Scalable and Distributed Deep Learning (DL): Co-Design MPI Runtimes and DL Frameworks

OSU Booth Talk (SC '18)

Ammar Ahmad Awan, Hari Subramoni, and Dhabaleswar K. Panda

Network Based Computing Laboratory

Dept. of Computer Science and Engineering

The Ohio State University

Agenda

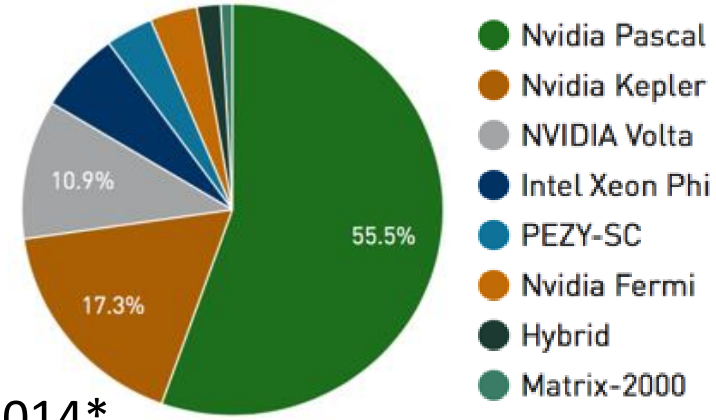
- **Introduction**
  - **Deep Learning Trends**
  - **CPUs and GPUs for Deep Learning**
  - **Message Passing Interface (MPI)**
- Research Challenges: Exploiting HPC for Deep Learning
- Proposed Solutions
- Conclusion

Deep Learning Frameworks

- Easily implement and experiment with Deep Neural Networks
  - Several Deep Learning (DL) frameworks have emerged
- Caffe, Microsoft Cognitive Toolkit (CNTK), TensorFlow, PyTorch, and counting....
  - *Focus on CUDA-Aware MPI based DL frameworks*
- Most frameworks have been optimized for NVIDIA GPUs and the CUDA programming model
  - However, distributed training (MPI+CUDA) is still emerging
  - Fragmentation in efforts also exists – gRPC, MPI, NCCL, Gloo, etc.

Deep Learning and GPUs

- *NVIDIA GPUs - main driving force for faster training of Deep Neural Networks (DNNs)*



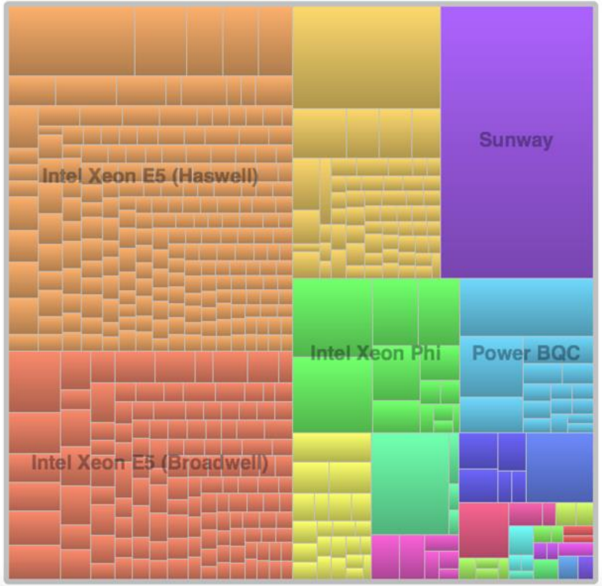
- The ImageNet Challenge - (ILSVRC)
  - DL models like AlexNet, ResNet, and VGG
  - 90% of the ImageNet teams used GPUs in 2014*
  - GPUs: A natural fit for DL –throughput-oriented (dense-compute)
  - *And, GPUs are growing in the HPC arena as well! – Top500 (Jun '18)*

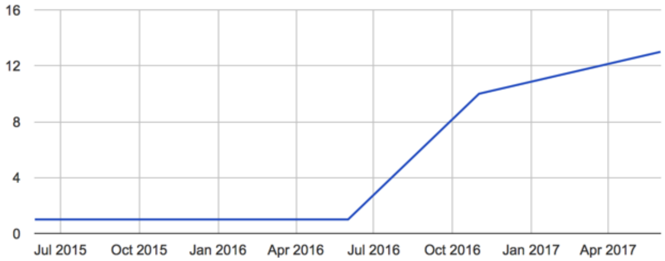
https://www.top500.org/

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

And CPUs are catching up fast

https://www.top500.org/statistics/list/





System Count for Xeon Phi

- Intel CPUs are everywhere and many-core CPUs are emerging according to Top500.org
- Host CPUs exist even on the GPU nodes
  - Many-core Xeon Phis are increasing
- Usually, we hear CPUs are ***10x – 100x*** slower than GPUs? [1-3]
  - But, CPU-based ML/DL is getting attention and performance has significantly improved now

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

What to use for scale-out? (Distributed training of Neural Nets.)

