MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

HiDL
High-Performance Deep Learning

Benchmarking Deep Learning Workloads on Large-scale HPC Systems

Benchmarking in the Datacenter Workshop @ PPoPP '20

Ammar Ahmad Awan and Dhabaleswar K. Panda

awan.10@osu.edu, panda@cse.ohio-state.edu

The Ohio State University

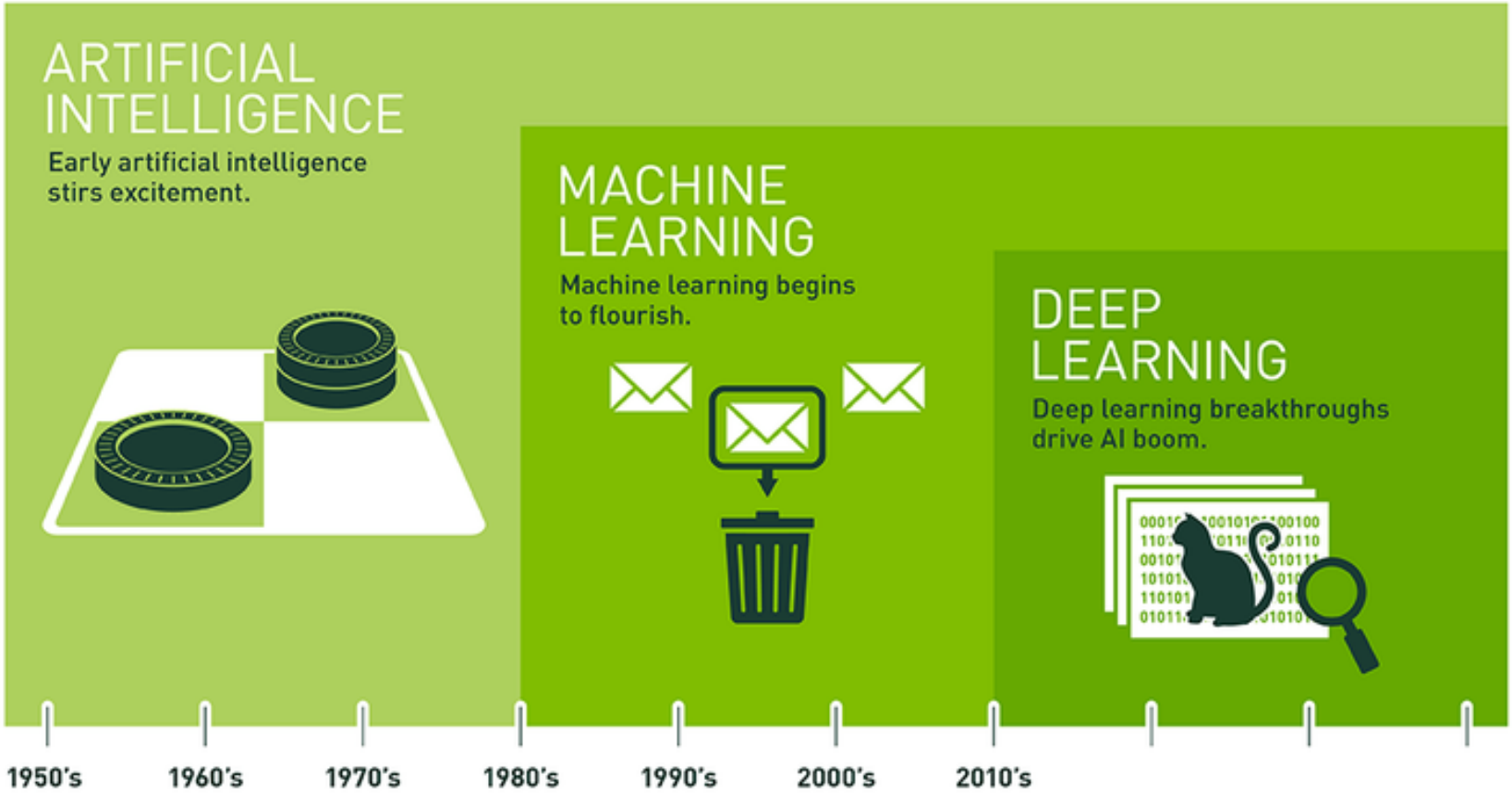
Follow us on

https://twitter.com/mvapich

Agenda

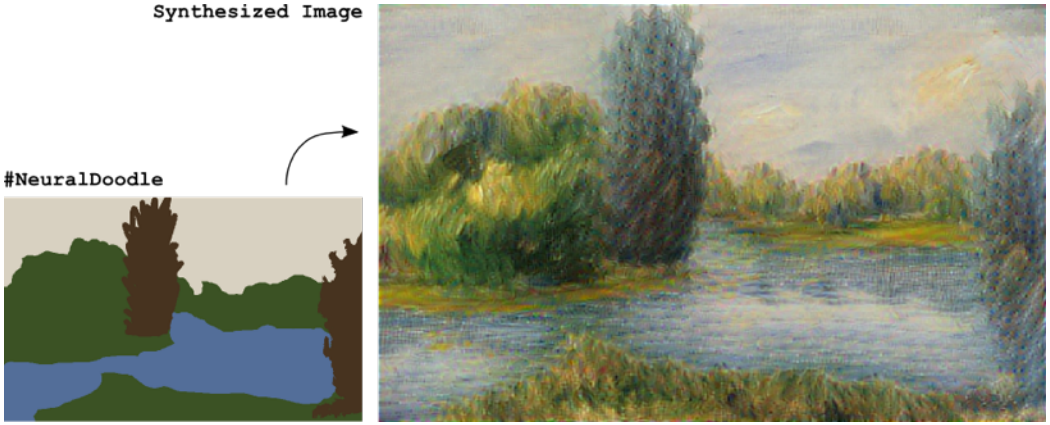
- Introduction
- Background
- ML/DL Benchmarks
- Solutions and Case Studies
- Conclusion and Future Directions

Overview of Artificial Intelligence

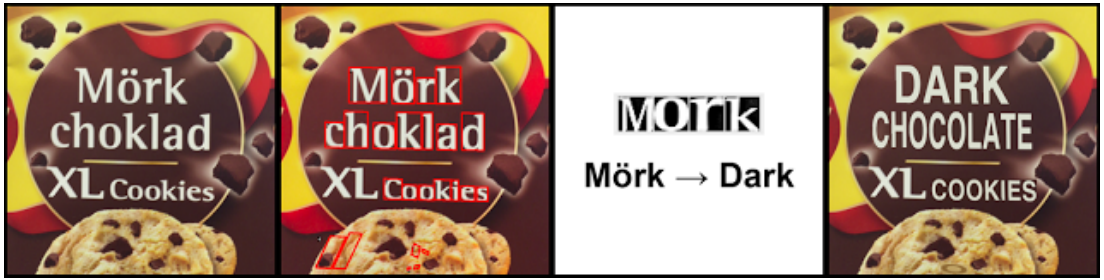


Courtesy: http://www.zdnet.com/article/caffe2-deep-learning-wide-ambitions-flexibility-scalability-and-advocacy/

Applications: Style Transfer, Caption Generation, Translation, etc.



Courtesy: https://github.com/alexjc/neural-doodle

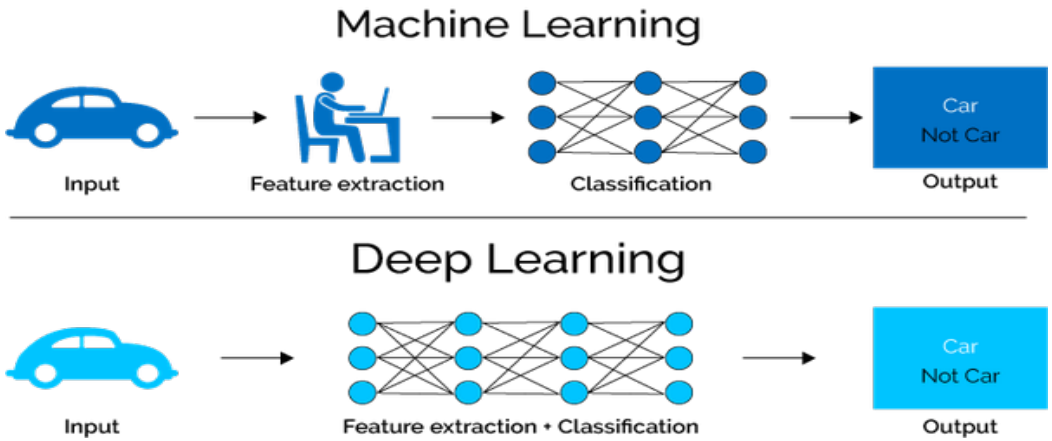


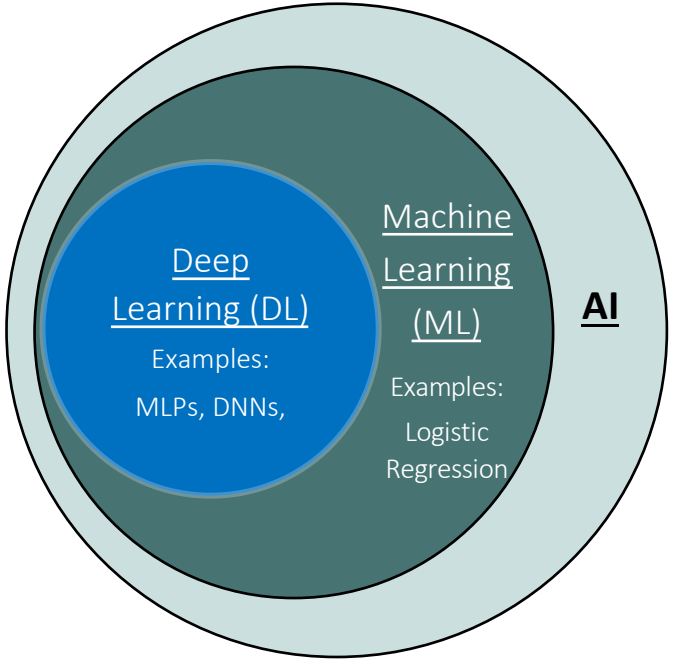
Courtesy: https://research.googleblog.com/2015/07/how-google-translate-squeezes-deep.html

