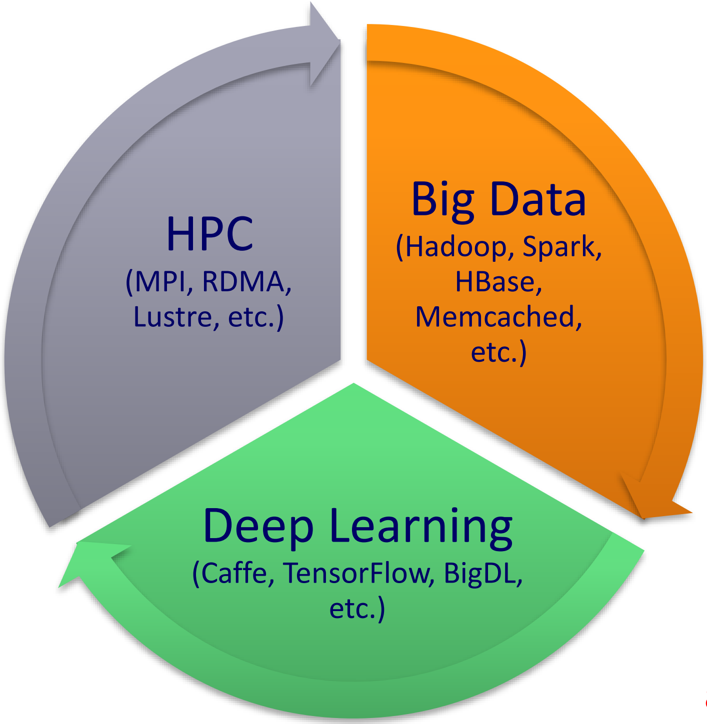
Overview of High-Performance Deep Learning (HiDL) Project

Talk at OSU Booth (SC ’18)

by

Dhabaleswar K. (DK) Panda
The Ohio State University
E-mail: panda@cse.ohio-state.edu
http://www.cse.ohio-state.edu/~panda

Increasing Usage of HPC, Big Data and Deep Learning



Convergence of HPC, Big Data, and Deep Learning!

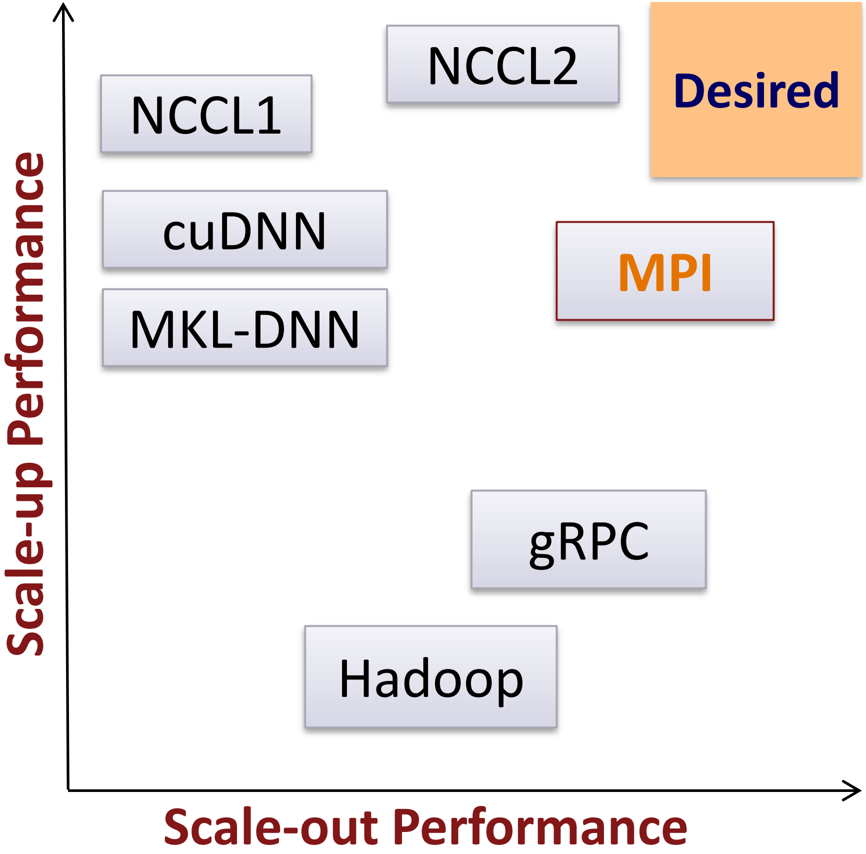
Increasing Need to Run these applications on the Cloud!!

Trends in Training for Deep Learning frameworks

- Early (2014) frameworks used a single fast GPU
  - As DNNs became larger, faster and better GPUs became available
  - At the same time, parallel (multi-GPU) training gained traction as well
- Today
  - Parallel training on multiple GPUs is being supported by most frameworks
  - Distributed (multiple nodes) training is still upcoming
    - A lot of fragmentation in the efforts (MPI, Big-Data, NCCL, Gloo, etc.)

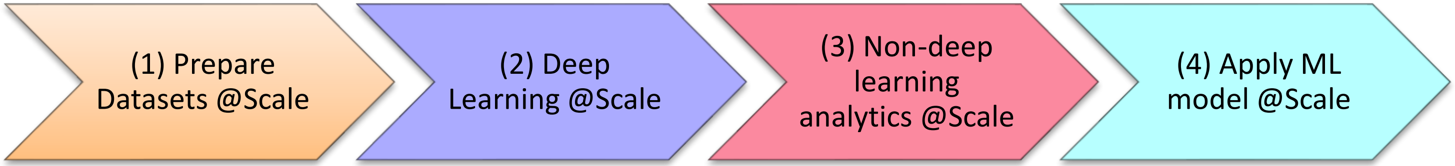
Scale-up and Scale-out

- **Scale-up**: Intra-node Communication
  - Many improvements like:
    - NVIDIA cuDNN, cuBLAS, NCCL, etc.
    - CUDA 9 Co-operative Groups
- **Scale-out**: Inter-node Communication
  - DL Frameworks – most are optimized for single-node only
  - Distributed (Parallel) Training is an emerging trend
    - **OSU-Caffe – MPI-based**
    - Microsoft CNTK – MPI/NCCL2
    - Google TensorFlow – gRPC-based/MPI/NCCL2
    - Facebook Caffe2 – Hybrid (NCCL2/Gloo/MPI)



Deep Learning over Big Data (DLoBD)