- What is Message Passing Interface (**MPI**)?
  - a de-facto standard for expressing distributed-memory parallel programming
  - used for communication between processes in multi-process applications
- ***MVAPICH2 is a high performance implementation of the MPI standard***
- **What can MPI do for Deep Learning?**
  - MPI has been used for large scale scientific applications
  - Deep Learning can also exploit MPI to perform high-performance communication
- **Why do I need communication in Deep Learning?**
  - If you use one GPU or one CPU, you do not need communication
  - But, one GPU or CPU is not enough!
  - DL wants as many compute elements as it can get!
  - *MPI is a great fit – Broadcast, Reduce, and Allreduce is what most DL workloads require*

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 2,925 organizations in 86 countries**
  - **More than 489,000 (> 0.48 million) downloads from the OSU site direct**
  - Empowering many TOP500 clusters (Jul '18 ranking)
    - 2^{nd} ranked 10,649,640-core cluster (Sunway TaihuLight) at NSC, Wuxi, Chir
    - 12^{th}, 556,104 cores (Oakforest-PACS) in Japan
    - 15^{th}, 367,024 cores (Stampede2) at TACC
    - 24^{th}, 241,108-core (Pleiades) at NASA and many others
  - Available with software stacks of many vendors and Linux Distros (RedHat and SuSE)
  - **http://mvapich.cse.ohio-state.edu**
- Empowering Top500 systems for over a decade

Deep Learning Frameworks – CPUs or GPUs?

- There are several Deep Learning (DL) or DNN Training frameworks
- Every (almost every) framework has been optimized for NVIDIA GPUs
  - cuBLAS and cuDNN have led to significant performance gains!
- But every framework is able to execute on a CPU as well
  - So why are we not using them?
  - Performance has been "terrible" and several studies have reported significant degradation when using CPUs (see nvidia.qwiklab.com)
- But there is hope, actually a lot of great progress here!
  - And MKL-DNN, just like cuDNN, has definitely rekindled this!!
  - The landscape for CPU-based DL looks promising..

Agenda

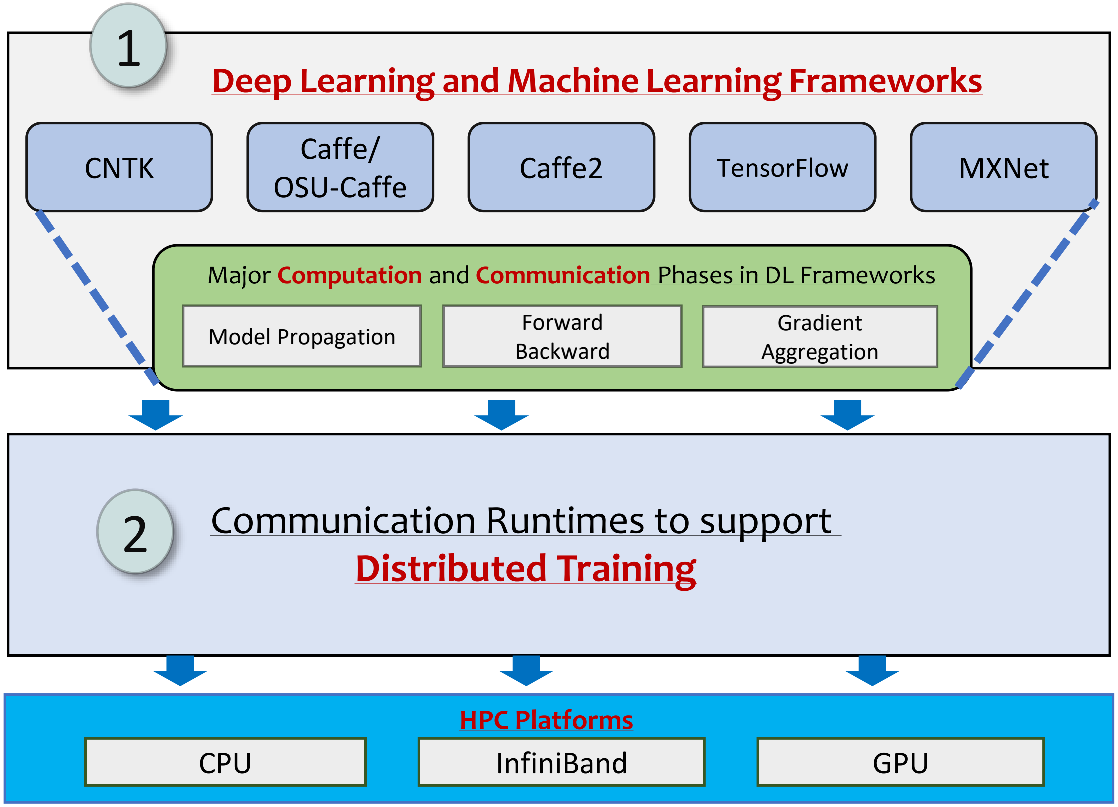
- Introduction
- **Research Challenges: Exploiting HPC for Deep Learning**
- Proposed Solutions
- Conclusion

The Key Question!

How to efficiently exploit heterogeneous High Performance Computing (HPC) resources for high-performance and high-productivity Deep Learning?