Courtesy: https://machinelearningmastery.com/inspirational-applications-deep-learning/

The Deep Learning (DL) Revolution

- DL – a revolutionary sub-set of ML
  - Feature extraction vs. hand-crafted features





Adopted from: http://www.deeplearningbook.org/contents/intro.html

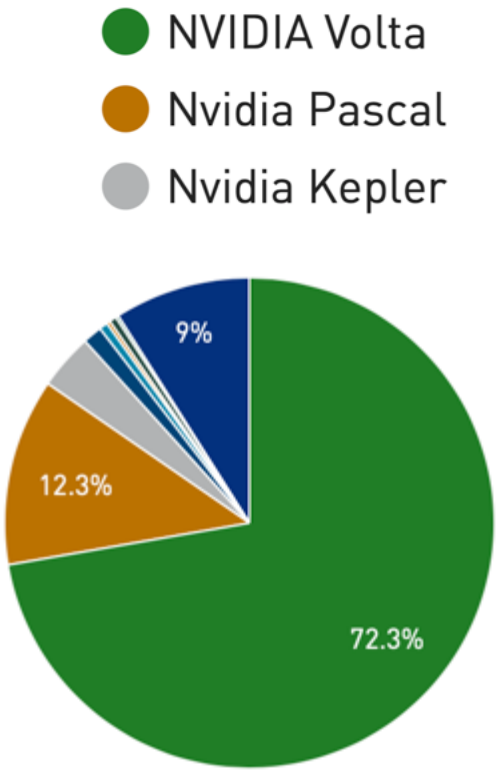
- Key success: Deep Neural Networks (DNNs)
  - Everything was invented in late 80s except ?

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 on GPUs

- NVIDIA GPUs - driving force for faster DL!
  - 90% ImageNet teams used GPUs (2014*)
  - DNNs like Inception, ResNet(s), NASNets, and AmoebaNets
  - Natural fit for DL workloads – throughput-oriented

- High Performance Computing (HPC) arena
  - 135/500 Top HPC systems used NVIDIA GPUs (Nov '19)
  - CUDA-Aware Message Passing Interface (MPI)
    - MVAPICH2-GDR, SpectrumMPI, OpenMPI, etc.
  - DGX-1/DGX-2- Dedicated DL supercomputers

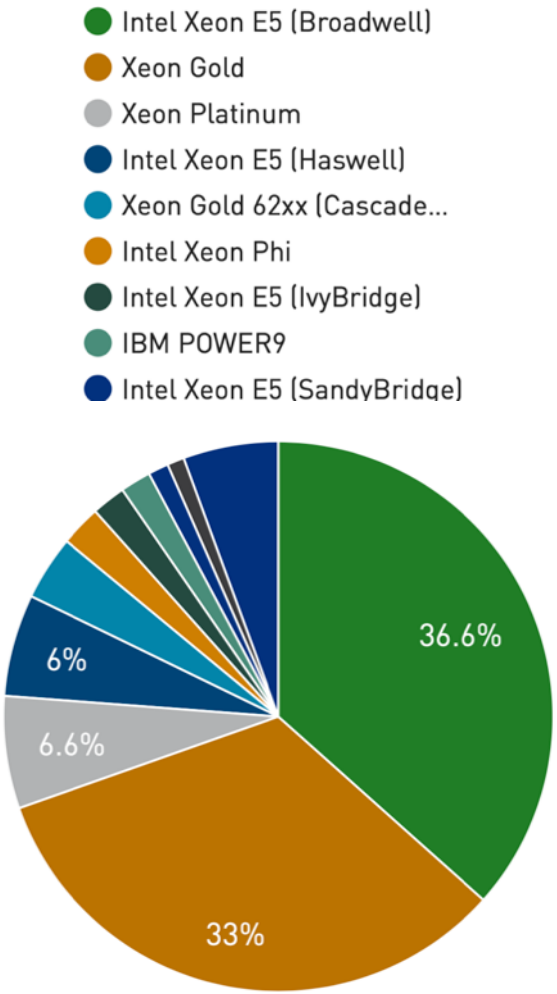


www.top500.org

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

Deep Learning on CPUs

- CPUs (dense many-cores) are emerging
- CPUs exist on nodes with GPUs
  - Many-core Xeon, POWER9, EPYC, ARM, etc.
- Are CPUs really *10x – 100x* slower than GPUs? [1-3]
- But CPU-based DL is getting much better and faster
  - MKL-DNN, Vectorization, MPI optimized for large messages
  - Data-Parallelism (Intel-Caffe – MLHPC '17)
  - Model/Hybrid-Parallelism (HyPar-Flow – ISC '20)

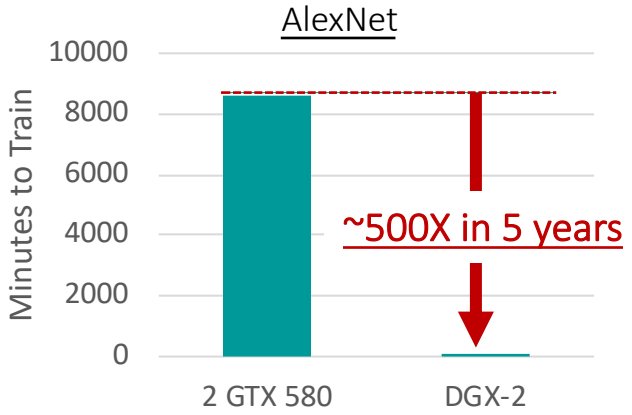


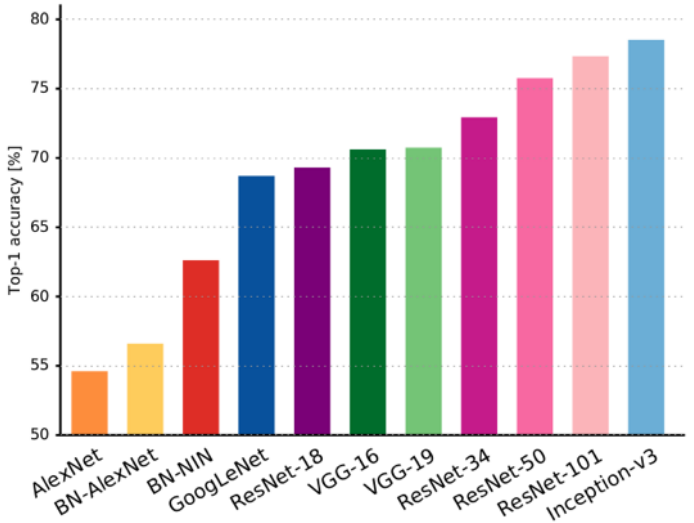
1. https://dl.acm.org/citation.cfm?id=1993516, 2. http://ieeexplore.ieee.org/abstract/document/5762730/,
3. https://dspace.mit.edu/bitstream/handle/1721.1/51839/MIT-CSAIL-TR-2010-013.pdf?sequence=1

Why HPC and DL?

Three Key Pieces in the story:

- Computability of DNNs
  - *HPC enables faster DL!*
- Datasets – ImageNet and beyond…
- State-of-the-art Accuracy – Vision, NLP, Translation, etc.





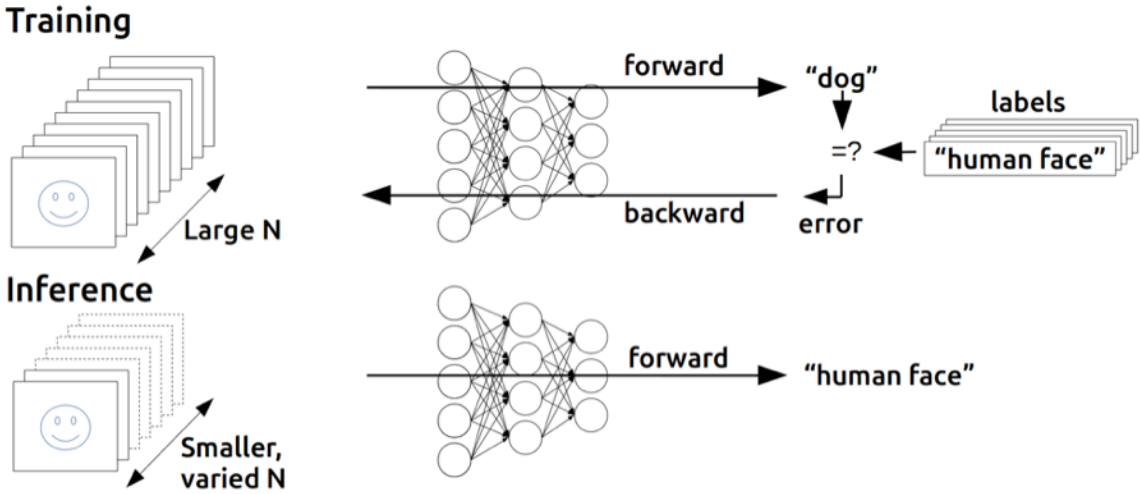
Courtesy: A. Canziani et al., "An Analysis of Deep Neural Network Models for Practical Applications", *CoRR*, 2016.