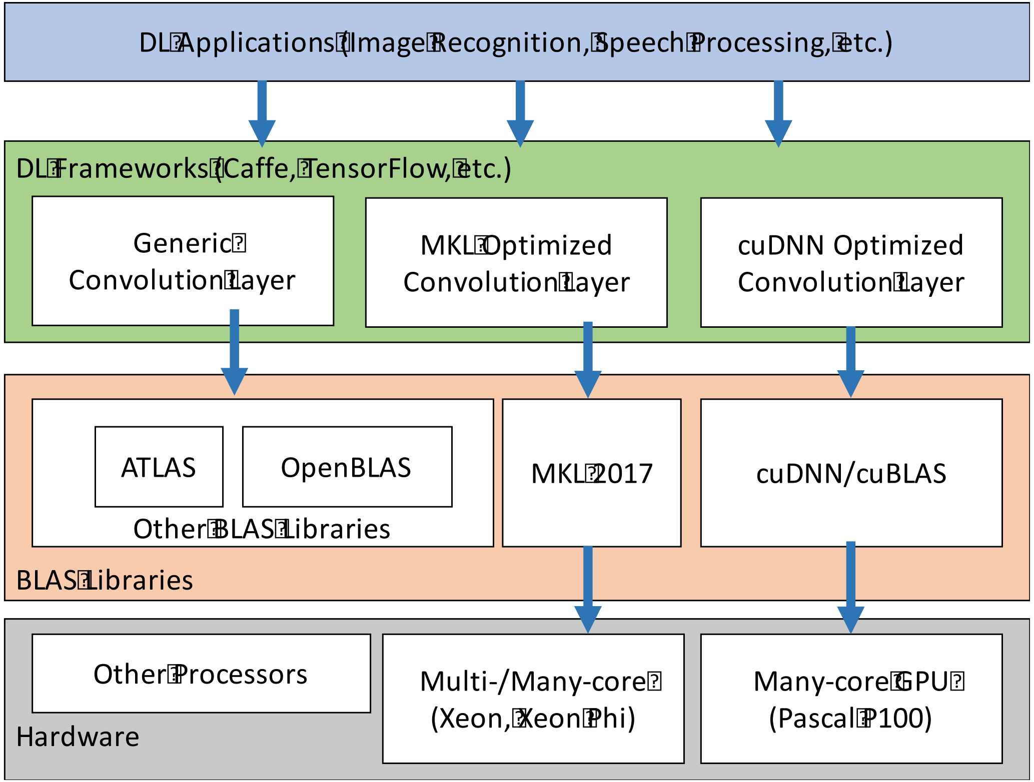
- Deep Learning over Big Data (**DLoBD**) is one of the most efficient analyzing paradigms
- More and more deep learning tools or libraries (e.g., Caffe, TensorFlow) start running over big data stacks, such as Apache Hadoop and Spark
- **Benefits** of the DLoBD approach
  - Easily build a powerful data analytics **pipeline**
    - E.g., Flickr DL/ML Pipeline, "*How Deep Learning Powers Flickr*", http://bit.ly/1KIDfof



  - Better data **locality**
  - Efficient resource sharing and **cost effective**

Holistic Evaluation is Important!!

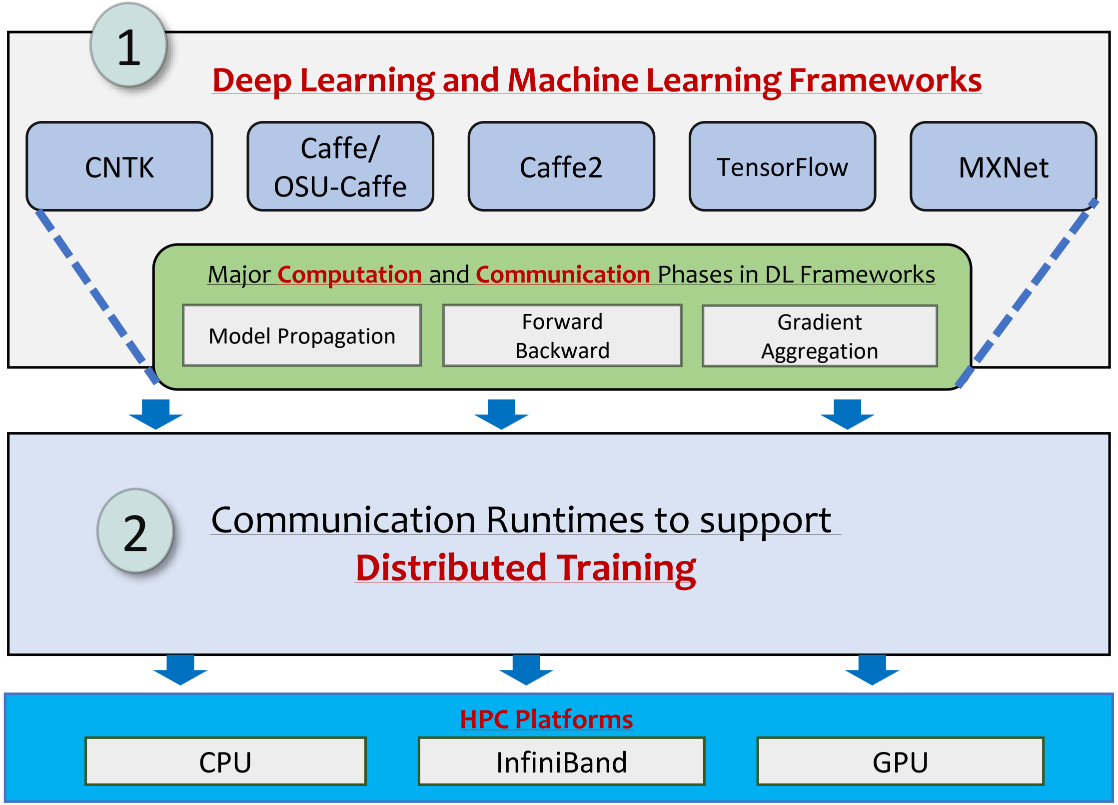
- My framework is faster than your framework!
- This needs to be understood in a holistic way.
- Performance depends on the entire execution environment (the full stack)
- Isolated view of performance is not helpful



A. A. Awan, H. Subramoni, and Dhabaleswar K. Panda. "An In-depth Performance Characterization of CPU- and GPU-based DNN Training on Modern Architectures", In Proceedings of the Machine Learning on HPC Environments (MLHPC'17). ACM, New York, NY, USA, Article 8.