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)

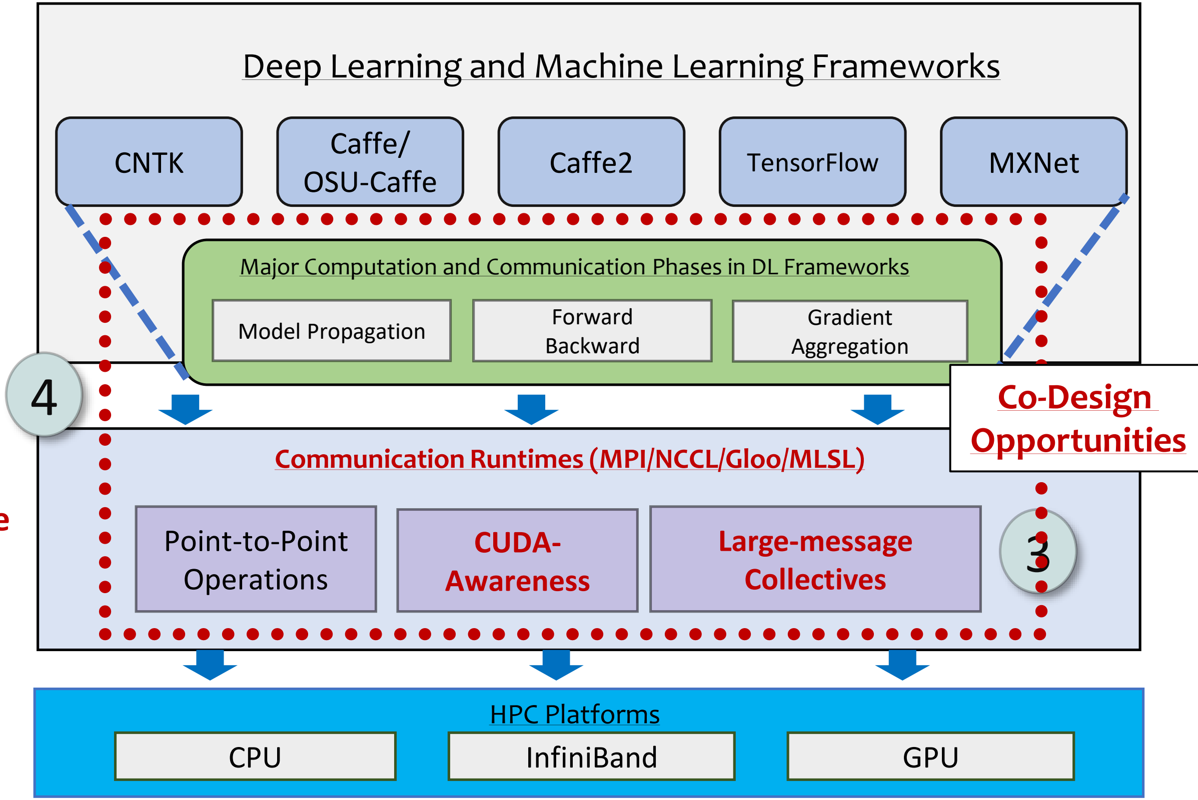
3. What are the **new design challenges** brought forward by DL frameworks for Communication runtimes?

- 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?