Agenda

- Introduction
- Background
- ML/DL Benchmarks
- Solutions and Case Studies
- Conclusion and Future Directions

Two Phases in Deep Learning

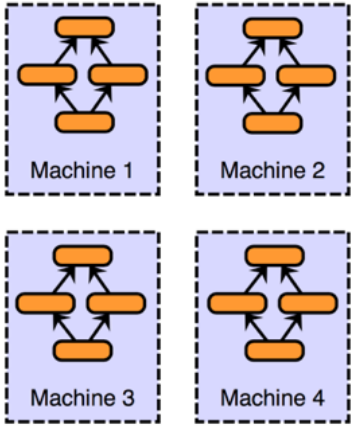
- Training is compute-intensive
  - Many passes over data
  - Can take days to weeks
  - Model adjustment is done
- Inference
  - Single pass over the data
  - Takes seconds
  - No model adjustment
- *Challenge: How to make **"Training"** faster?*
  - Exploit HPC hardware and software



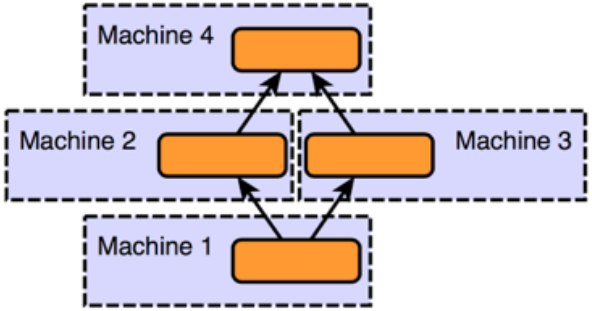
Courtesy: https://devblogs.nvidia.com/

Parallelization Strategies for DNN Training

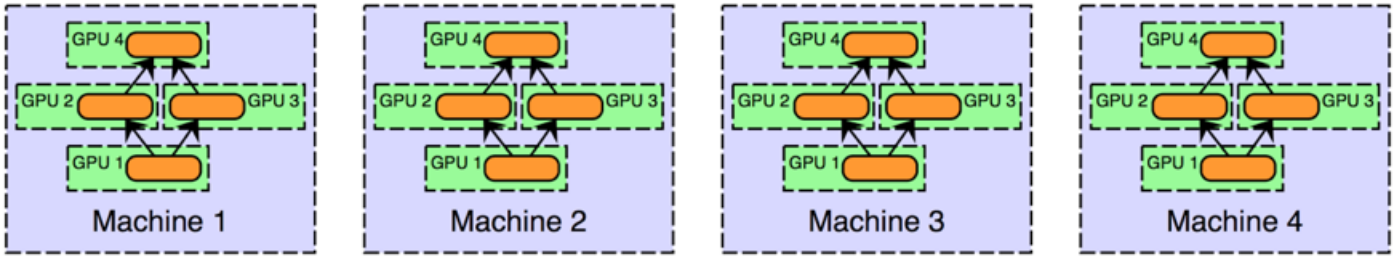
- Data Parallelism (most common)
- Model and Hybrid Parallelism (emerging)
- 'X'-Parallelism
  - 'X' —> Spatial, Channel, Filter, etc.



Data Parallelism



Model Parallelism



Hybrid (Model and Data) Parallelism

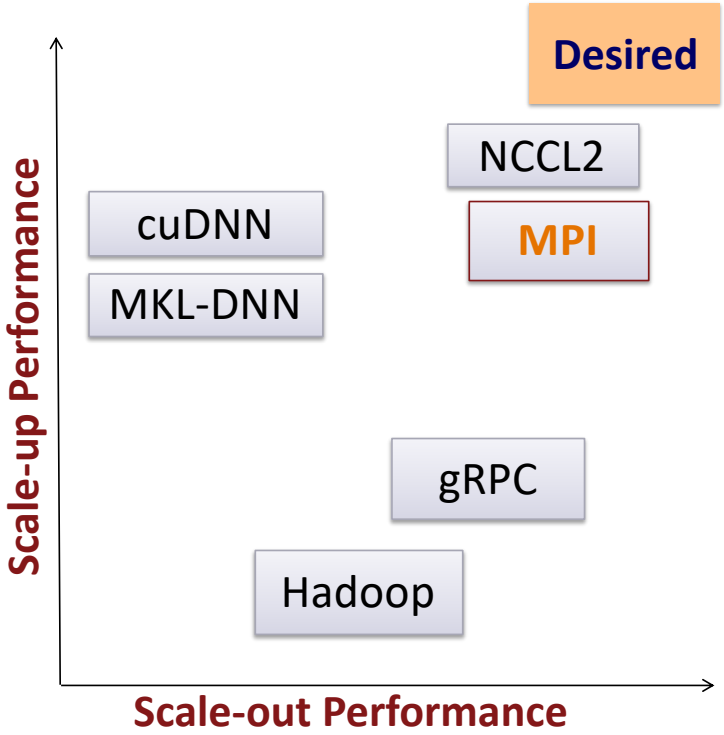
Courtesy: http://engineering.skymind.io/distributed-deep-learning-part-1-an-introduction-to-distributed-training-of-neural-networks

Agenda

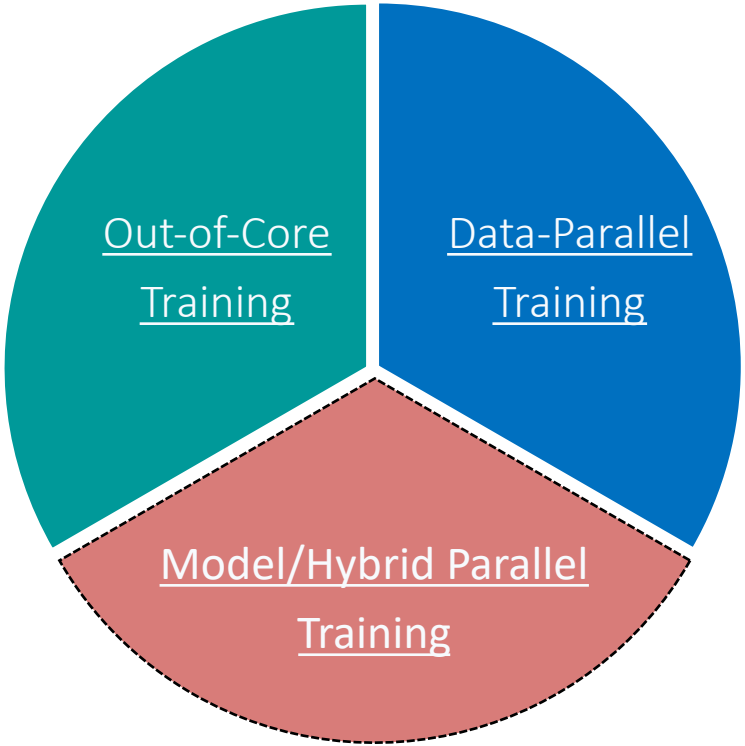
- Introduction
- Background
- ML/DL Benchmarks
- Solutions and Case Studies
- Conclusion and Future Directions

Benchmarking DL Workloads

Benchmarking Scale-up and Scale-out performance of DL workloads on large-scale HPC systems



DNN Training: A Complex Solution Space



In-depth Performance Characterization and Profiling Analysis

Several Efforts focused on DL Benchmarking

MLPerf
(mlperf.org – strong industry support)

Deep500
(ETH Zurich)

DAWNBench
(Stanford)

Many benchmarks but most are not suitable for HPC Systems

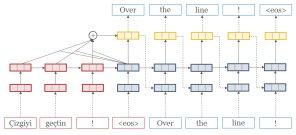
DL Bench
(Hong Kong University)

convnet-benchmarks
(Soumith Chintala)

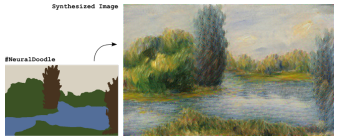
tf_cnn_benchmarks
(TensorFlow)

Time Benchmark
(CAFFE)

