16th ANNUAL WORKSHOP 2020

Designing a Deep-Learning Aware MPI Library: An MVAPICH2 Approach

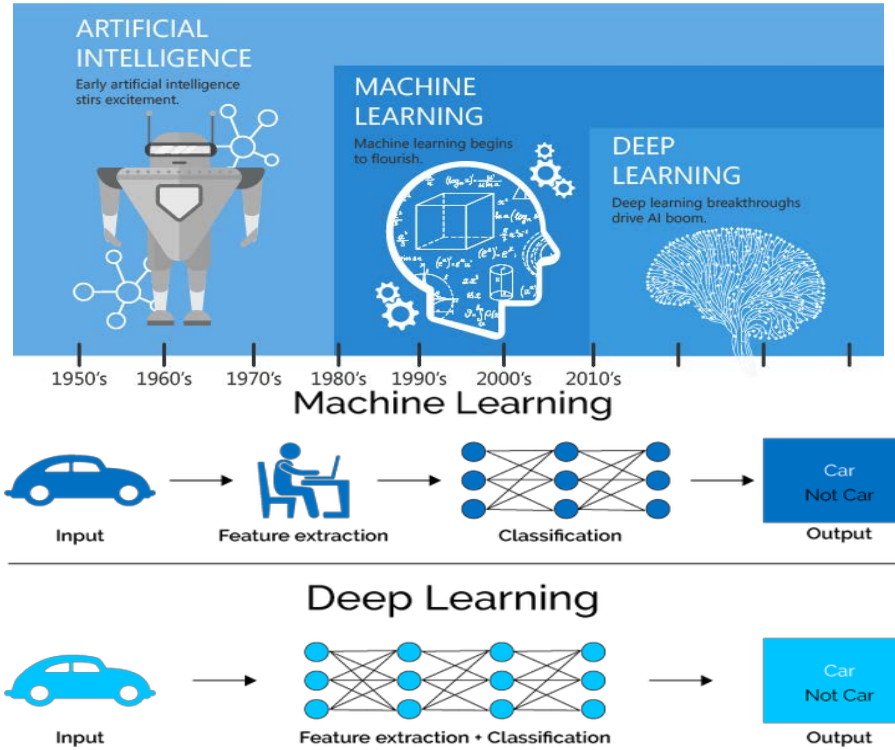
Ammar Ahmad Awan, Jahanzeb Maqbool Hashmi, Ching-Hsiang Chu, Hari Subramoni, and Dhabaleswar K. (DK) Panda

The Ohio State University

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

WHAT IS DEEP LEARNING?

- Deep Learning (DL)
  - A subset of Machine Learning that uses Deep Neural Networks (DNNs)
  - **Perhaps, the most revolutionary subset!**
- Based on learning data representation
- Examples Convolutional Neural Networks, Recurrent Neural Networks, Hybrid Networks
- Data Scientist or Developer Perspective
  1. Identify DL as solution to a problem
  2. Determine Data Set
  3. Select Deep Learning Algorithm to Use
  4. Use a large data set to train an algorithm



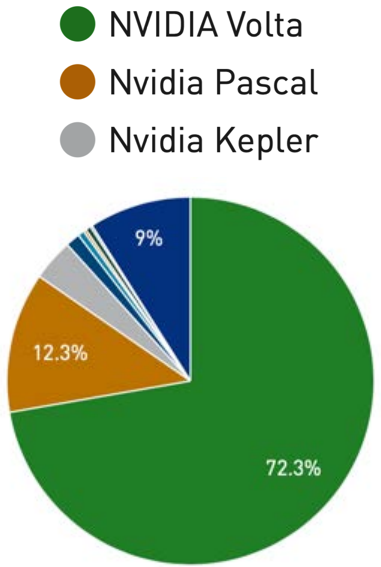
Courtesy: https://hackernoon.com/difference-between-artificial-intelligence-machine-learning-and-deep-learning-1pcv3zeg, https://blog.dataiku.com/ai-vs.-machine-learning-vs.-deep-learning

DEEP LEARNING AND HIGH-PERFORMANCE ARCHITECTURES

- **NVIDIA GPUs are the main driving force for faster training of DL models**
  - The ImageNet Challenge - (ILSVRC) -- 90% of the teams used GPUs (2014)
  - Deep Neural Networks (DNNs) like ResNet(s) and Inception

- **However, High Performance Architectures for DL and HPC are evolving**
  - 135/500 Top HPC systems use NVIDIA GPUs (Nov '19)
  - DGX-1 (Pascal) and DGX-2 (Volta)
    - Dedicated DL supercomputers
  - Cascade-Lake Xeon CPUs have 28 cores/socket (TACC Frontera– #5 on Top500)
  - AMD EPYC (Rome) CPUs have 64 cores/socket (Upcoming DOE Clusters)
  - AMD GPUs will be powering the Frontier – DOE's Exascale System at ORNL
  - Domain Specific Accelerators for DNNs are also emerging

*https://blogs.nvidia.com/blog/2014/09/07/imagenet/



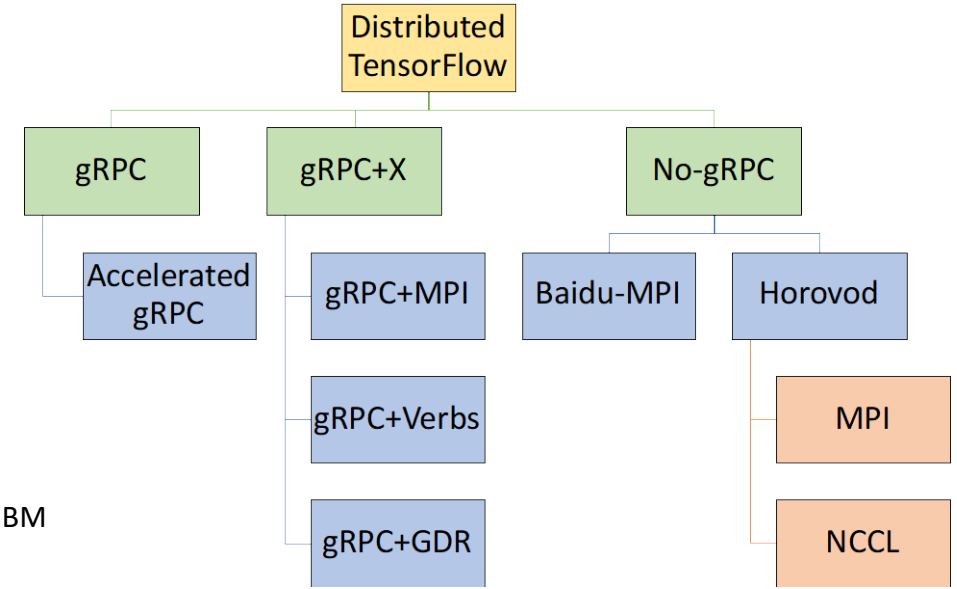
Accelerator/CP Performance Share
www.top500.org

BROAD CHALLENGE: EXPLOITING HPC FOR DEEP LEARNING

How to efficiently scale-out Deep Learning (DL) workloads by better exploiting High Performance Computing (HPC) resources like Multi-/Many-core CPUs and GPUs?