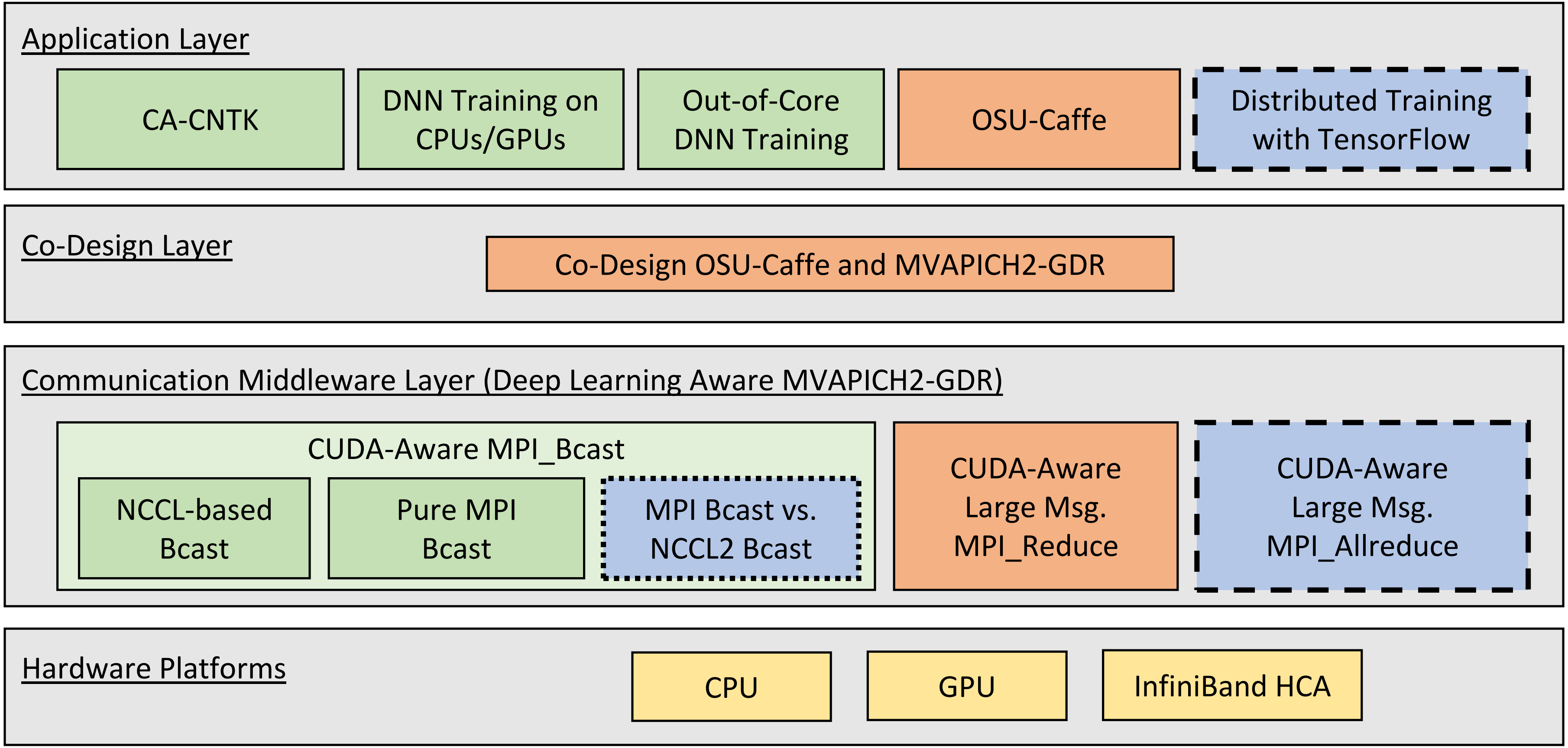
5.