Simplified Deep Learning Stack



Language



Style Transfer

Application Areas

Translation

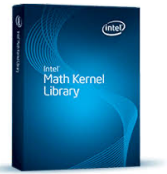
Image Captioning

Frameworks

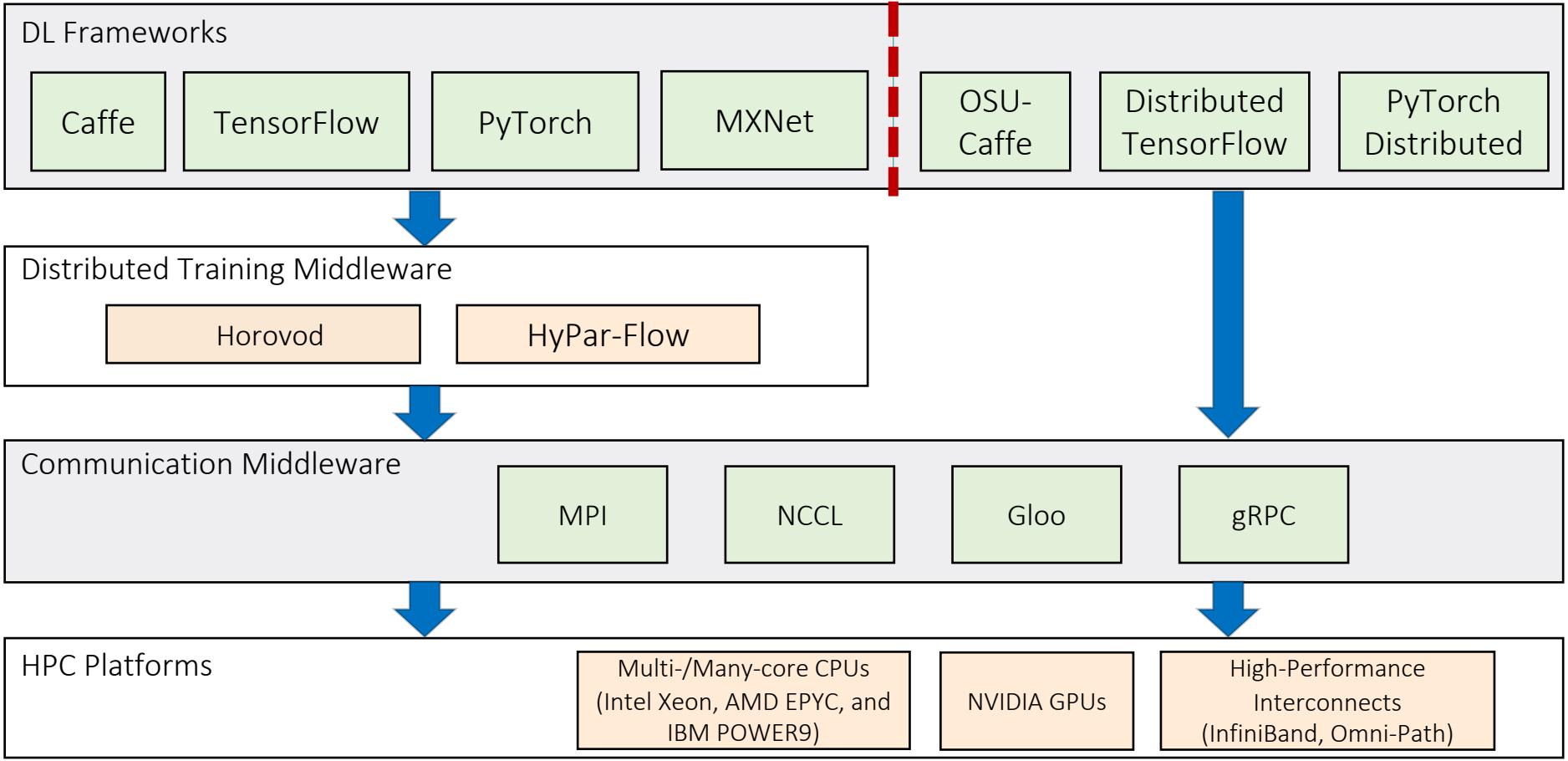
Communication Libraries

I/O (?)

Compute Libraries



DL Execution Stack: Details

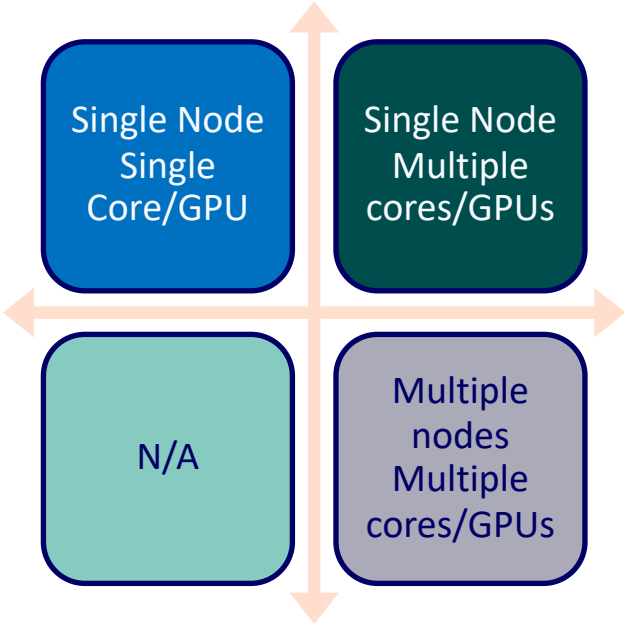


Agenda

- Introduction
- Background
- ML/DL Benchmarks
- Solutions and Case Studies
- Conclusion and Future Directions

Solutions and Case Studies: Different Benchmarking Directions

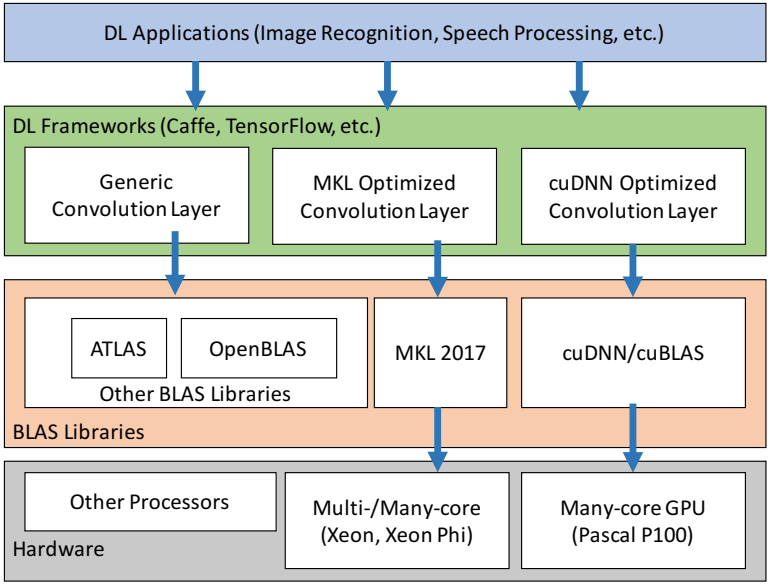
- Single Node
  - Computation only - cuDNN, MKL-DNN, etc.
  - Communication - shared-memory, CUDA IPC, etc.)
- Multiple Nodes
  - Computation - same as single node
  - Communication - MPI, NCCL, Gloo, etc.
- **Studies Covered in today's talk**

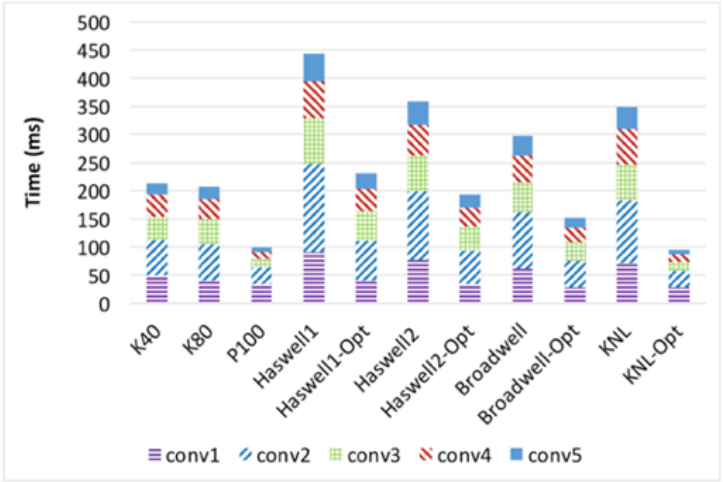


1. CPU vs. GPU comparison is usually unfair -- MLHPC '17
2. Different approaches, different end-to-end performance for same DL framework -- CCGrid '19
3. Communication in DL is not the same as Communication in HPC -- HotI '19, IEEE Micro '19
4. Beyond Data-Parallelism -- HyPar-Flow (accepted to be presented at ISC '20)

1. Caffe: CPUs vs. GPUs

- Caffe – the first framework NVIDIA optimized using cuDNN
  - Everyone has optimized it ever since!
- Holistic View of Performance is needed!





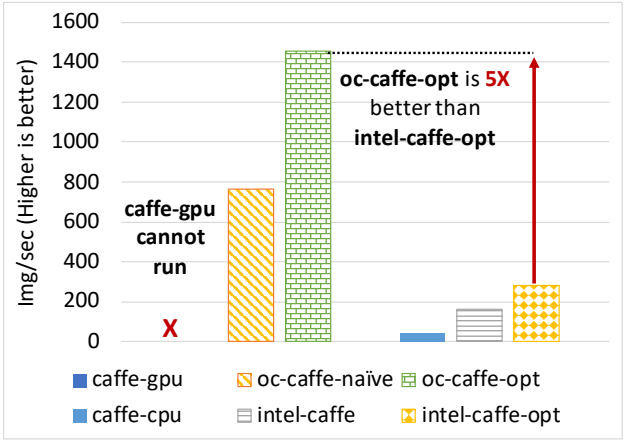
(a) AlexNet: Forward Propagation

- *Faster Convolutions → Faster Training*
- Performance of Intel KNL == NVIDIA P100 for AlexNet
- *Volta; different league!*

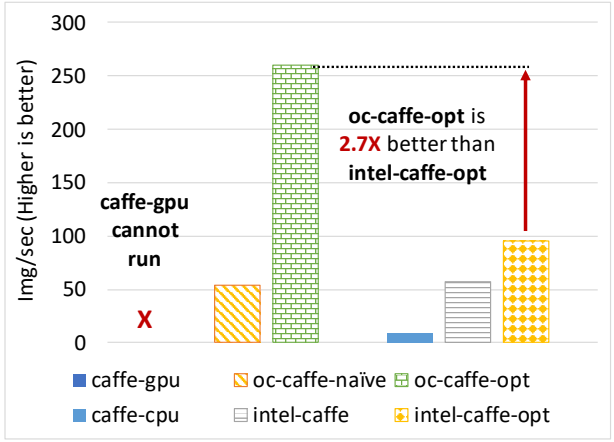
A. A. Awan, H. Subramoni, and Dhabaleswar K. Panda. "An In-depth Performance Characterization of CPU- and GPU-based DNN Training on Modern Architectures", In Proceedings of the Machine Learning on HPC Environments (MLHPC'17), in conjunction with SC '17, Denver, CO