Research Challenges to Exploit HPC Technologies

1. What are the fundamental issues in designing **DL frameworks**?
   - Memory Requirements
   - **Computation** Requirements
   - **Communication** Overhead
2. Why do we need to support **distributed training**?
   - To overcome the limits of single-node training
   - To better utilize hundreds of existing HPC Clusters



Research Challenges to Exploit HPC Technologies (Cont'd)

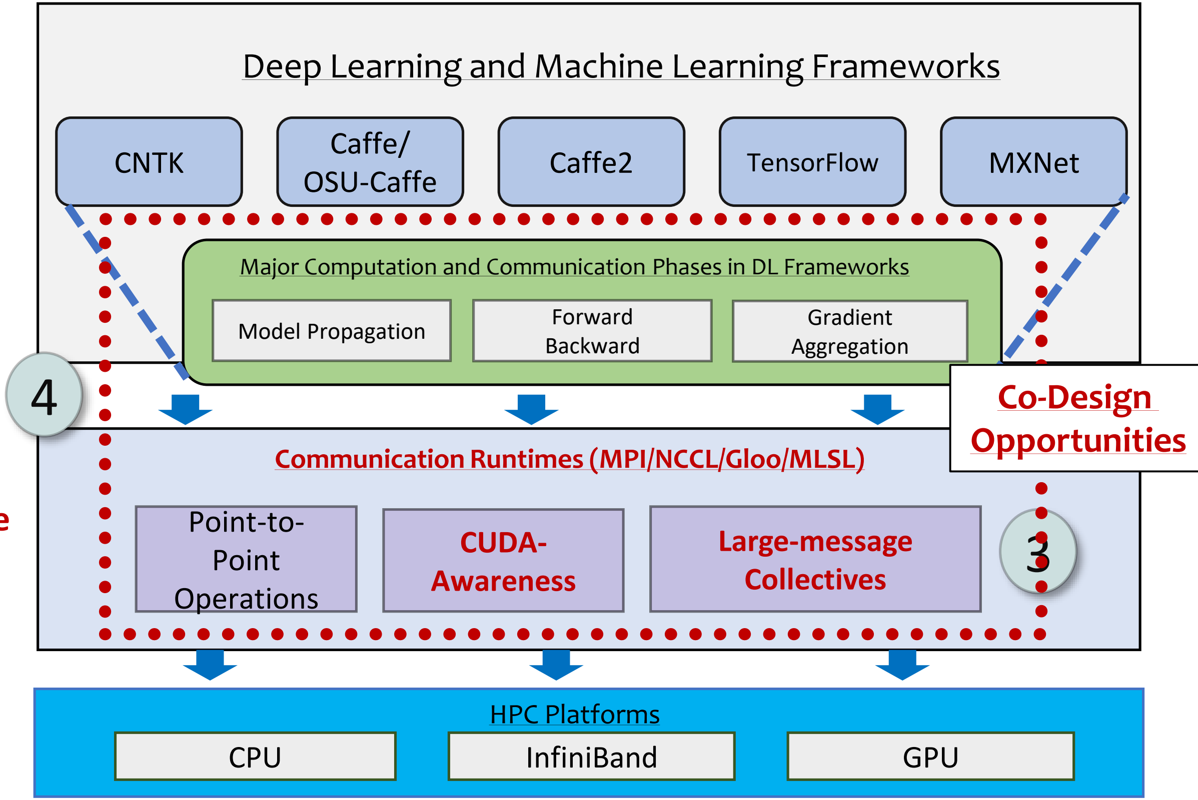
3. What are the **new design challenges** brought forward by DL frameworks for Communication runtimes?

- Large Message **Collective Communication** and Reductions
- GPU Buffers (**CUDA-Awareness**)

4. Can a **Co-design** approach help in achieving Scale-up and Scale-out efficiently?

- **Co-Design** the support at **Runtime level** and Exploit it at the **DL Framework level**
- What performance benefits can be observed?
- What needs to be fixed at the **communication runtime** layer?

5.

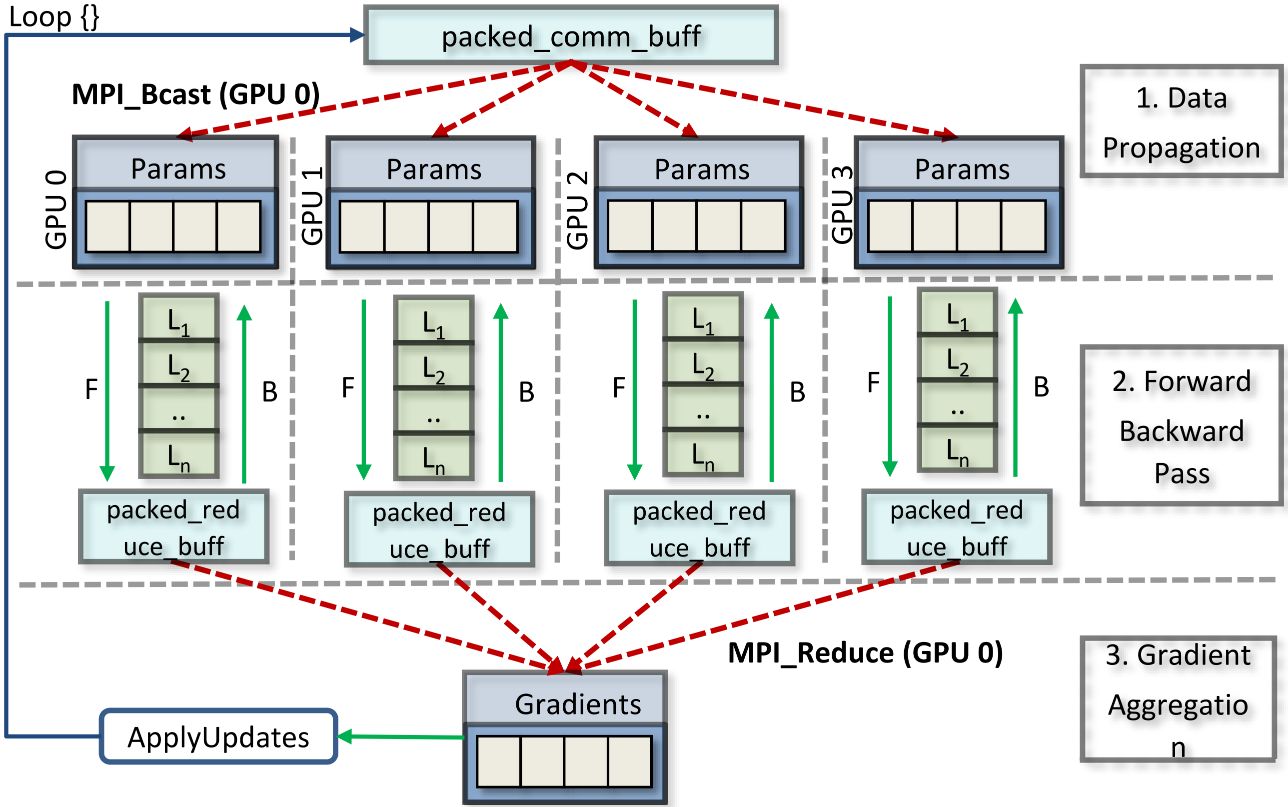


Multiple Approaches taken up by OSU

- MPI-driven Deep Learning
- Co-designing Deep Learning Stacks with High-Performance MPI
- Accelerating TensorFlow on HPC Systems
- Accelerating Big Data Stacks
- Efficient Deep Learning over Big Data