HIGH-PERFORMANCE DISTRIBUTED DATA PARALLEL TRAINING WITH TENSORFLOW

- **gRPC**
  - Officially available and supported
  - 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 DK Panda., "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs, and Performance Evaluation", CCGrid'19

OVERVIEW OF THE MVAPICH2 PROJECT

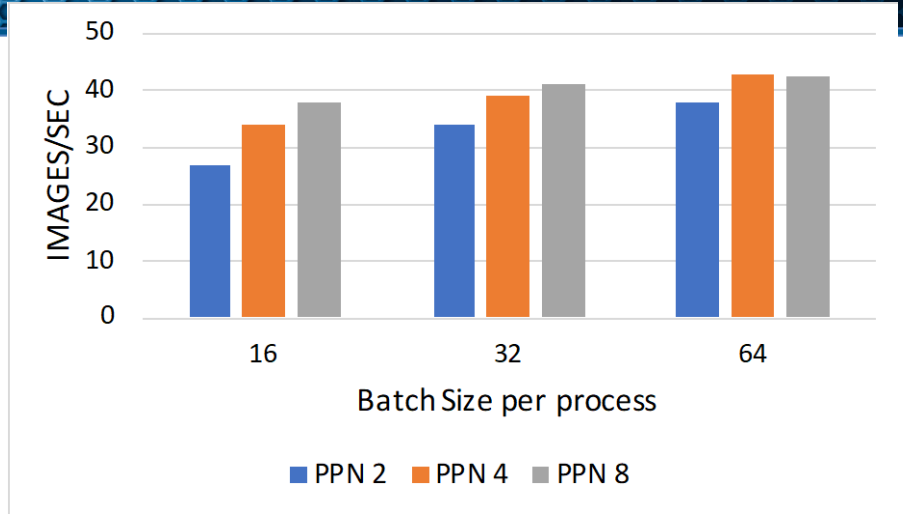
- **High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)**
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.1), Started in 2001, First version available in 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - Support for GPGPUs (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 3,050 organizations in 89 countries**
  - **More than 665,000 (> 0.6 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (June '19 ranking)
    - 3rd ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, China
    - 8th, 391,680 cores (ABCI) in Japan
    - 16th, 556,104 cores (Oakforest-PACS) in Japan
    - 19th, 367,024 cores (Stampede2) at TACC
    - 31st, 241,108-core (Pleiades) at NASA and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, OpenHPC, and Spack)
  - http://mvapich.cse.ohio-state.edu
- **Empowering Top500 systems for over a decade**

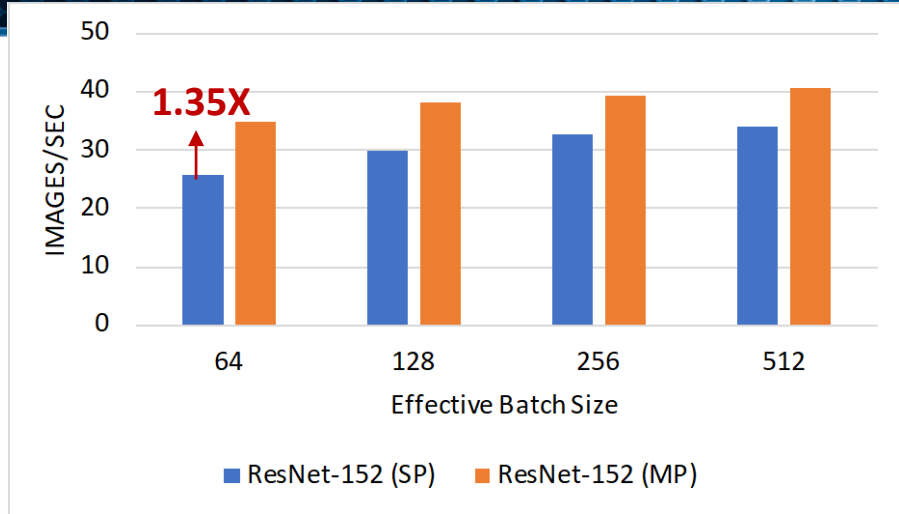
Partner in the 5th ranked TACC Frontera System

HIGH-PERFORMANCE DEEP LEARNING

- **CPU-based Deep Learning**
  - **Using MVAPICH2-X**
- **GPU-based Deep Learning**
  - Using MVAPICH2-GDR

CPU-BASED TRAINING: SINGLE-NODE MULTI-PROCESS (MP) MODE





ResNet-152 Training performance

- **BS=64, 4ppn is better; BS=32, 8ppn is slightly better**
- **However, keeping effective batch size (EBS) low is more important! – Why? (DNN does not converge to SOTA when batch size is large)**

ResNet-152: Single Process (SP) vs. Multi-Process(MP)