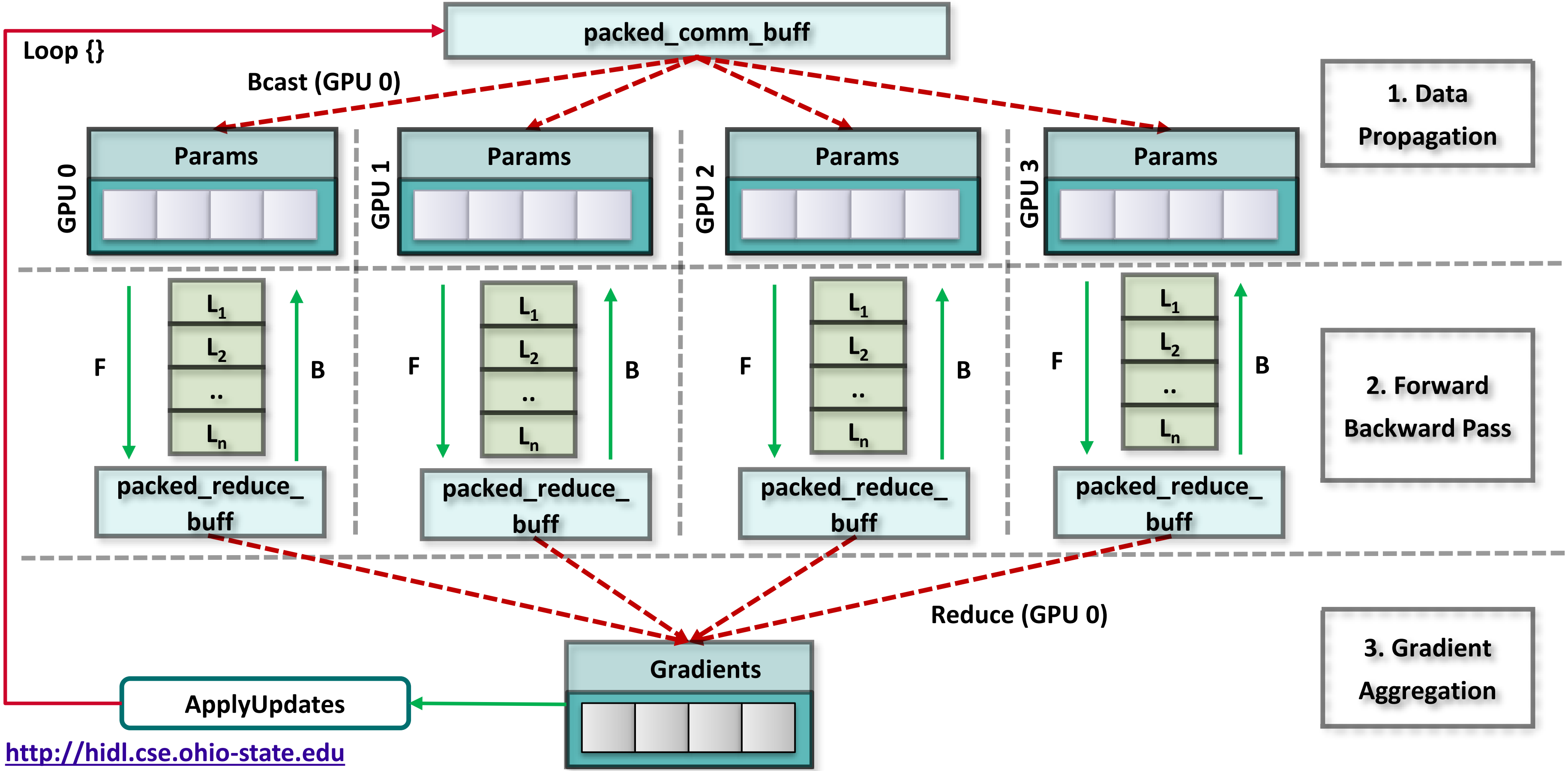
Agenda

- Introduction
- Research Challenges: Exploiting HPC for Deep Learning
- **Proposed Solutions**
- Conclusion

Overview of the Proposed Solutions



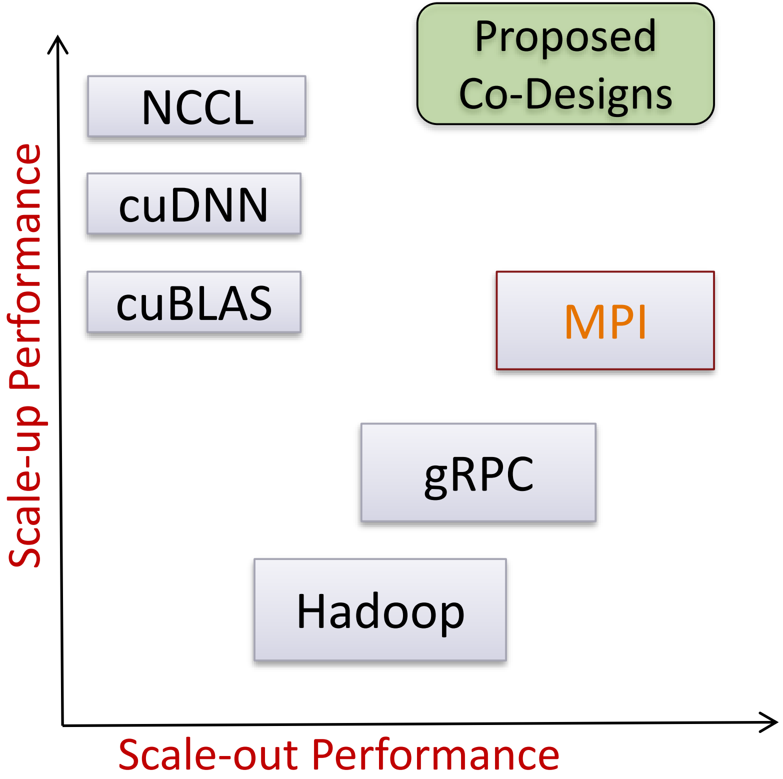
Caffe Architecture



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

OSU-Caffe: Co-design to Tackle New Challenges for MPI Runtimes

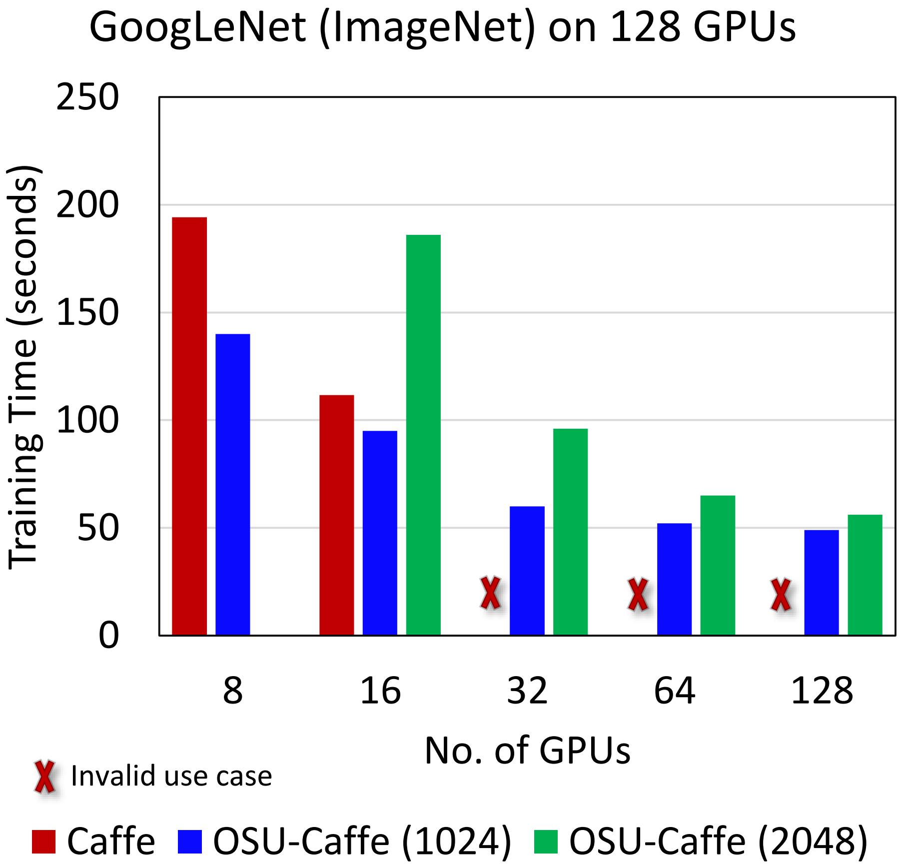
- Deep Learning frameworks are a different game
  - Unusually large message sizes (order of megabytes)
  - Most communication based on GPU buffers
- Existing State-of-the-art
  - cuDNN, cuBLAS, NCCL --> **scale-up** performance
  - CUDA-Aware MPI --> **scale-out** performance
- Proposed: Can we **co-design** the MPI runtime (**MVAPICH2-GDR**) and the DL framework (**Caffe**) to achieve both?
  - Efficient **Overlap** of Computation and Communication
  - Efficient **Large-Message** Communication (Reductions)
  - What **application co-designs** are needed to exploit **communication-runtime co-designs**?