Data Parallel Deep Learning and MPI Collectives

- Major **MPI Collectives** involved in Designing distributed frameworks
- **MPI_Bcast** – required for DNN parameter exchange
- **MPI_Reduce** – needed for gradient accumulation from multiple solvers
- **MPI_Allreduce** – use just one Allreduce instead of Reduce and Broadcast



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)

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 2,950 organizations in 86 countries**
  - **More than 505,000 (> 0.5 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (Jul '18 ranking)
    - 2nd ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, China
    - 12th, 556,104 cores (Oakforest-PACS) in Japan
    - 15th, 367,024 cores (Stampede2) at TACC
    - 24th, 241,108-core (Pleiades) at NASA and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, and OpenHPC)
  - **http://mvapich.cse.ohio-state.edu**

Partner in the upcoming TACC Frontera System

- Empowering Top500 systems for over a decade

Architecture of MVAPICH2 Software Family

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)

Transport Protocols

- RC
- XRC
- 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

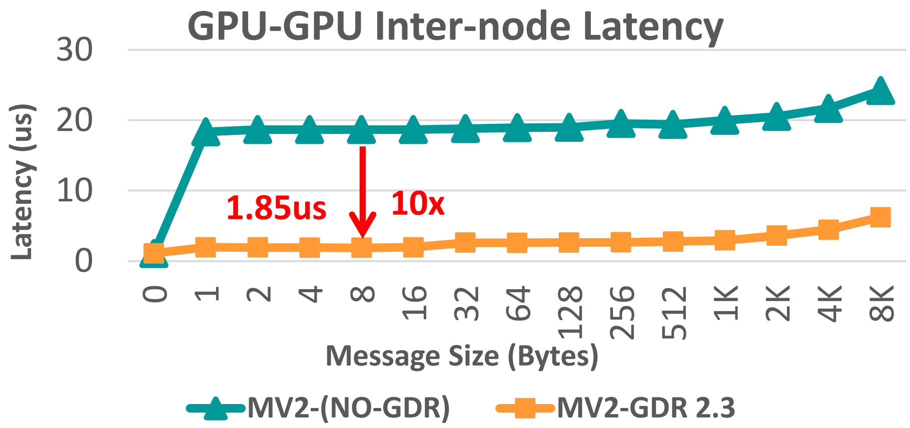
- Shared Memory
- CMA
- IVSHMEM
- XPMEM*

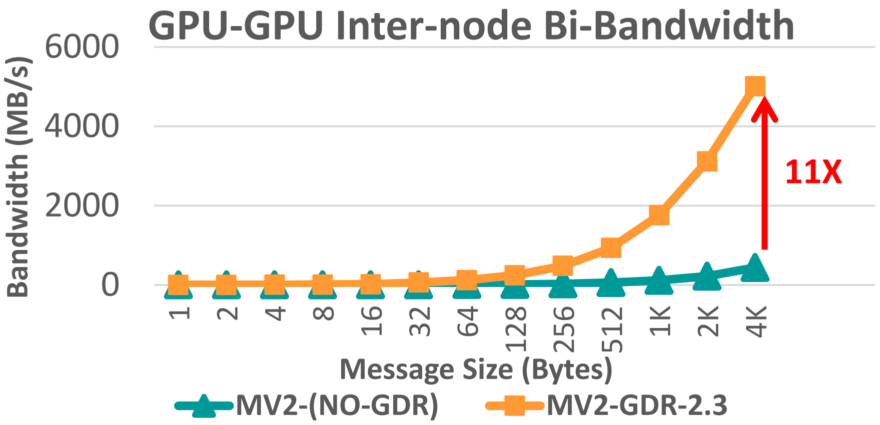
Modern Features

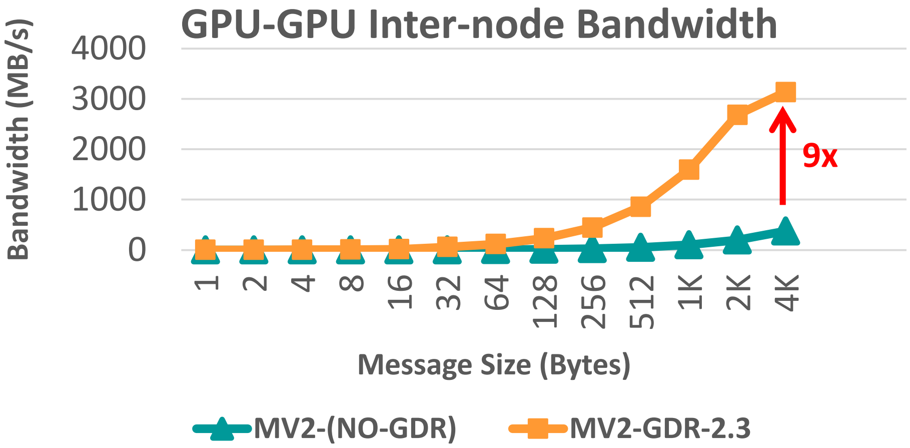
- MCDRAM*
- NVLink*
- CAPI*

* Upcoming

Optimized MVAPICH2-GDR Design



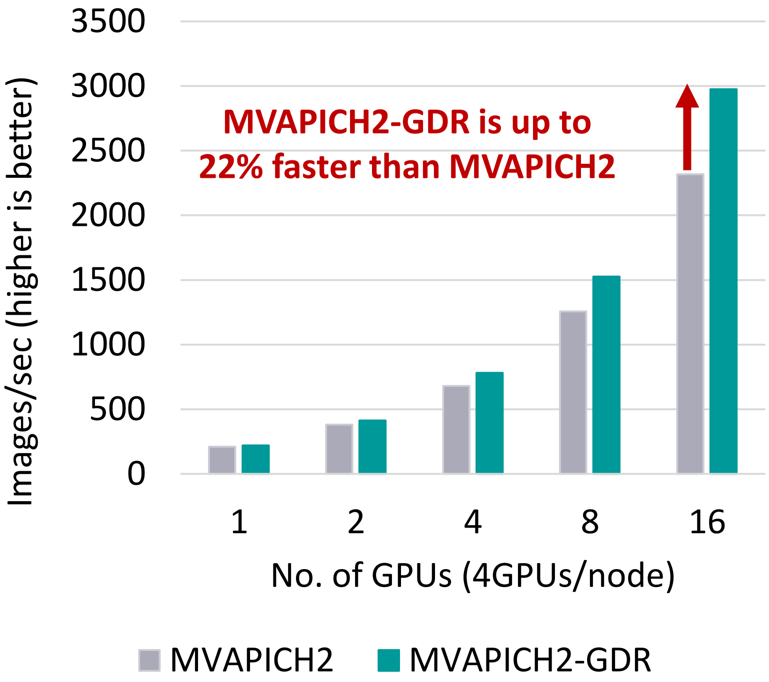




MVAPICH2-GDR-2.3
Intel Haswell (E5-2687W @ 3.10 GHz) node - 20 cores
NVIDIA Volta V100 GPU
Mellanox Connect-X4 EDR HCA
CUDA 9.0
Mellanox OFED 4.0 with GPU-Direct-RDMA