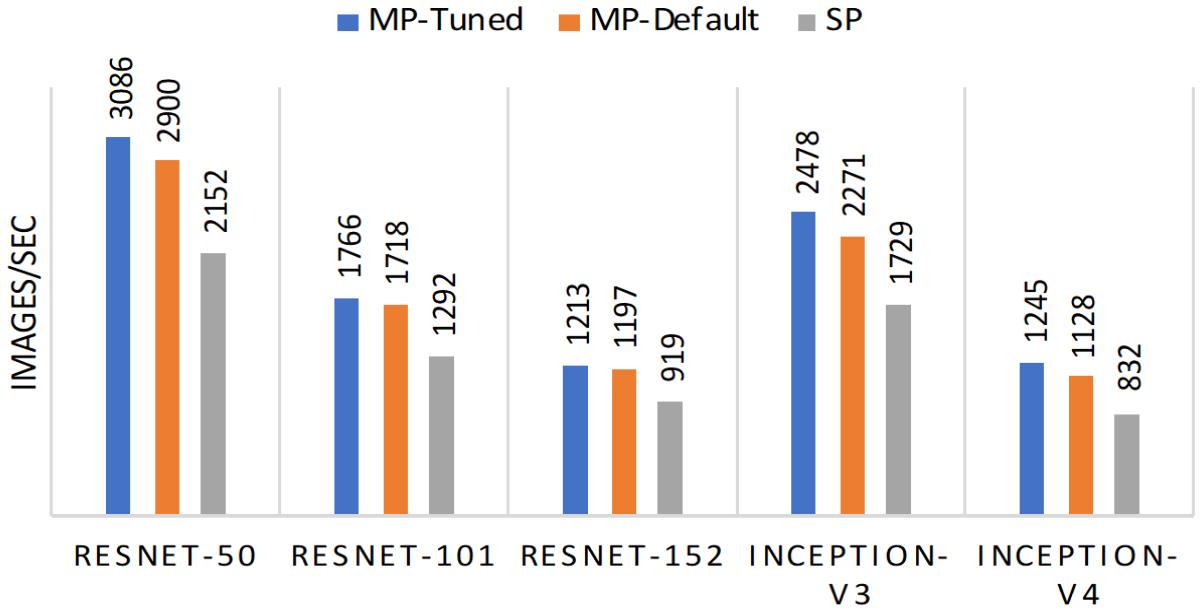
- **MP is better for all effective batch sizes**
- **Up to 1.35X better performance for MP compared to SP for BS=64.**

A. Jain, A. A. Awan, Q. Anthony, H. Subramoni and DK Panda, "Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters", Cluster '19

CPU-BASED TRAINING: MULTI-PROCESS (MN): MP VS. SP?

Skylake-3 (48 cores, 96 threads)

- **Scale—32 nodes**
- **MP-Tuned—up to 1.5X better than SP**
- **MP-Tuned—10% better than MP-Default**
- **Why MP-Tuned is better?**
  - **Uses the best possible number of inter-op and intra-op threads**

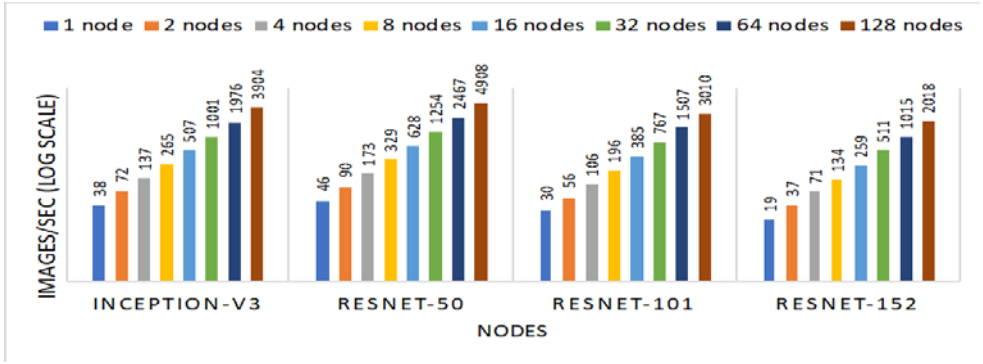


A. Jain, A. A. Awan, Q. Anthony, H. Subramoni and DK Panda, "Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters", Cluster '19

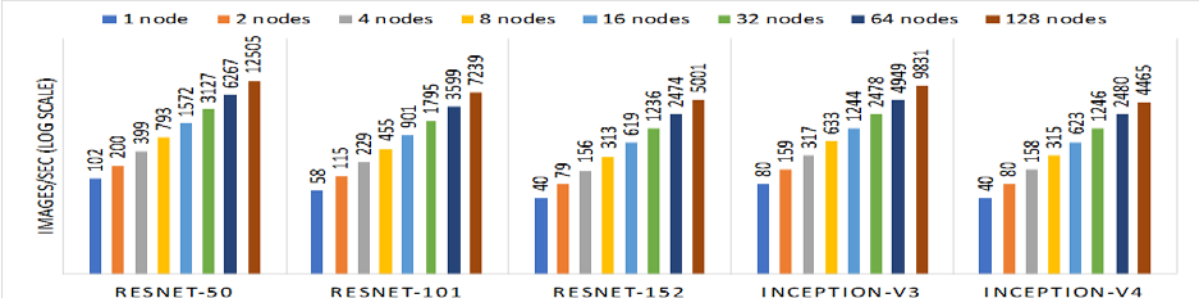
MULTI-NODE MULTI-PROCESS (MN): TENSORFLOW VS. PYTORCH

- This is an early experience with PyTorch
- TensorFlow is up to 2.5X faster than PyTorch for 128 Nodes.
- TensorFlow: up to 125X speedup for ResNet-152 on 128 nodes
- PyTorch: Scales well but overall lower performance than TensorFlow

PyTorch



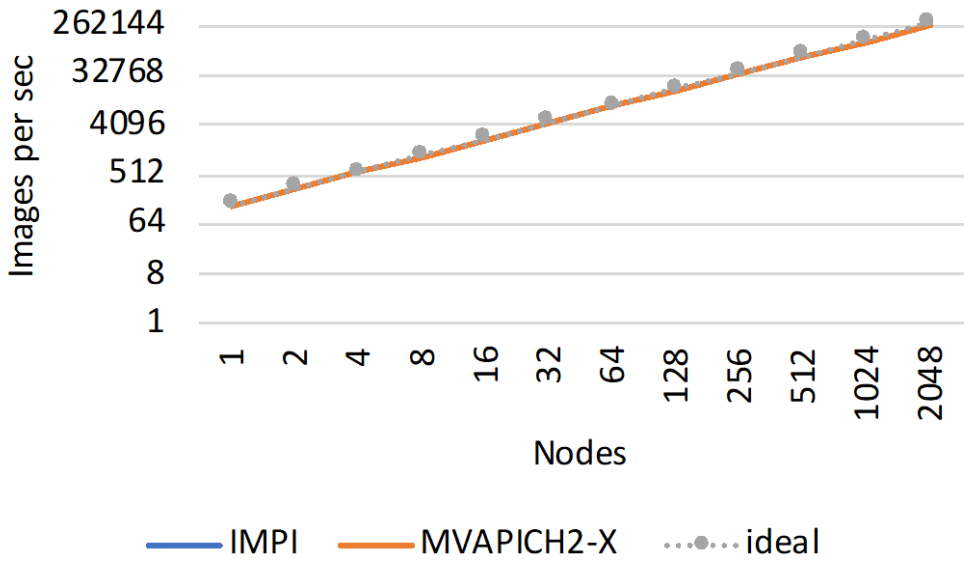
TensorFlow



A. Jain, A. A. Awan, Q. Anthony, H. Subramoni and DK Panda, "Performance Characterization of DNN Training using TensorFlow and PyTorch on Modern Clusters", Cluster '19

SCALING RESNET-50 ON TACC FRONTERA: 2,048 NODES!