OC-Caffe: GPU (Unified Memory) vs. CPU

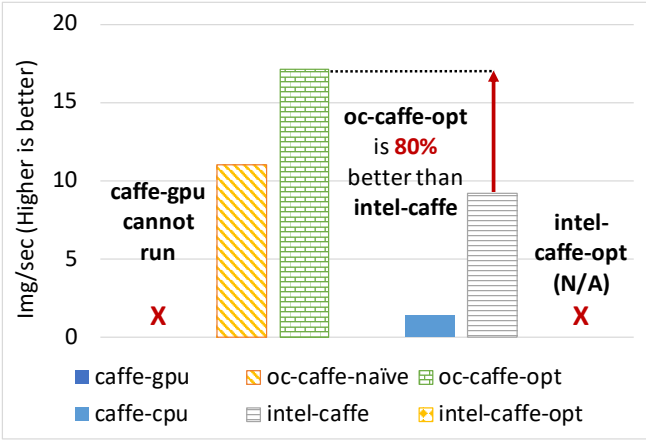
Out-of-Core AlexNet



Out-of-Core GoogLeNet



Out-of-Core ResNet-50

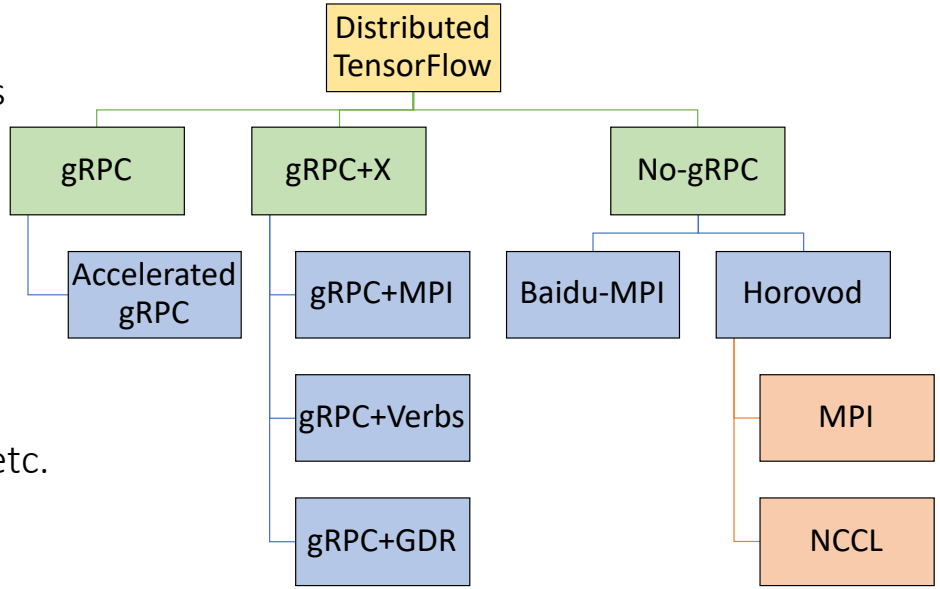


- Out-of-Core workloads – no good baseline to compare
  - Easiest fallback is to use CPU –> A lot more CPU memory available than GPU memory
- OC-Caffe-Optimized (Opt) designs provide much better than CPU/Optimized CPU designs!
  - DNN depth is the major cause for slow-downs → significantly more intra-GPU communication

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

2. Same Framework, Different Communication Approaches

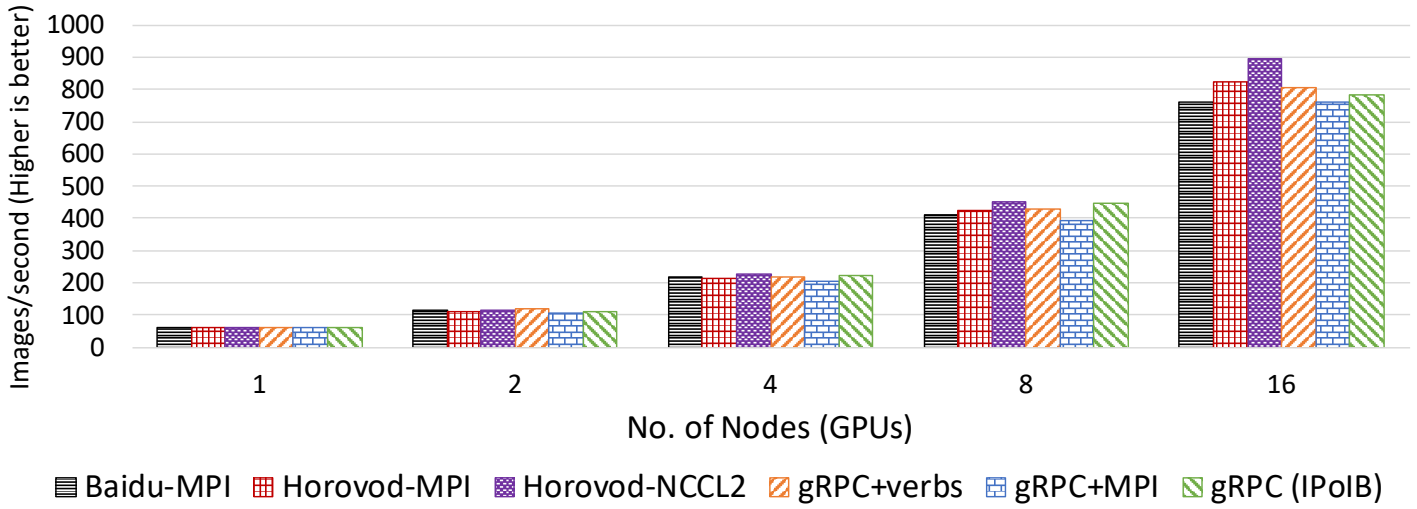
- gRPC (official support)
  - Open-source – can be enhanced by others
  - Accelerated gRPC (add RDMA to gRPC)
- gRPC+X
  - Use gRPC for bootstrap and rendezvous
  - *Actual communication is in "X"*
  - X→ MPI, Verbs, GPUDirect RDMA (GDR), etc.
- No-gRPC
  - Baidu – the first one to use MPI Collectives 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

Performance Characterization of Distributed TensorFlow

- gRPC and gRPC+X designs are slower than No-gRPC designs
- Baidu design (ring-allreduce) still slower than Horovod-NCCL and gRPC+'X'
- Horovod-MPI is about 10% slower than Horovod-NCCL2.



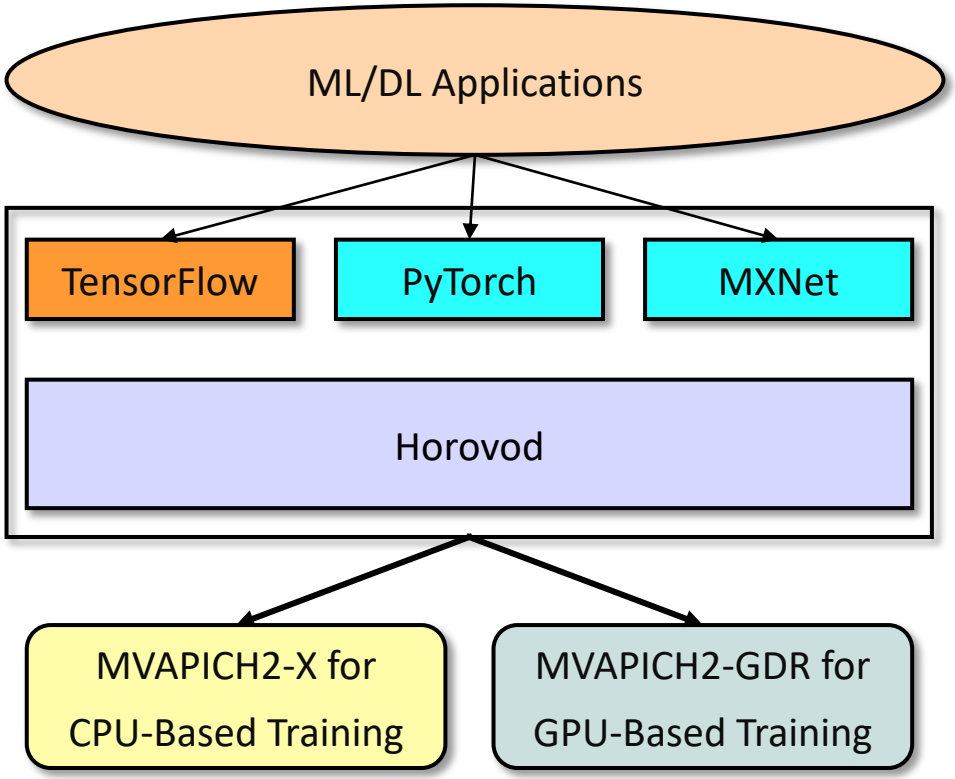
A. A. Awan, J. Bedorf, C.-H. Chu, H. Subramoni and D. K. Panda, "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs, and Performance Evaluation", CCGrid '19. https://arxiv.org/abs/1810.11112