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

OSU-Caffe 0.9: Scalable Deep Learning on GPU Clusters

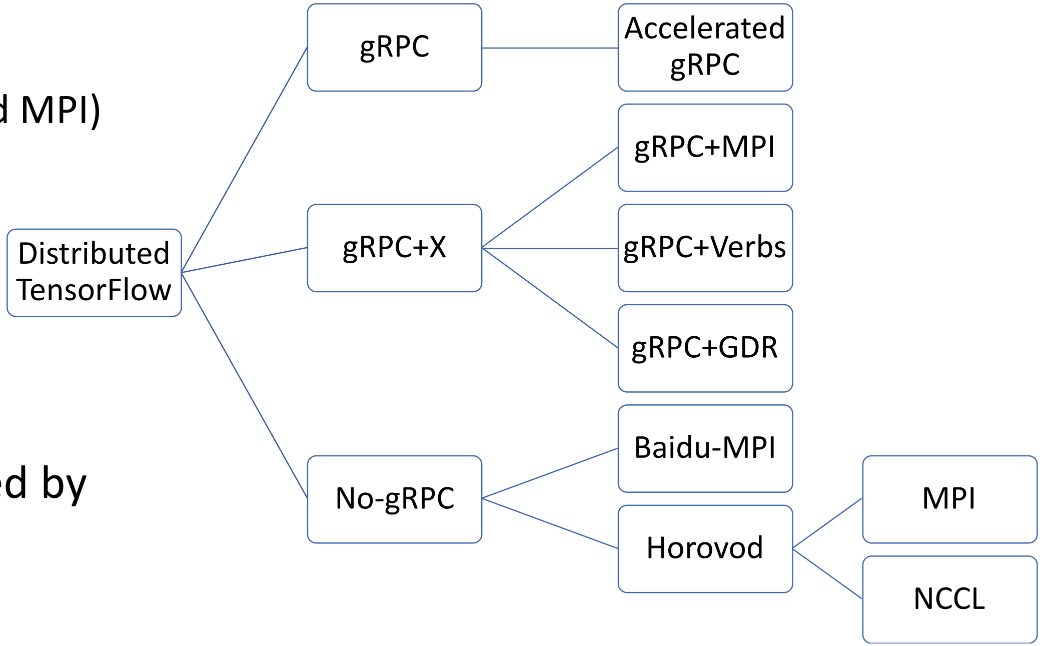
- Caffe : A flexible and layered Deep Learning framework.
- Benefits and Weaknesses
  - Multi-GPU Training within a single node
  - Performance degradation for GPUs across different sockets
  - Limited Scale-out
- OSU-Caffe: MPI-based Parallel Training
  - Enable Scale-up (within a node) and Scale-out (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 0.9 available from HiDL site

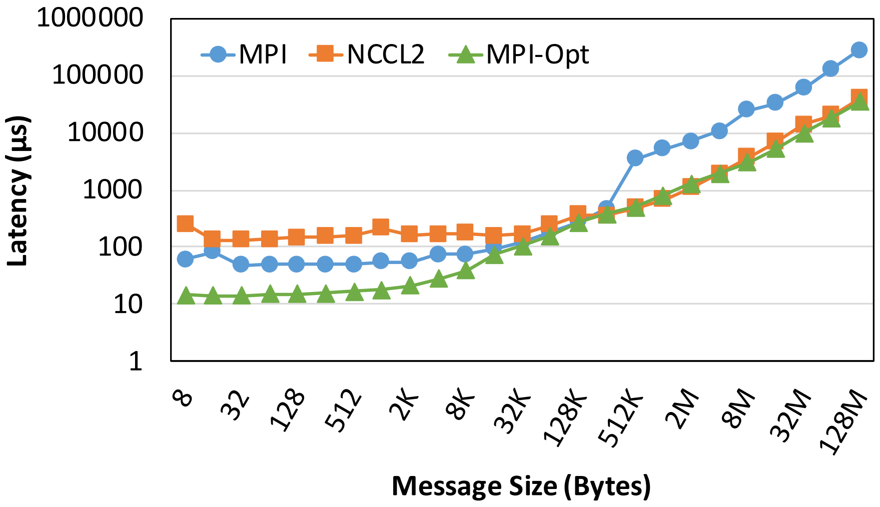
Scalable Distributed DNN Training using TensorFlow and MPI