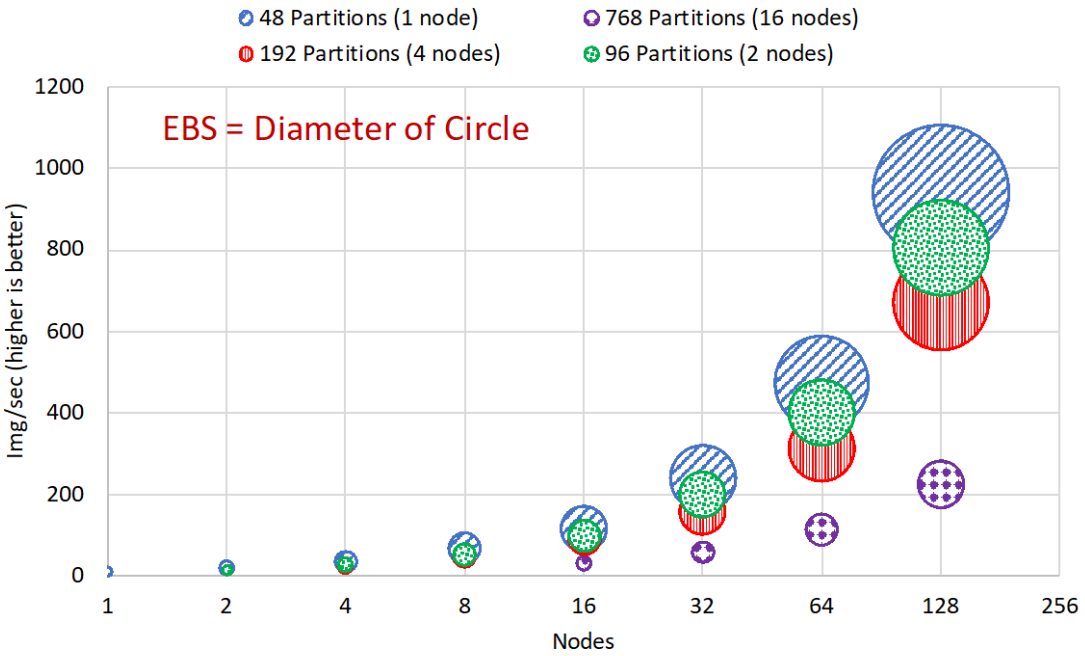
- Scaled TensorFlow to 2048 nodes on Frontera using MVAPICH2 and IntelMPI
- MVAPICH2 and IntelMPI give similar performance for DNN training
- Report a peak of 260,000 images/sec on 2048 nodes
- On 2048 nodes, ResNet-50 can be trained in 7 minutes!



A. Jain, A. A. Awan, H. Subramoni and DK Panda, "Scaling TensorFlow, PyTorch, and MXNet using MVAPICH2 for High-Performance Deep Learning on Frontera", DLS '19 (in conjunction with SC '19).

BENCHMARKING HYPAR-FLOW ON STAMPEDE2

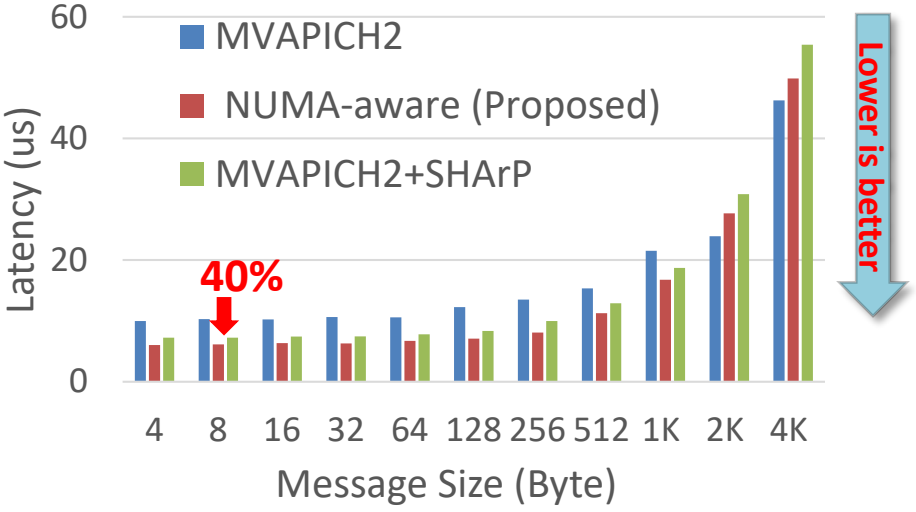
- **CPU based Hybrid-Parallel (Data Parallelism and Model Parallelism) training on Stampede2**
- **Benchmark developed for various configuration**
  - Batch sizes
  - No. of model partitions
  - No. of model replicas
- **Evaluation on a very deep model**
  - ResNet-1000 (a 1,000-layer model)