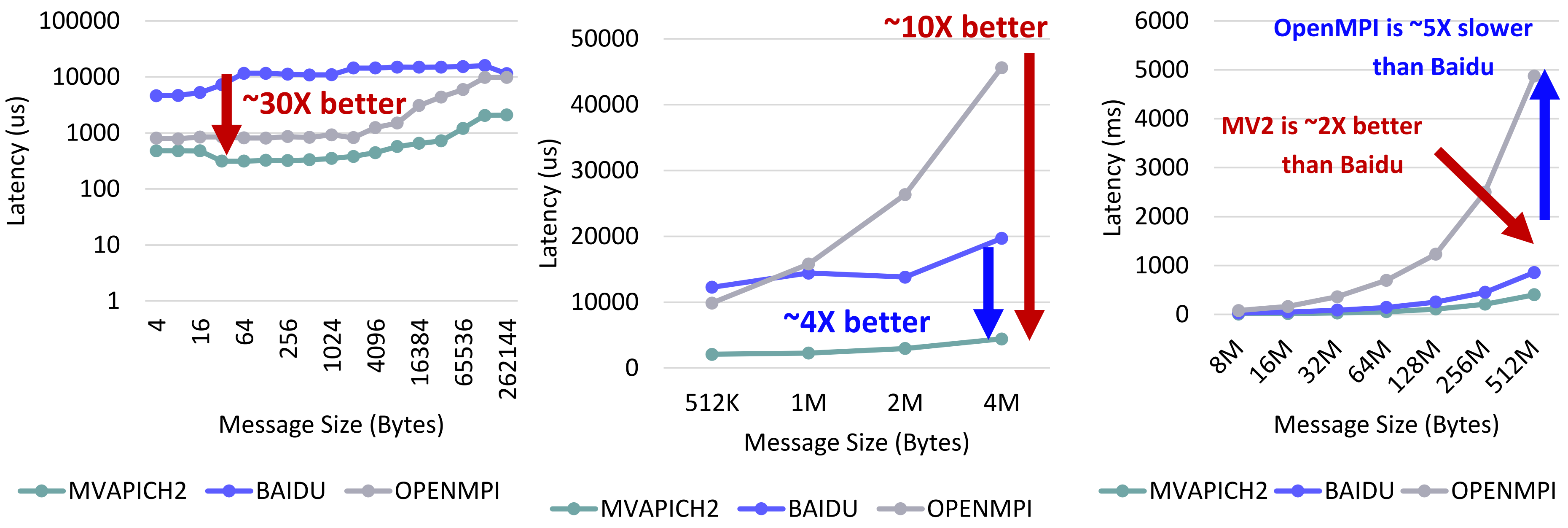
Exploiting CUDA-Aware MPI for TensorFlow (Horovod)

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



MVAPICH2-GDR: Allreduce Comparison with Baidu and OpenMPI

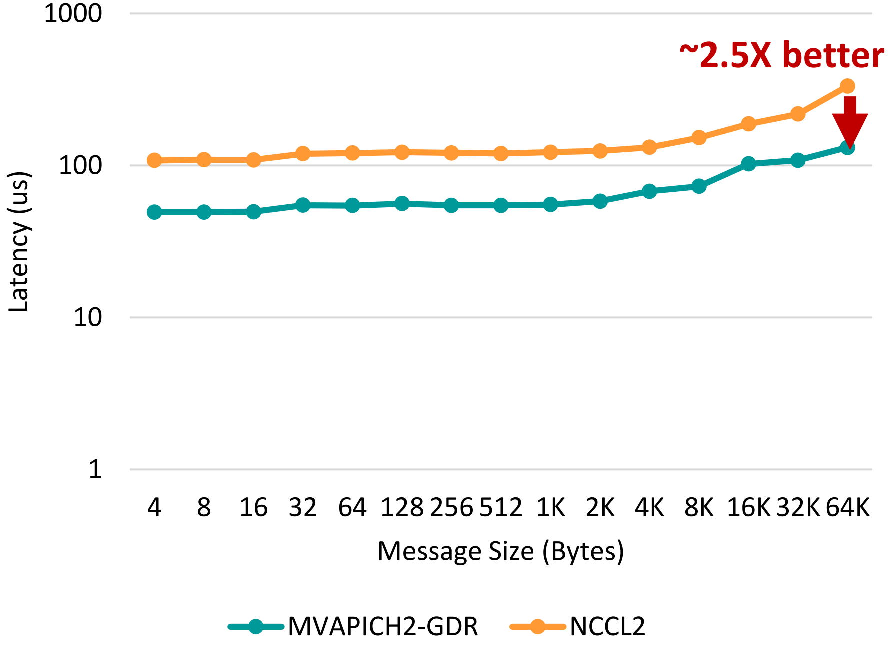
- 16 GPUs (4 nodes) MVAPICH2-GDR vs. Baidu-Allreduce and OpenMPI 3.0