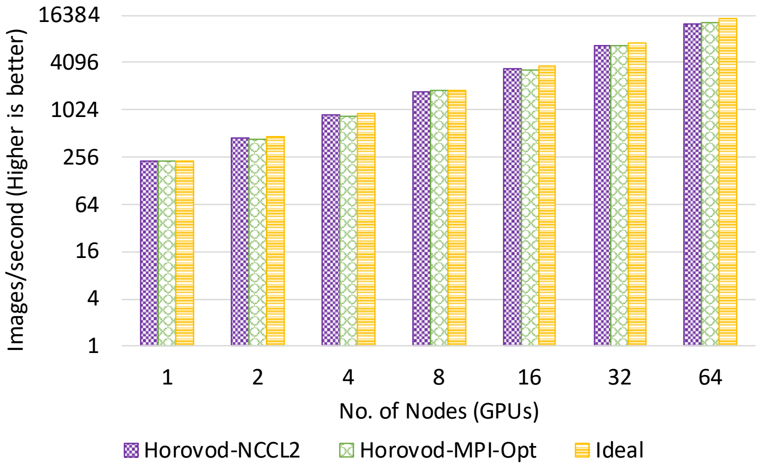
- Several Approaches to Distributed Training
  - Google RPC (gRPC)
  - gRPC+X (X= Verbs API, GDR, and MPI)
  - No-gRPC
- No-gRPC designs use:
  - MPI or
  - NCCL
- Performance is heavily influenced by
  - MPI_Allreduce



A. A. Awan et al. "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs and Performance Evaluation", Submitted to IPDPS-19 for peer-review, Available from: https://arxiv.org/abs/1810.11112

TensorFlow with CUDA-Aware MPI: NCCL vs. MVAPICH2-GDR





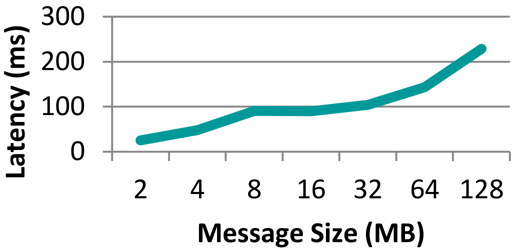
Faster Allreduce in the proposed MPI-Opt implemented in MVAPICH2-GDR –> Faster (near-ideal) DNN Training speed-ups in TensorFlow-Horovod

A. A. Awan et al. "Scalable Distributed DNN Training using TensorFlow and CUDA-Aware MPI: Characterization, Designs and Performance Evaluation", Submitted to IPDPS-19 for peer-review, Available from: https://arxiv.org/abs/1810.11112

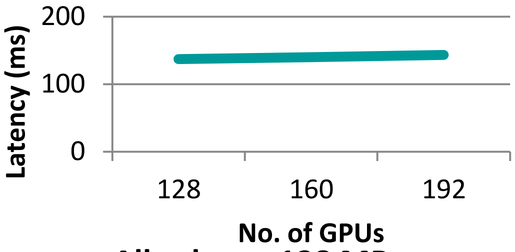
Large Message Optimized Collectives for Deep Learning

- MVAPICH2-GDR provides optimized collectives for **large message sizes**
- Optimized Reduce, Allreduce, and Bcast
- **Good scaling with large number of GPUs**
- **Available in MVAPICH2-GDR 2.2 and higher**