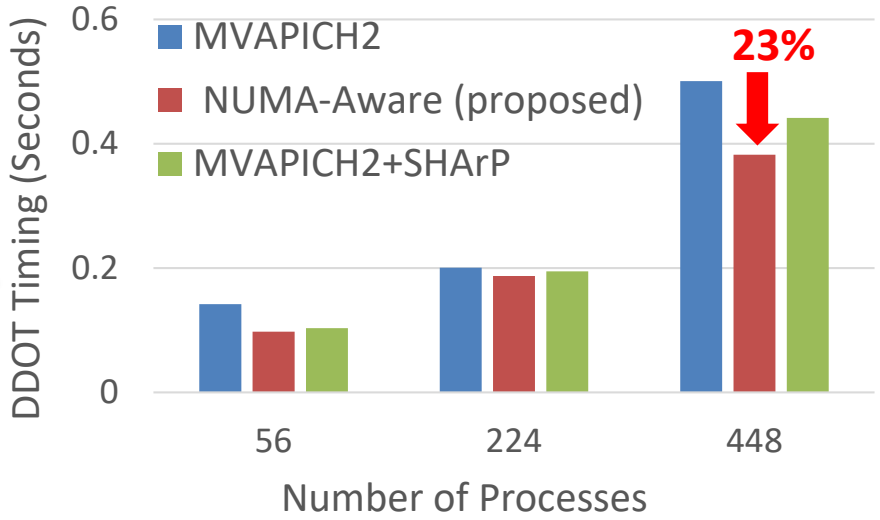
110x speedup on 128 Intel Xeon Skylake nodes (TACC Stampede2 Cluster)

A. A. Awan, A. Jain, Q. Anthony, H. Subramoni, and DK Panda, "HyPar-Flow: Exploiting MPI and Keras for Hybrid Parallel Training of TensorFlow models", arXiv '19. https://arxiv.org/pdf/1911.05146.pdf

PERFORMANCE OF NUMA-AWARE SHARP DESIGN ON XEON + IB CLUSTER



Lower is better



OSU Micro Benchmark (16 Nodes, 28 PPN)

HPCG (16 nodes, 28 PPN)

- As the message size decreases, the benefits of using Socket-based design increases
- NUMA-aware design can reduce the latency by up to 23% for DDOT phase of HPCG

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 since MVAPICH2-X 2.3b

SHARED ADDRESS SPACE (XPMEM-BASED) COLLECTIVES

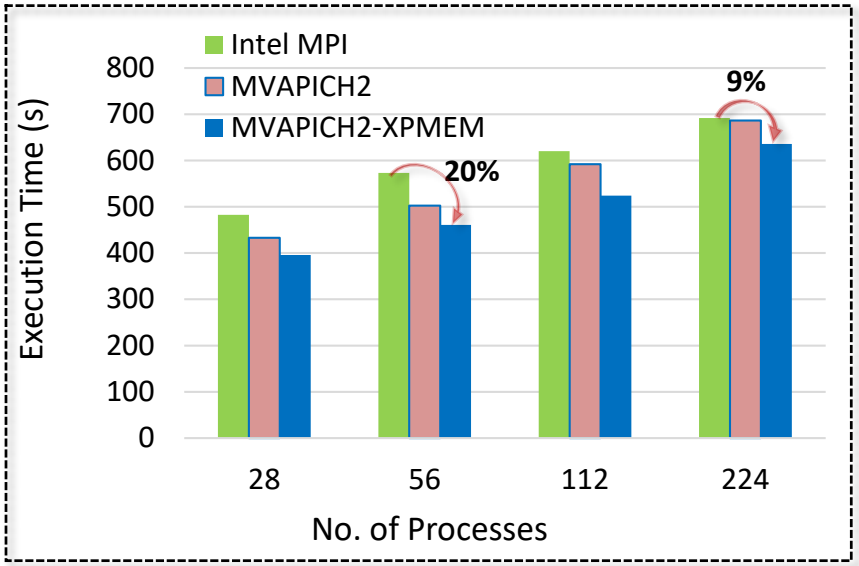
- **Offload Reduction computation and communication to peer MPI ranks**
  - Every Peer has direct "load/store" access to other peer's buffers
  - Multiple pseudo roots independently carry-out reductions for intra-and inter-node
  - Directly put reduced data into root's receive buffer
- ***True "Zero-copy"*** **design for Allreduce and Reduce**
  - No copies require during the entire duration of Reduction operation
  - Scalable to multiple nodes
- ***Zero contention*** **overheads as memory copies happen in "*user-space*"**

Available since MVAPICH2-X 2.3rc1

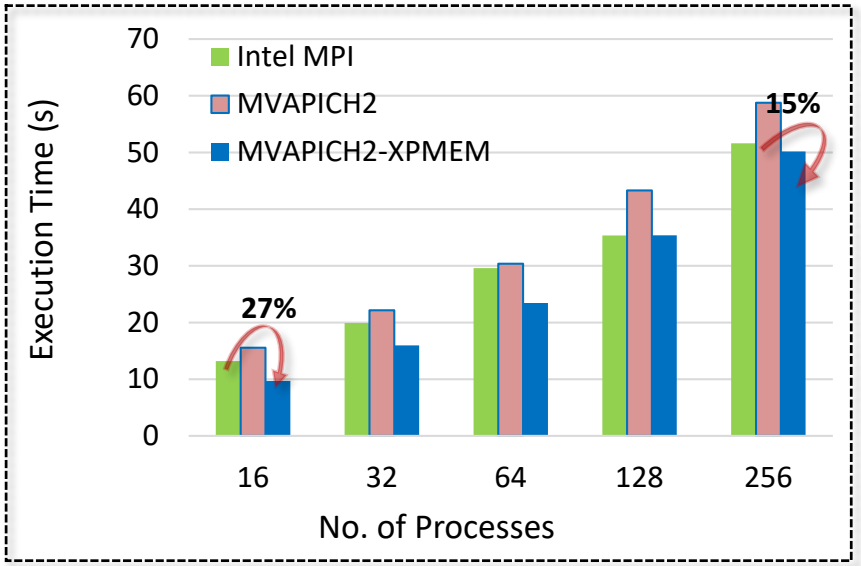
J. Hashmi, S. Chakraborty, M. Bayatpour, H. Subramoni, and D. Panda, Designing Efficient Shared Address Space Reduction Collectives for Multi-/Many-cores, International Parallel & Distributed Processing Symposium (IPDPS '18), May 2018.