Overview of the MVAPICH2 Project

- High Performance open-source MPI Library
- Support for multiple interconnects
  - InfiniBand, Omni-Path, Ethernet/iWARP, RDMA over Converged Ethernet (RoCE), and AWS EFA
- Support for multiple platforms
  - x86, OpenPOWER, ARM, Xeon-Phi, GPGPUs
- Started in 2001, first open-source version demonstrated at SC '02
- Supports the latest MPI-3.1 standard
- http://mvapich.cse.ohio-state.edu
- Additional optimized versions for different systems/environments:
  - MVAPICH2-X (Advanced MPI + PGAS), since 2011
  - MVAPICH2-GDR with support for NVIDIA GPGPUs, since 2014
  - MVAPICH2-MIC with support for Intel Xeon-Phi, since 2014
  - MVAPICH2-Virt with virtualization support, since 2015
  - MVAPICH2-EA with support for Energy-Awareness, since 2015
  - MVAPICH2-Azure for Azure HPC IB instances, since 2019
  - MVAPICH2-X-AWS for AWS HPC+EFA instances, since 2019
- Tools:
  - OSU MPI Micro-Benchmarks (OMB), since 2003
  - OSU InfiniBand Network Analysis and Monitoring (INAM), since 2015

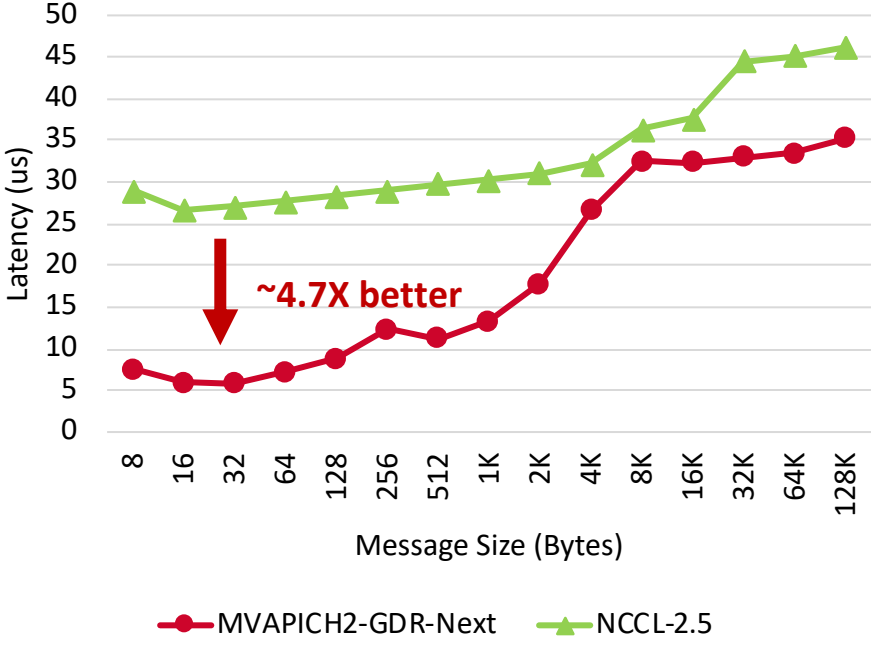
- **Used by more than 3,075 organizations in 89 countries**
- **More than 694,000 (> 0.6 million) downloads from the OSU site directly**
- Empowering many TOP500 clusters (Nov '19 ranking)
  - 3rd, 10,649,600-core (Sunway TaihuLight) at NSC, Wuxi, China
  - 5th, 448, 448 cores (Frontera) at TACC
  - 8th, 391,680 cores (ABCI) in Japan
  - 14th, 570,020 cores (Nurion) in South Korea and many others
- Available with software stacks of many vendors and Linux Distros (RedHat, SuSE, OpenHPC, and Spack)
- Partner in the 5th ranked TACC Frontera system
- **Empowering Top500 systems for more than 15 years**

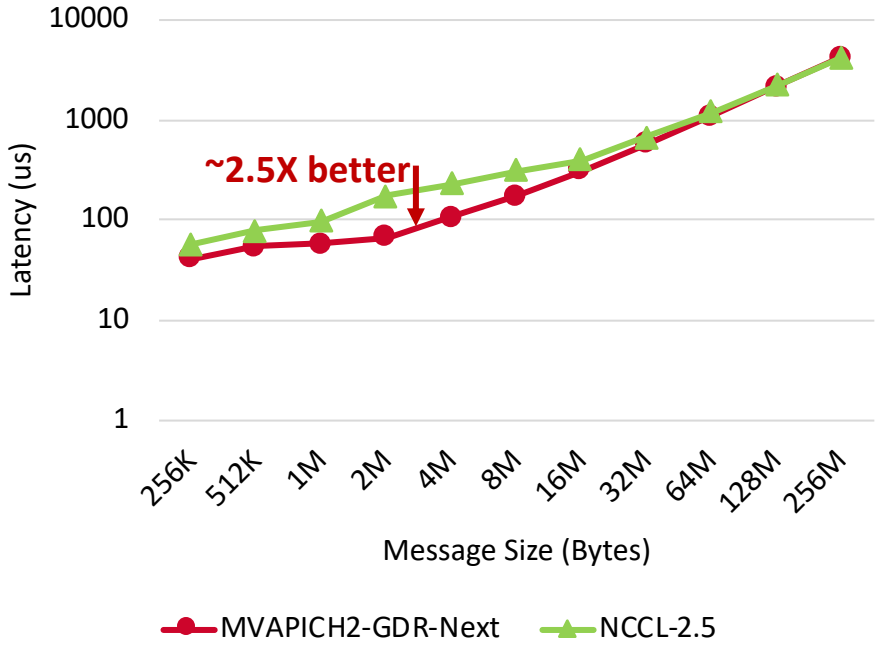
MVAPICH2 (MPI)-driven Infrastructure for ML/DL Training



Communication Benchmark: MV2-GDR vs. NCCL2 – Allreduce (DGX-2)

- **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)**

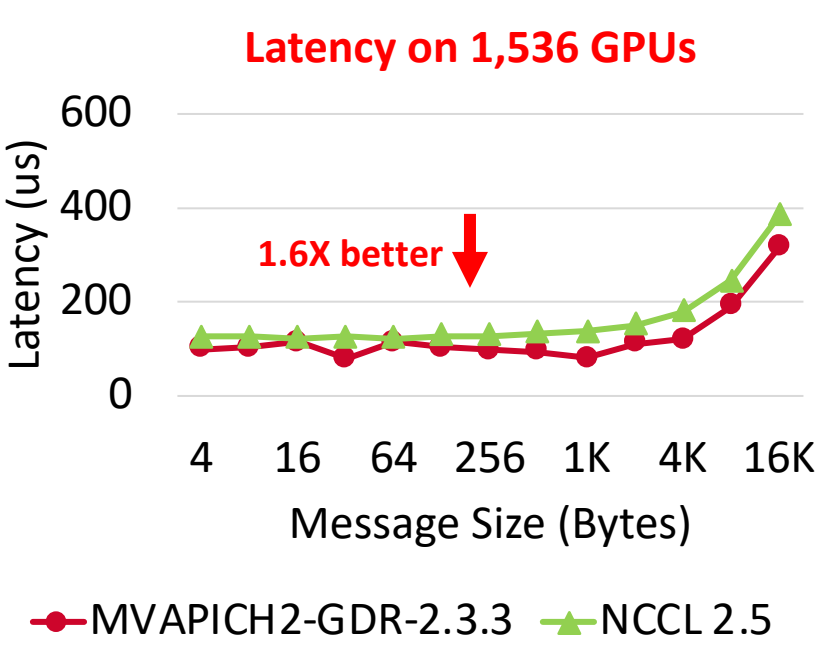


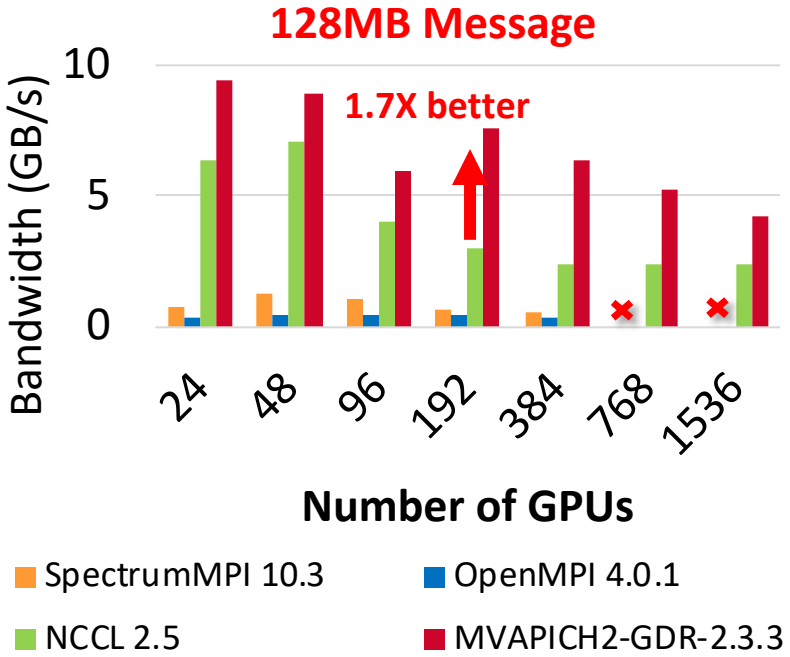


Platform: Nvidia DGX-2 system (16 Nvidia Volta GPUs connected with NVSwitch), CUDA 9.2