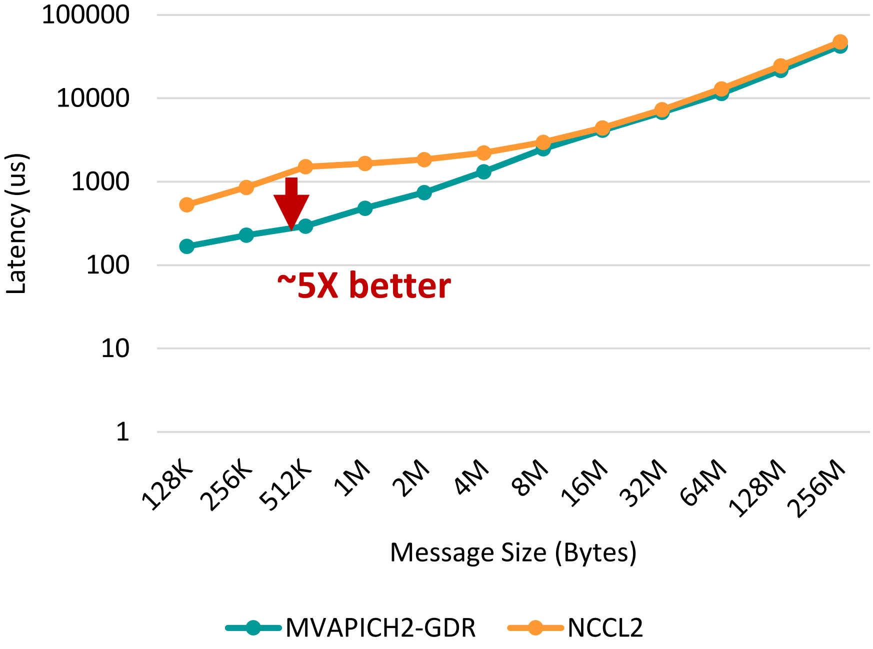


**Available since MVAPICH2-GDR 2.3a*

MVAPICH2-GDR vs. NCCL2 – 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**



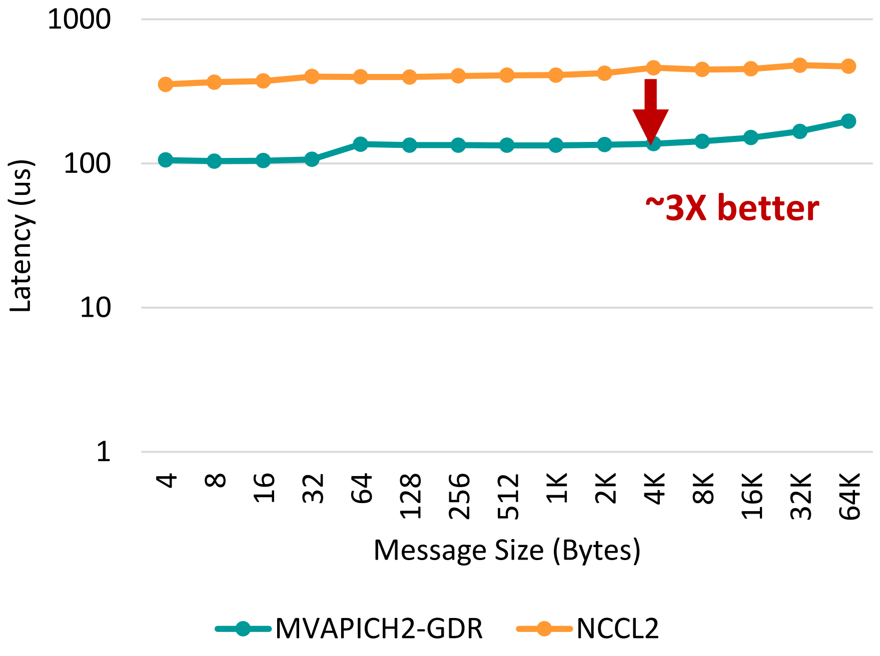


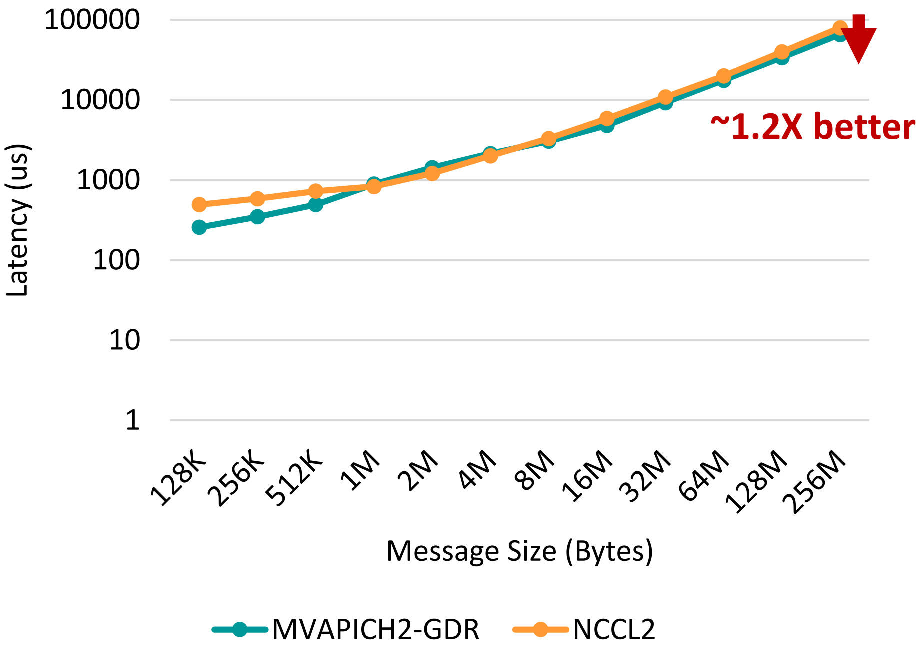
***Will be available with upcoming MVAPICH2-GDR 2.3b**

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 2.3rc1 offer better/comparable performance for most cases**
- **MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) on 16 GPUs**



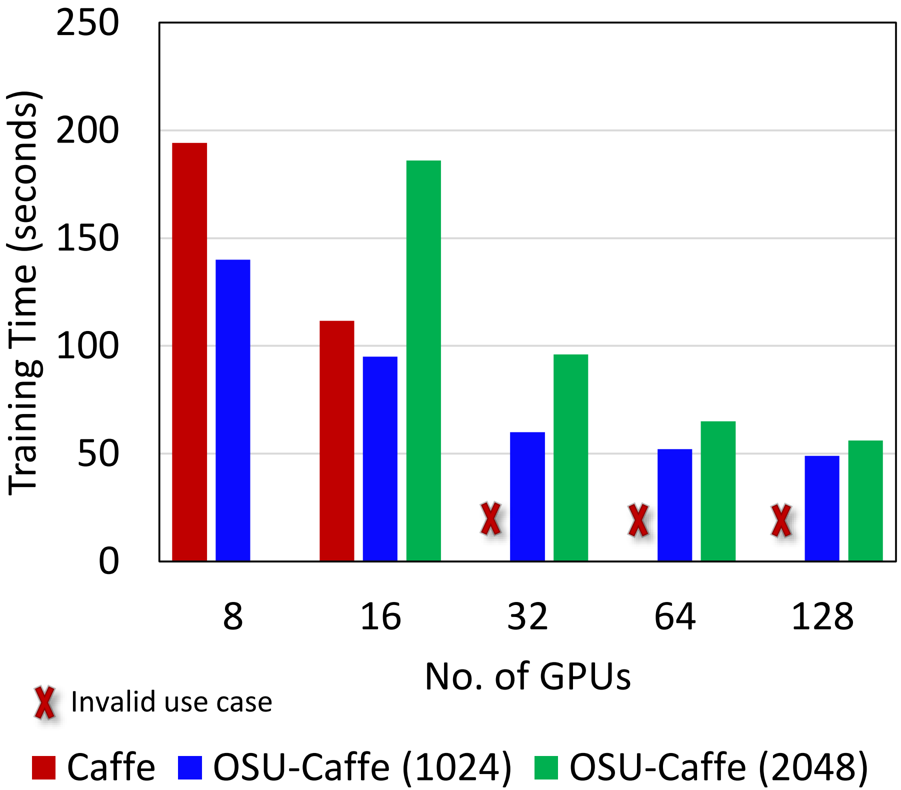


Platform: Intel Xeon (Broadwell) nodes equipped with a dual-socket CPU, 1 K-80 GPUs, and EDR InfiniBand Inter-connect