APPLICATION-LEVEL BENEFITS OF XPMEM-BASED COLLECTIVES

CNTK AlexNet Training
(Broadwell, B.S=default, iteration=50, ppn=28)



MiniAMR (Broadwell, ppn=16)



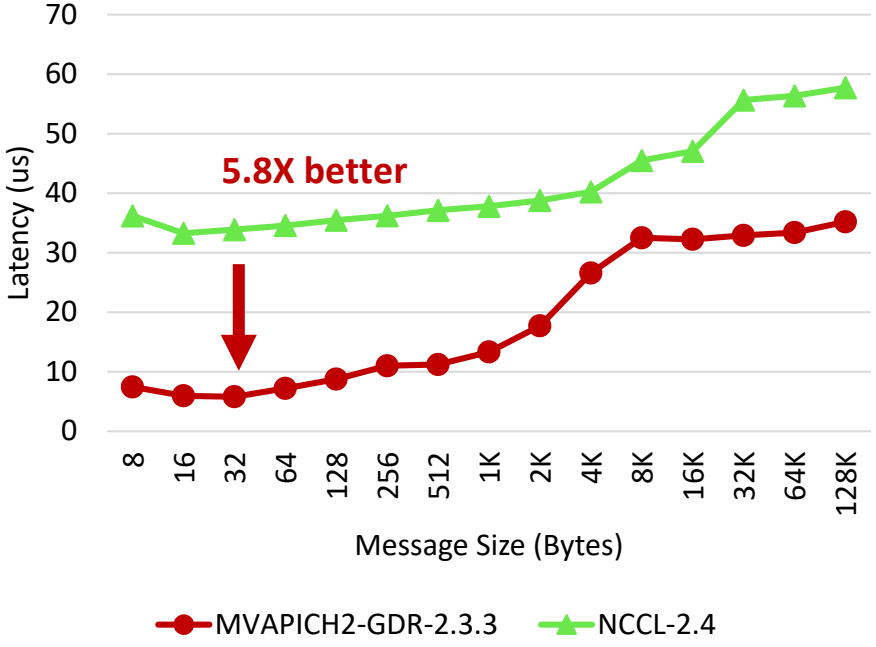
- Up to **20%** benefits over IMPI for CNTK DNN training using AllReduce
- Up to **27%** benefits over IMPI and up to **15%** improvement over MVAPICH2 for MiniAMR application kernel

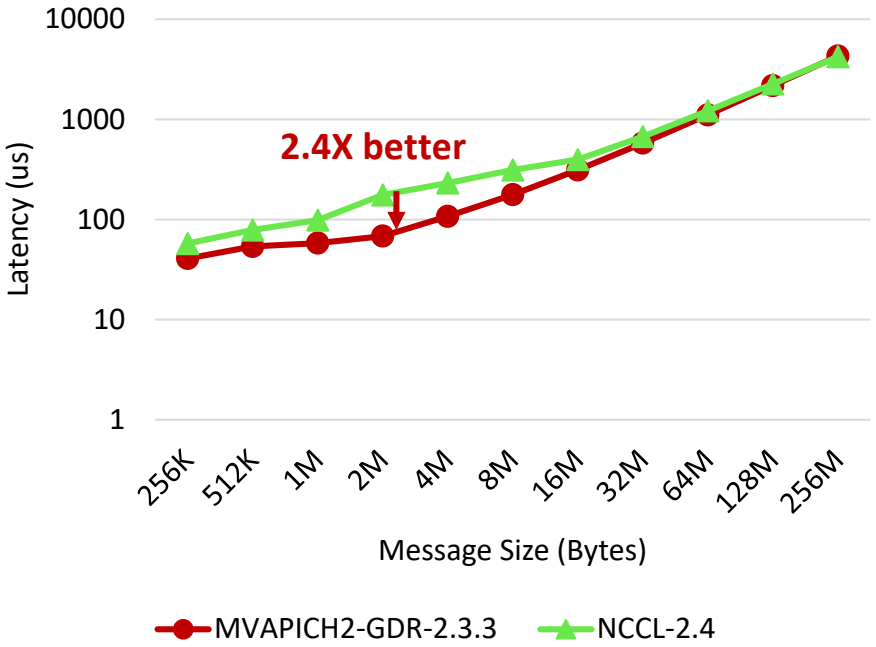
HIGH-PERFORMANCE DEEP LEARNING

- CPU-based Deep Learning
  - Using MVAPICH2-X
- **GPU-based Deep Learning**
  - Using MVAPICH2-GDR

MVAPICH2-GDR VS. NCCL2 – ALLREDUCE ON GPU SYSTEMS

- Optimized designs in upcoming MVAPICH2-GDR offer better/comparable performance for most cases
- MPI_Allreduce (MVAPICH2-GDR) vs. ncclAllreduce (NCCL2) on 1 DGX-2 node (16 Volta GPUs)