Communication Benchmark at Scale

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

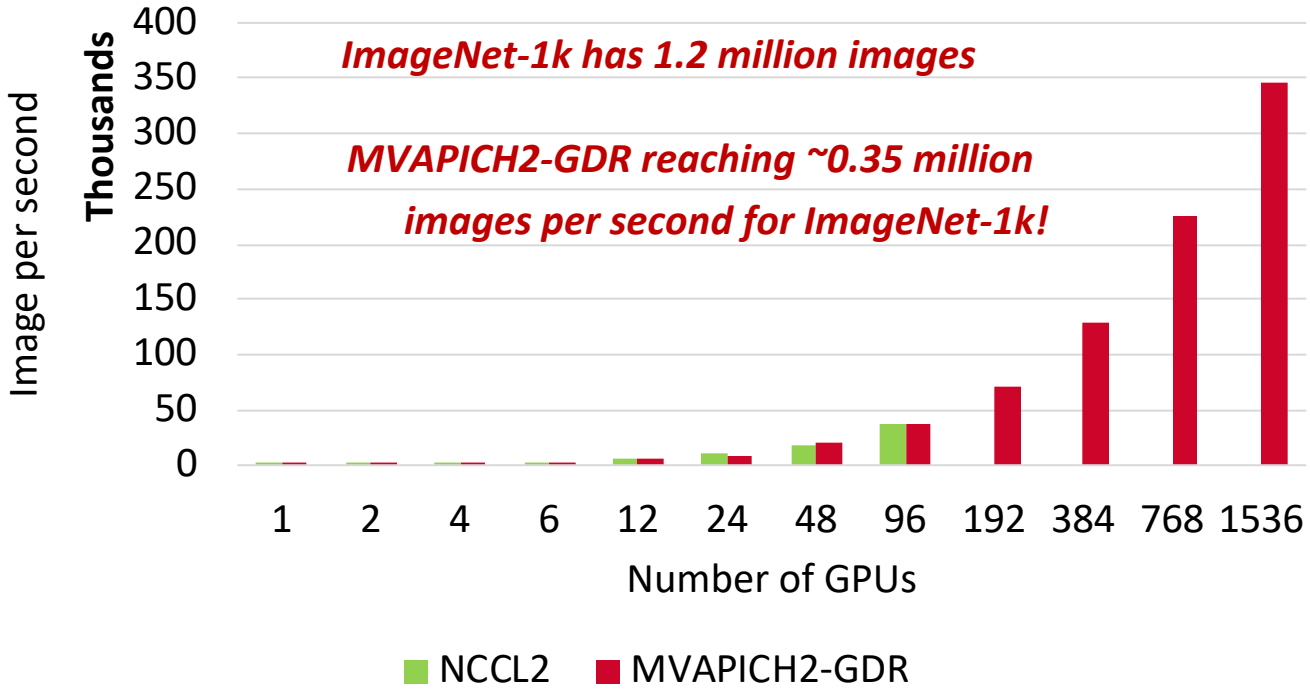




Platform: Dual-socket IBM POWER9 CPU, 6 NVIDIA Volta V100 GPUs, and 2-port InfiniBand EDR Interconnect

End-to-end Performance Benchmark for TF on Summit

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

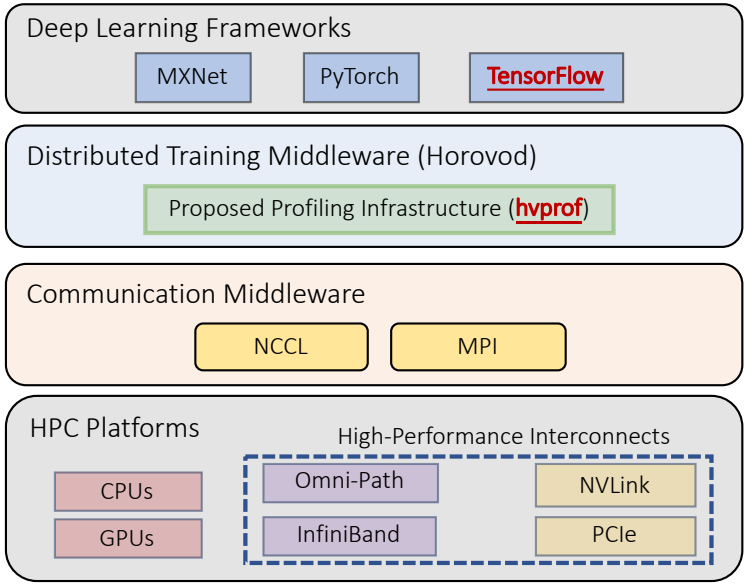


*We observed errors for NCCL2 beyond 96 GPUs

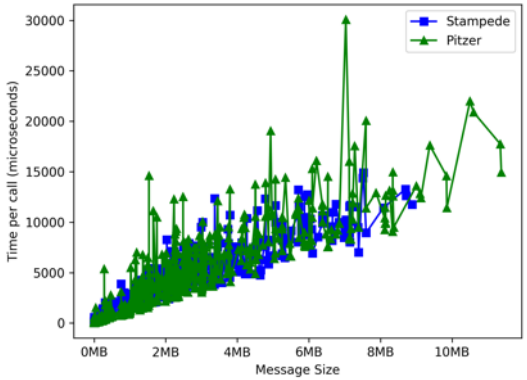
Platform: The Summit Supercomputer (#1 on Top500.org) – 6 NVIDIA Volta GPUs per node connected with NVLink, CUDA 9.2

3. Communication in DL vs. HPC

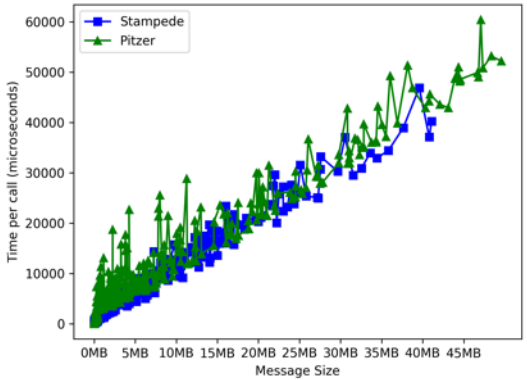
- White-box profiling is needed for complex DL frameworks
- hvprof provides multiple types of valuable metrics for
  - 1) ML/DL developers and 2) Designers of MPI libraries
- Profile of Latency for Allreduce (NVLink, PCIe, IB, Omni-Path)
- *Summary: Non-power of 2 is under-optimized for all libraries!*

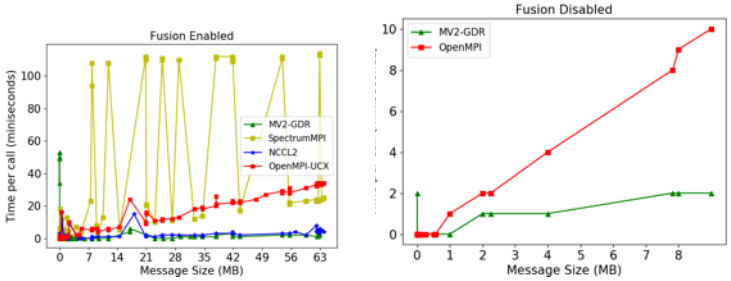


Inception-v4– Intel MPI



ResNet-101– MVAPICH2

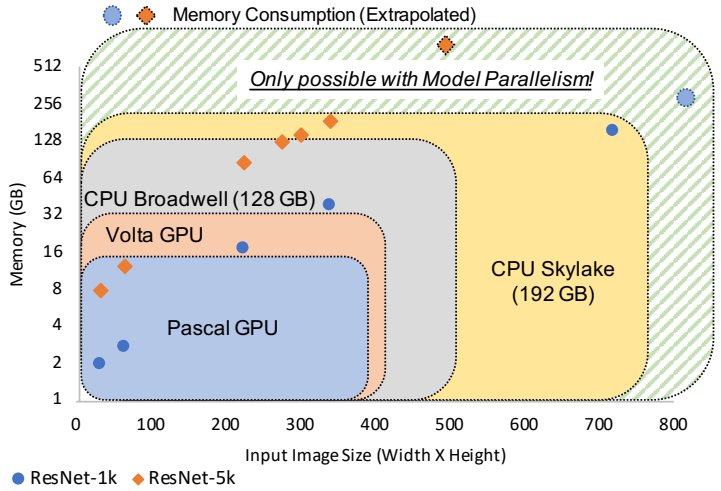




Awan et al., "Communication Profiling and Characterization of Deep Learning Workloads on Clusters with High-Performance Interconnects", IEEE Micro (Magazine) '20, Hot Interconnects '19.

4. Beyond Data Parallelism in DNNs

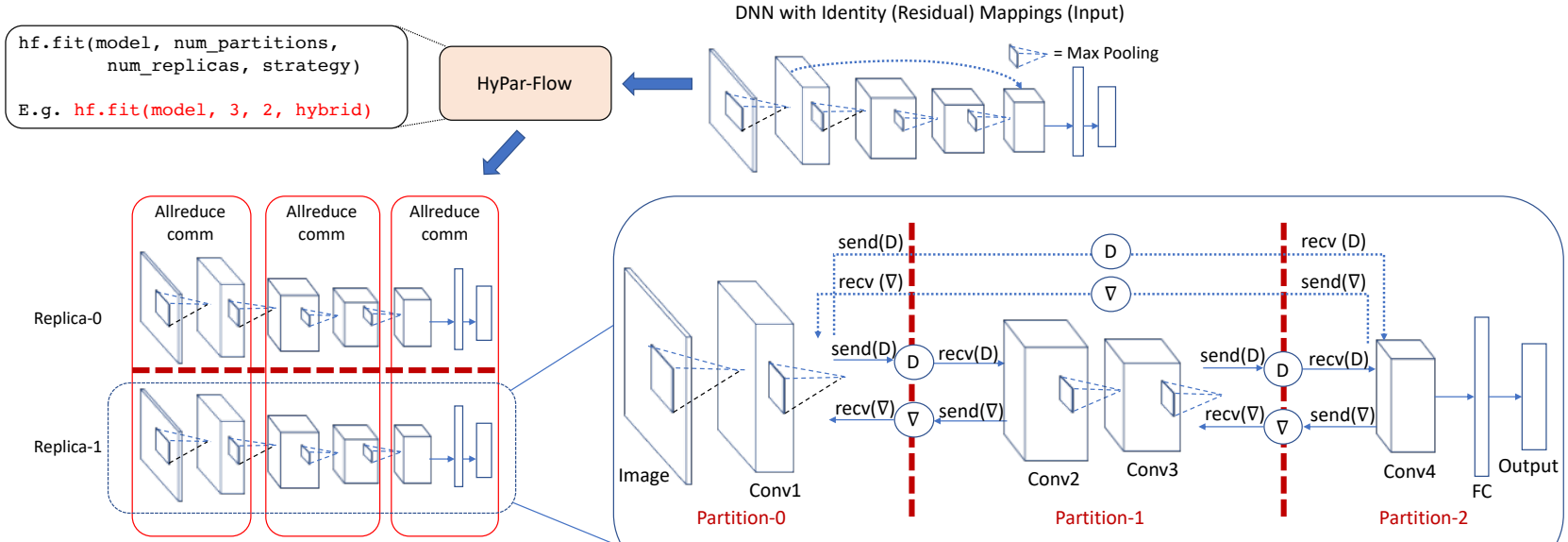
- Data-Parallelism– only for models that fit the memory
- Out-of-core models
  - Deeper model → Better accuracy but more memory required!
- *Model parallelism can work for out-of-core models!*
- Key Challenges
  - Model Partitioning is difficult for application programmer
  - Finding the right partition (grain) size is hard
    - *cut at which layer and why*?
  - Developing a practical system for model-parallelism
    - Redesign DL Framework or create additional layers?
    - Existing Communication middleware or extensions needed?