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

GoogLeNet (ImageNet) on 128 GPUs



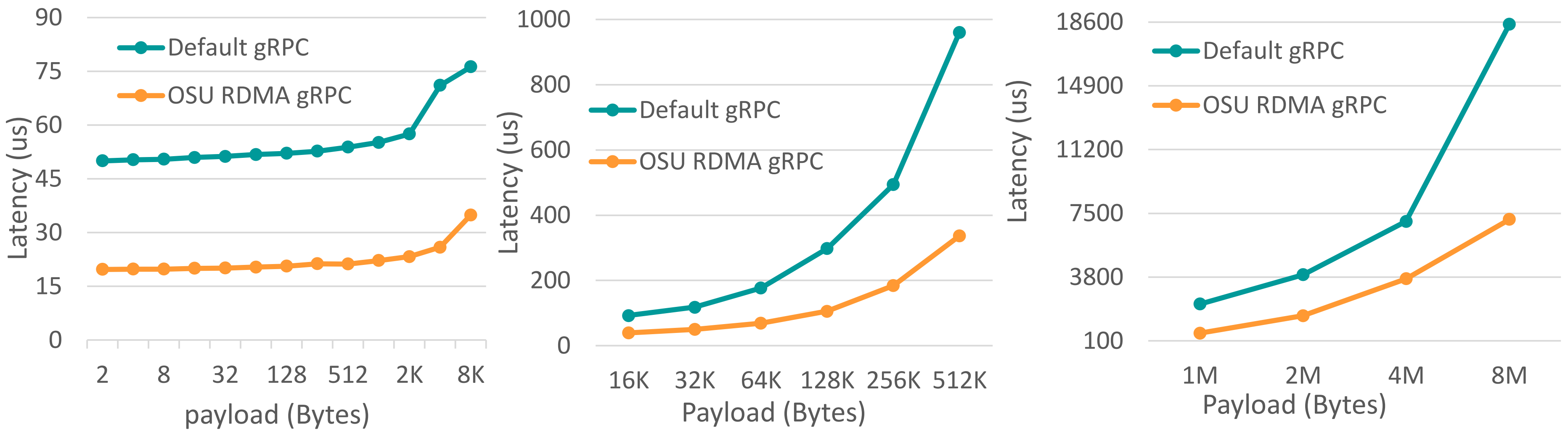
OSU-Caffe publicly available from

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

Multiple Approaches taken up by OSU

- MPI-driven Deep Learning
- Co-designing Deep Learning Stacks with High-Performance MPI
- Accelerating TensorFlow on HPC Systems
- Accelerating Big Data Stacks
- Efficient Deep Learning over Big Data

Performance Benefits for RDMA-gRPC with Micro-Benchmark

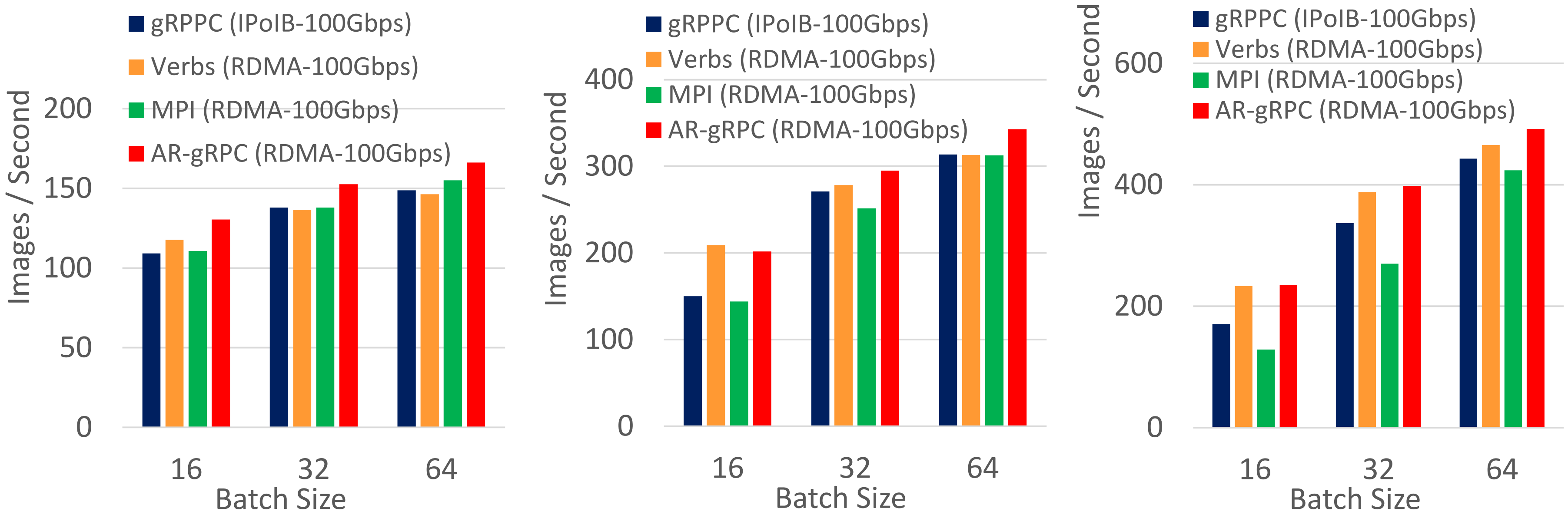


RDMA-gRPC RPC Latency

- **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, HiPC '18.

Performance Benefit for RDMA-TensorFlow (Inception3)



4 Nodes (8 GPUS)　**8 Nodes (16 GPUS)**　**12 Nodes (24 GPUS)**

- TensorFlow Inception3 performance evaluation on an IB EDR cluster
  - Up to 20% performance speedup over Default gRPC (IPoIB) for 8 GPUs
  - Up to 34% performance speedup over Default gRPC (IPoIB) for 16 GPUs
  - Up to 37% performance speedup over Default gRPC (IPoIB) for 24 GPUs

R. Biswas, X. Lu, and D. K. Panda, Accelerating TensorFlow with Adaptive RDMA-based gRPC. HiPC '18

Multiple Approaches taken up by OSU

- MPI-driven Deep Learning
- Co-designing Deep Learning Stacks with High-Performance MPI
- Accelerating TensorFlow on HPC Systems
- Accelerating Big Data Stacks
- Efficient Deep Learning over Big Data

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 Kafka
- 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: 290 organizations from 34 countries
- More than 28,200 downloads from the project site