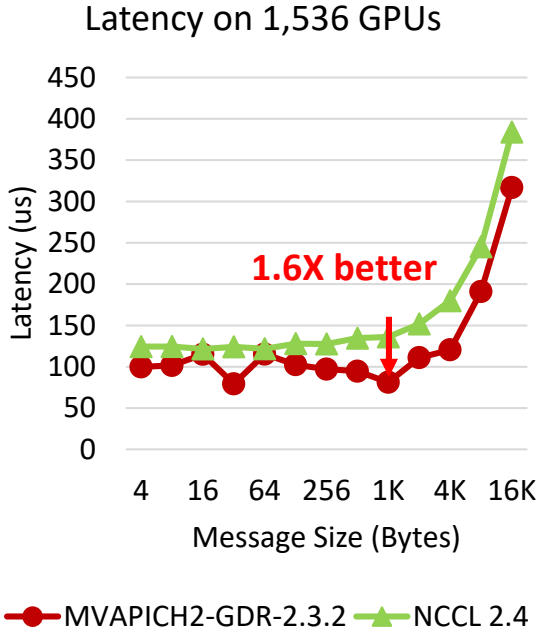


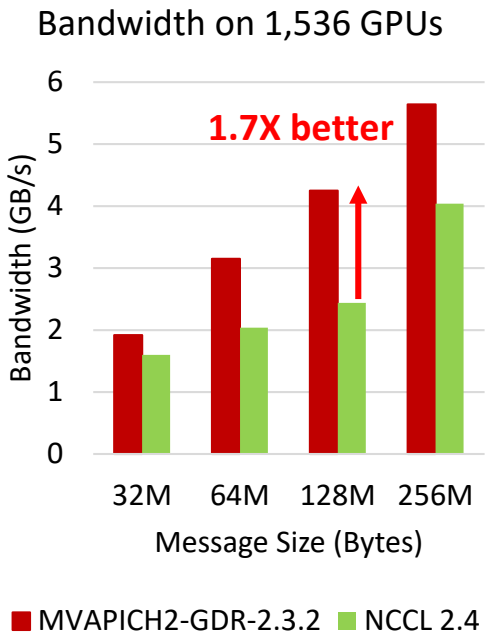


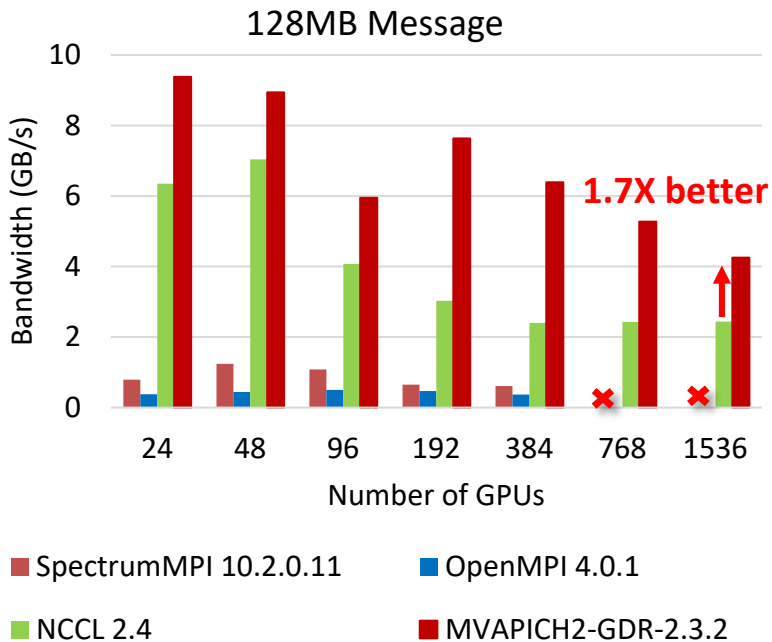
Platform: Nvidia DGX-2 system (16 Nvidia Volta GPUs connected with NVSwitch), CUDA 9.2

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**



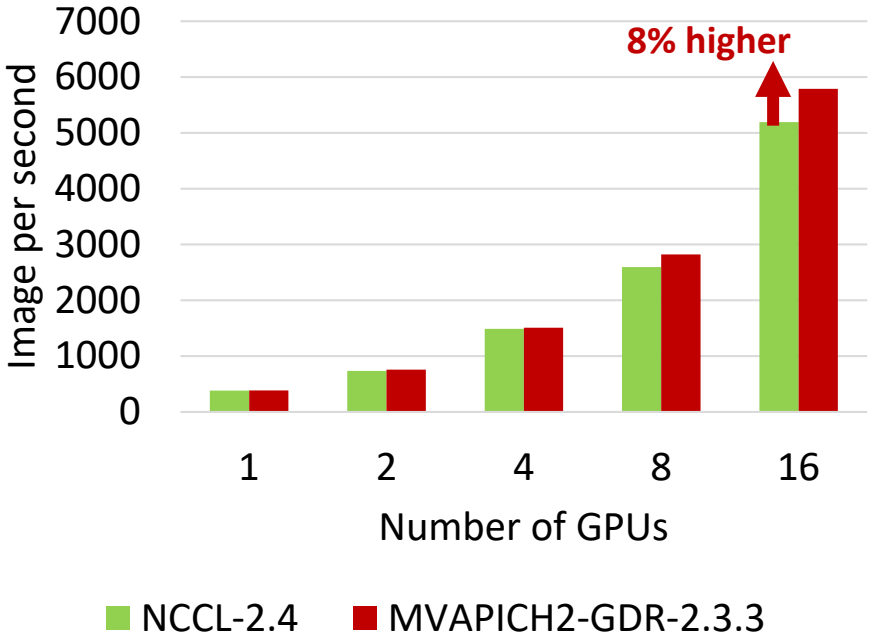




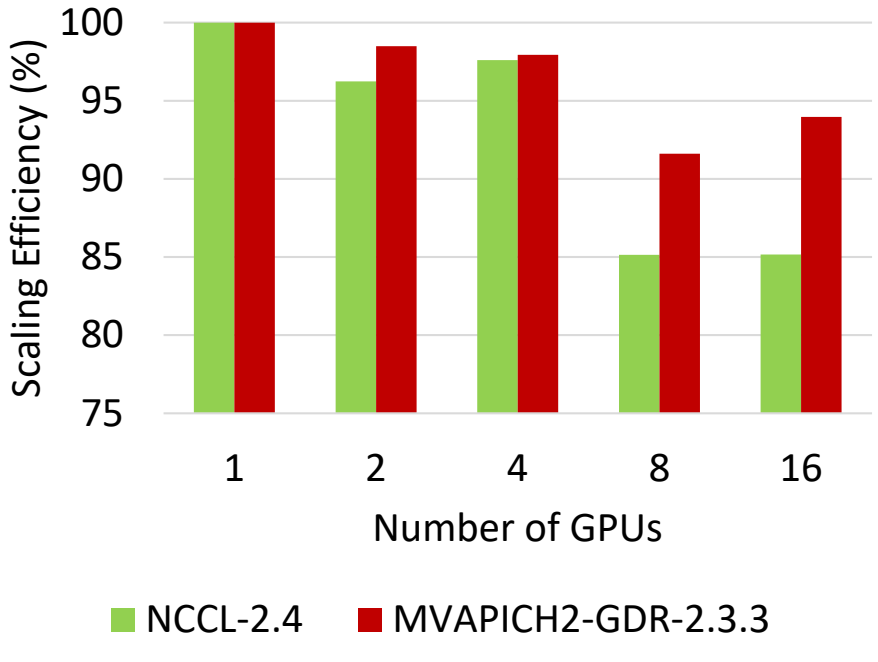
Platform: Dual-socket IBM POWER9 CPU, 6 NVIDIA Volta V100 GPUs, and 2-port InfiniBand EDR Interconnect

MVAPICH2-GDR VS. NCCL2 – RESNET-50 TRAINING

- **ResNet-50 Training using TensorFlow benchmark on 1 DGX-2 node (8 Volta GPUs)**