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", ISC '20 (accepted to be presented), https://arxiv.org/pdf/1911.05146.pdf

Model and Hybrid Parallelism

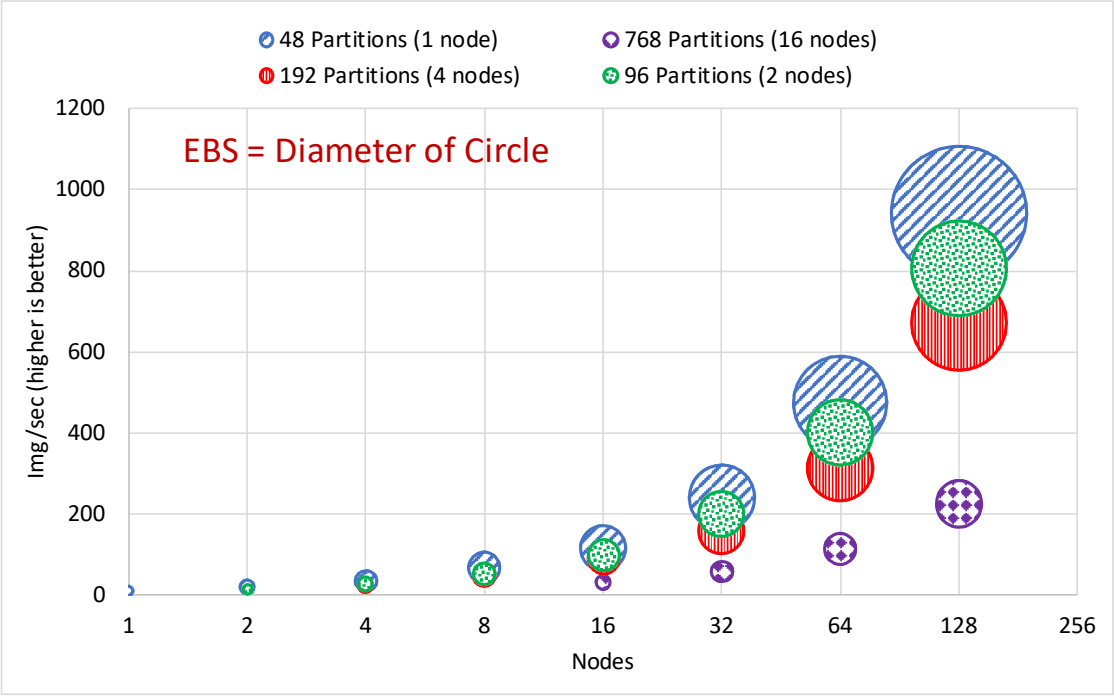
- HyPar-Flow is practical (easy-to-use) and high-performance (uses MPI)
  - Based on Keras models and exploits TF 2.0 Eager Execution
  - Leverages performance of MPI pt-to-pt. and collectives for communication



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", ISC '20 (accepted to be presented), https://arxiv.org/pdf/1911.05146.pdf

Benchmarking HyPar-Flow in Different Configurations

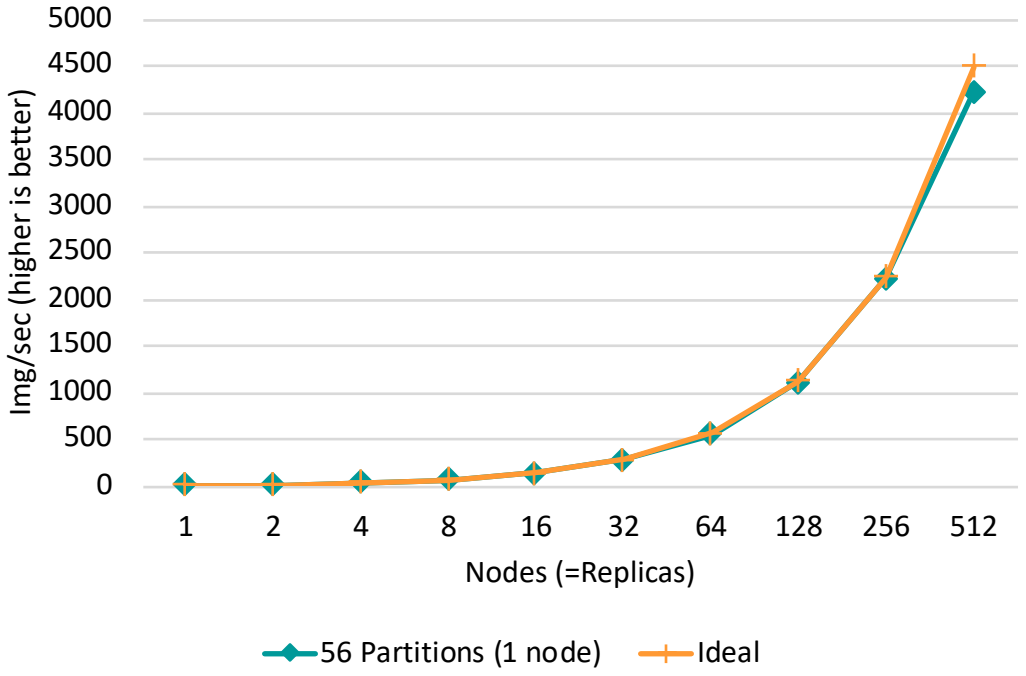
- CPU based results
  - AMD EPYC
  - Intel Xeon
- Excellent speedups for
  - VGG-19
  - ResNet-110
  - ResNet-1000 (1k layers)
- Able to train "future" models
  - E.g. ResNet-5000 (a synthetic 5000-layer model we benchmarked)



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

End-to-end Performance at Scale (512 nodes on Frontera)

- ResNet-1001 with variable batch size
- Approach:
  - 48 model-partitions for 56 cores
  - 512 model-replicas for 512 nodes
  - Total cores: 48 x 512 = 24,576
- Speedup
  - 253X on 256 nodes
  - 481X on 512 nodes
- Scaling Efficiency
  - 98% up to 256 nodes
  - 93.9% for 512 nodes



481x speedup on 512 Intel Xeon Skylake nodes (TACC Frontera)

Agenda

- Introduction
- Background
- ML/DL Benchmarks
- Solutions and Case Studies
- Conclusion and Future Directions

Future Direction 1: Reproducibility via Open Infrastructures

- Reproducibility is a challenge for HPC and DL
- Challenges
  - No standard and user-friendly way of benchmarking ML/DL models
  - Disconnect between HPC and DL communities
  - Metrics cause confusion b/w communities -- images/second, time to train, latency, bandwidth, etc.
- MLPerf, a good start but mostly for a single-node/GPU
  - Can we extend for HPC systems?
- Deep500 – a meta DL framework to evaluate DL or DL frameworks?
- Other benchmarks – framework specific (tf_cnn_benchmarks) or low-level (latency/bw)

Future Direction 2: Benchmarking Emerging Models and Areas

- Models beyond ResNet(s) – Generative Adversarial Networks, Transformer, BERT, GPT-2, Mini-go, etc.

- Applications beyond Image Classification – Neural Machine Translation, Language Processing, Recommendation Systems, Reinforcement Learning, Neural Code Gen. etc.

Investigate scale-up/scale-out for published models on current systems like Sierra and Summit and upcoming systems El Capitan and Frontier

Conclusion

- Deep Learning is an important area
- Several benchmarking efforts to better understand DL workloads
  - MLPerf, Deep500, DAWNBench, etc.
- DL on single-node systems is complex enough
  - cuDNN, MKL, BLAS, shared-memory communication, CUDA IPC, etc.
- DL on HPC systems is even more challenging
  - All of the single node + MPI, NCCL, Gloo, etc.
- Several design choices and benchmarking studies
- No single benchmark can cover all aspects!

Thank You!

awan.10@osu.edu

panda@cse.ohio-state.edu

https://awan-10.github.io

Questions?

The High-Performance Deep Learning Project
http://hidl.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/

Please join us here from 1pm - 5pm

- Tutorial on High Performance Distributed Deep Learning
- Several topics on Distributed DL Trends and Designs
- Includes a Hands-on section on Distributed TensorFlow
- Speakers: DK Panda, Ammar Awan, and Hari Subramoni