Available for InfiniBand and RoCE

Also run on Ethernet

Available for x86 and OpenPOWER

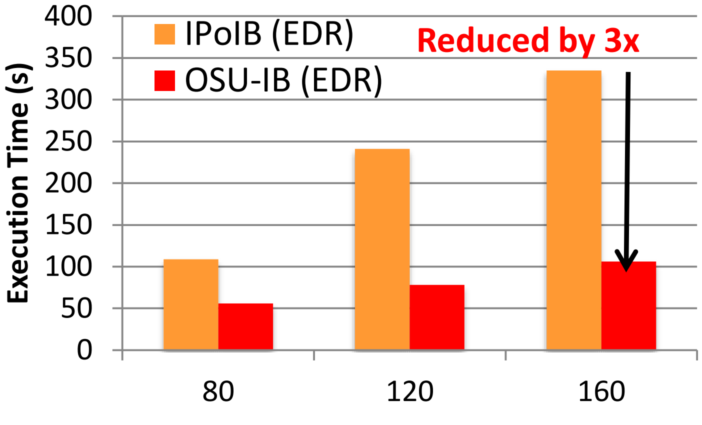
Support for Singularity and Docker

HiBD
High-Performance
Big Data

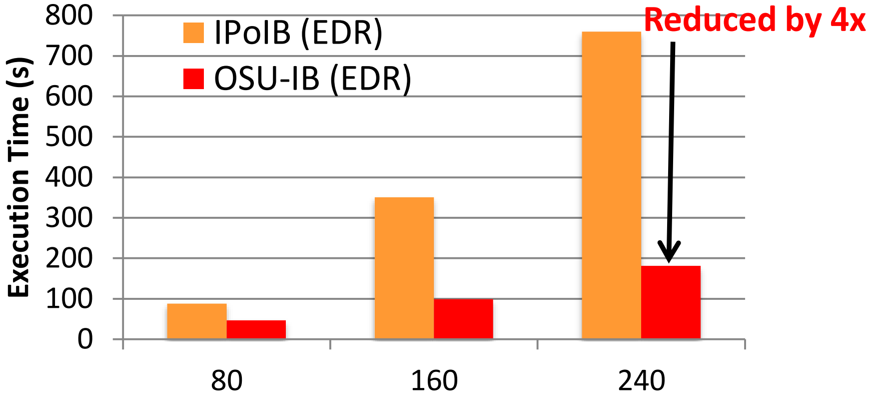
Network Based Computing Laboratory

The Ohio State University

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



RandomWriter

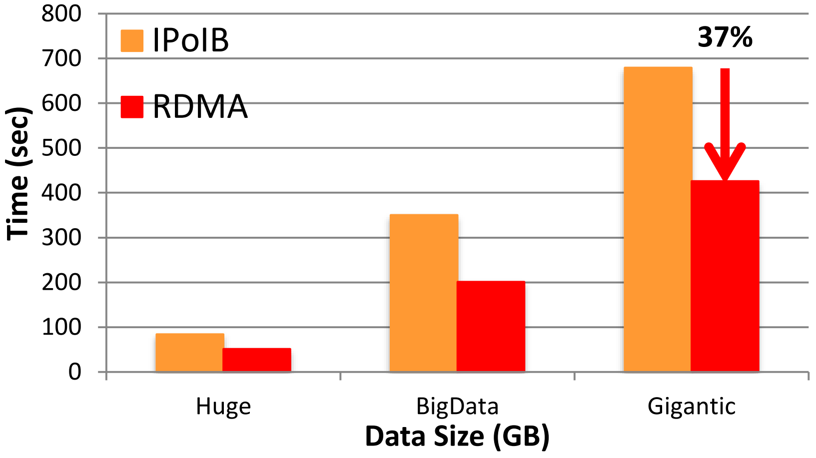


TeraGen

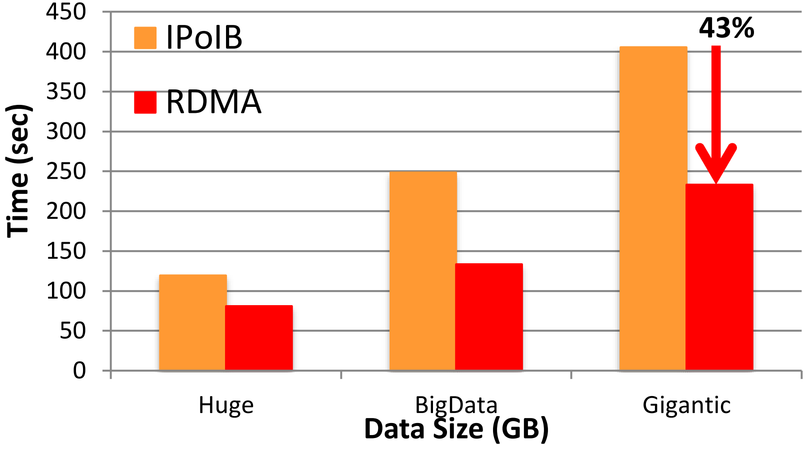
Cluster with 8 Nodes with a total of 64 maps

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

Performance Evaluation on SDSC Comet – HiBench PageRank



32 Worker Nodes, 768 cores, PageRank Total Time

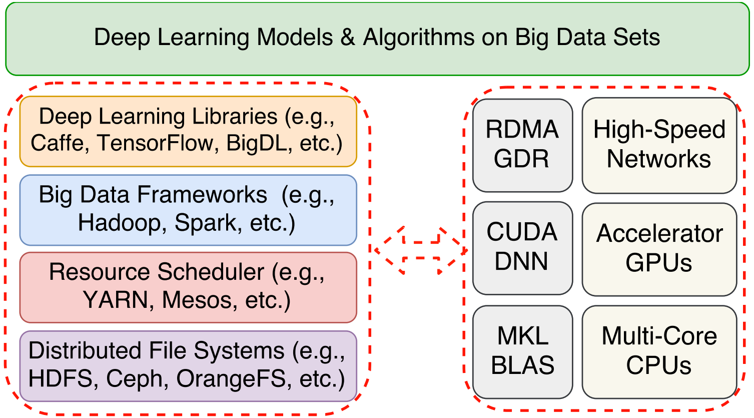


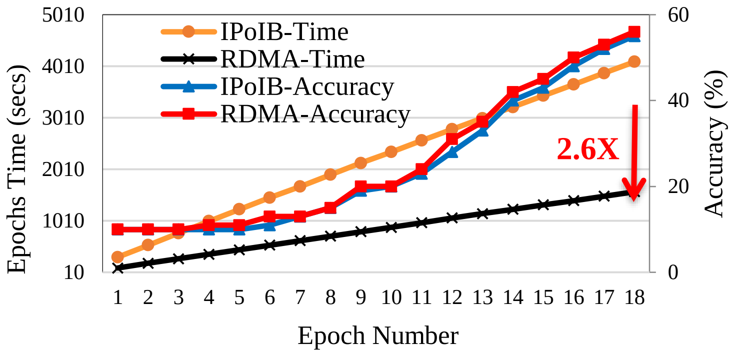
64 Worker Nodes, 1536 cores, PageRank Total Time

- InfiniBand FDR, SSD, 32/64 Worker Nodes, 768/1536 Cores, (768/1536M 768/1536R)
- RDMA-based design for Spark 1.5.1
- 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)

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

- Challenges of Deep Learning over Big Data (DLoBD)
  - 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.

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/