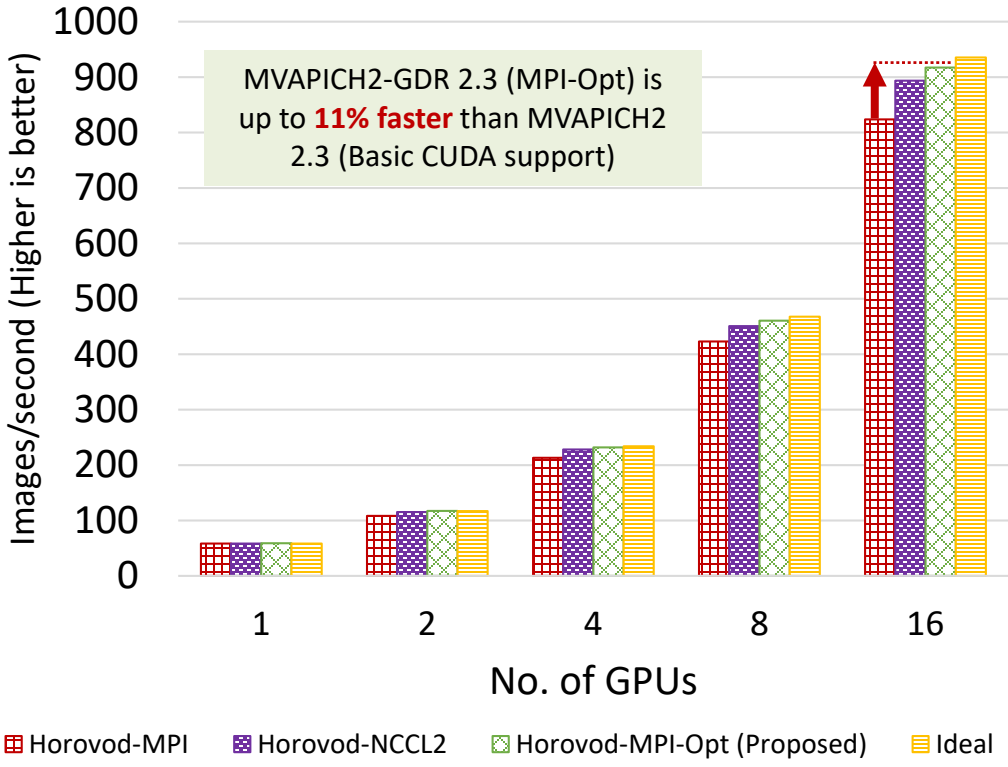
$$\text{Scaling Efficiency} = \frac{\text{Actual throughput}}{\text{Ideal throughput at scale}} \times 100\%$$



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

EXPLOITING CUDA-AWARE MPI FOR TENSORFLOW (HOROVOD)

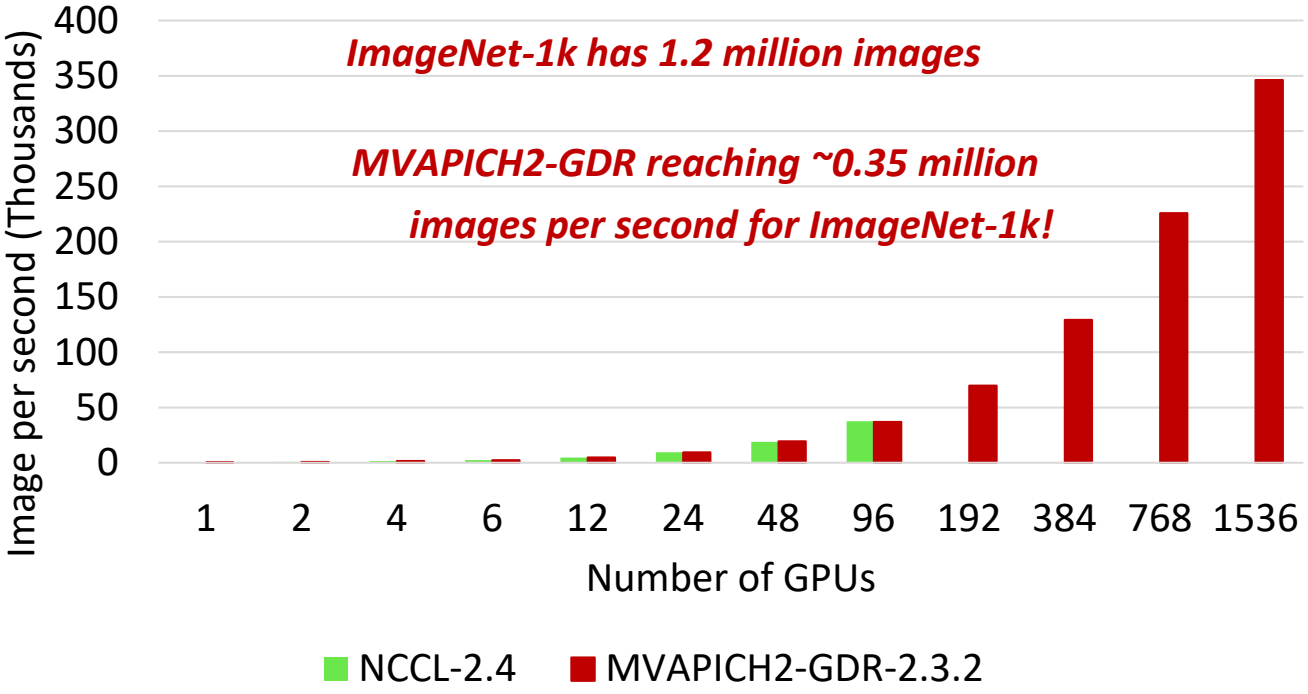
- MVAPICH2-GDR offers excellent performance via advanced designs for MPI_Allreduce.
- Up to 11% better performance on the RI2 cluster (16 GPUs)
- Near-ideal – 98% scaling efficiency



A. A. Awan, J. Bedorf, C-H Chu, H. Subramoni, and DK Panda., "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs, and Performance Evaluation", CCGrid'19

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!**



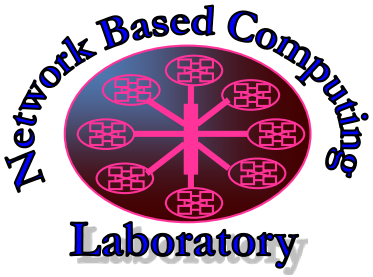
Platform: The Summit Supercomputer (#1 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 9.2

*We observed errors for NCCL2 beyond 96 GPUs

THANK YOU!

{awan.10, hashmi.29, chu.368}@osu.edu

subramon@cse.ohio-state.edu, panda@cse.ohio-state.edu



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

MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

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

HiDL
High-Performance Deep Learning

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