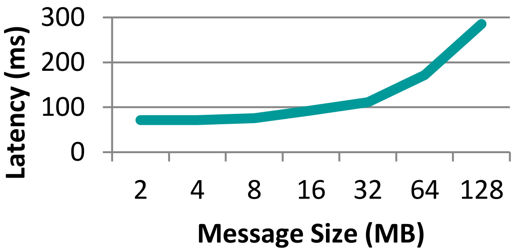
Reduce – 192 GPUs



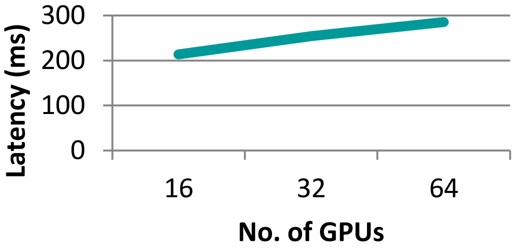
Reduce – 64 MB



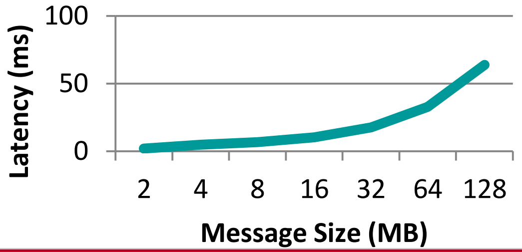
Allreduce – 64 GPUs



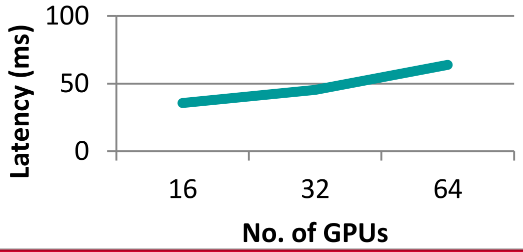
Allreduce - 128 MB



Bcast – 64 GPUs

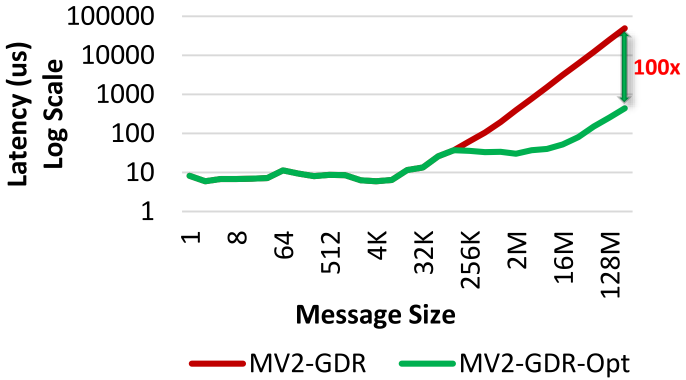


Bcast 128 MB

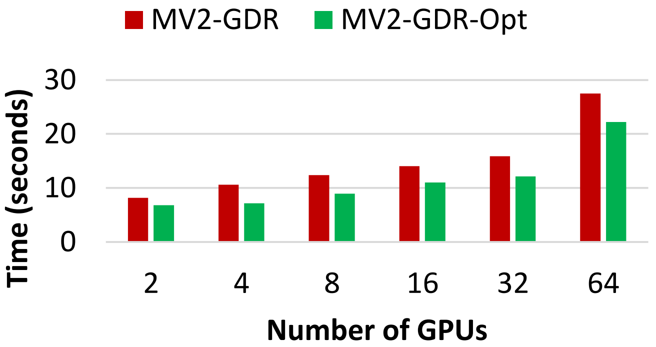


Efficient Broadcast for MVAPICH2-GDR using NVIDIA NCCL

- NCCL has some limitations
  - Only works for a single node, thus, no scale-out on multiple nodes
  - Degradation across IOH (socket) for scale-up (within a node)
- We propose optimized MPI_Bcast
  - Communication of very large GPU buffers (order of megabytes)
  - Scale-out on large number of dense multi-GPU nodes
- Hierarchical Communication that efficiently exploits:
  - CUDA-Aware MPI_Bcast in MV2-GDR
  - NCCL Broadcast primitive



Performance Benefits: OSU Micro-benchmarks

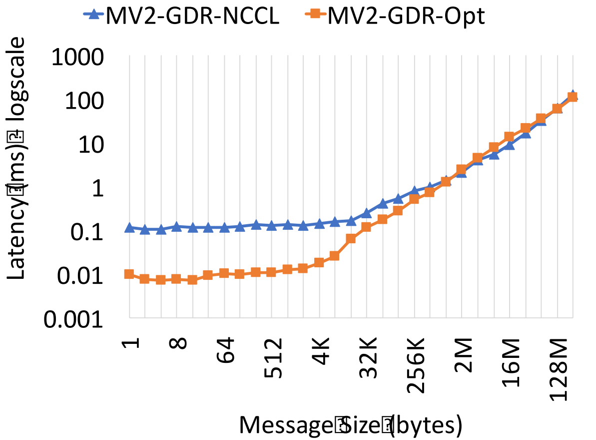


Performance Benefits: Microsoft CNTK DL framework
(25% avg. improvement)

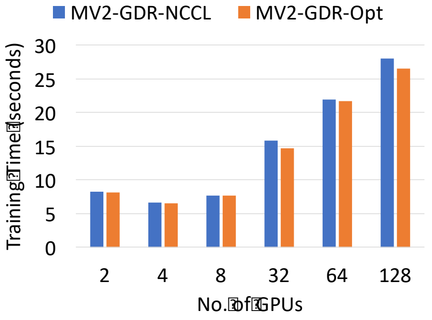
Efficient Large Message Broadcast using NCCL and CUDA-Aware MPI for Deep Learning, A. Awan , K. Hamidouche , A. Venkatesh , and D. K. Panda, EuroMPI 16
[Best Paper Runner-Up]

Pure MPI Large Message Broadcast

- MPI_Bcast: Design and Performance Tuning for DL Workloads
  - Design ring-based algorithms for large messages
  - Harness a multitude of algorithms and techniques for bes performance across the full range of message size and process/GPU count
- Performance Benefits
  - Performance comparable or better than NCCL-augmented approaches for large messages
  - Up to 10X improvement for small/medium message sizes with micro-benchmarks and up to 7% improvement for VGG training



MPI Bcast Benchmark: 128 GPUs (8 nodes)

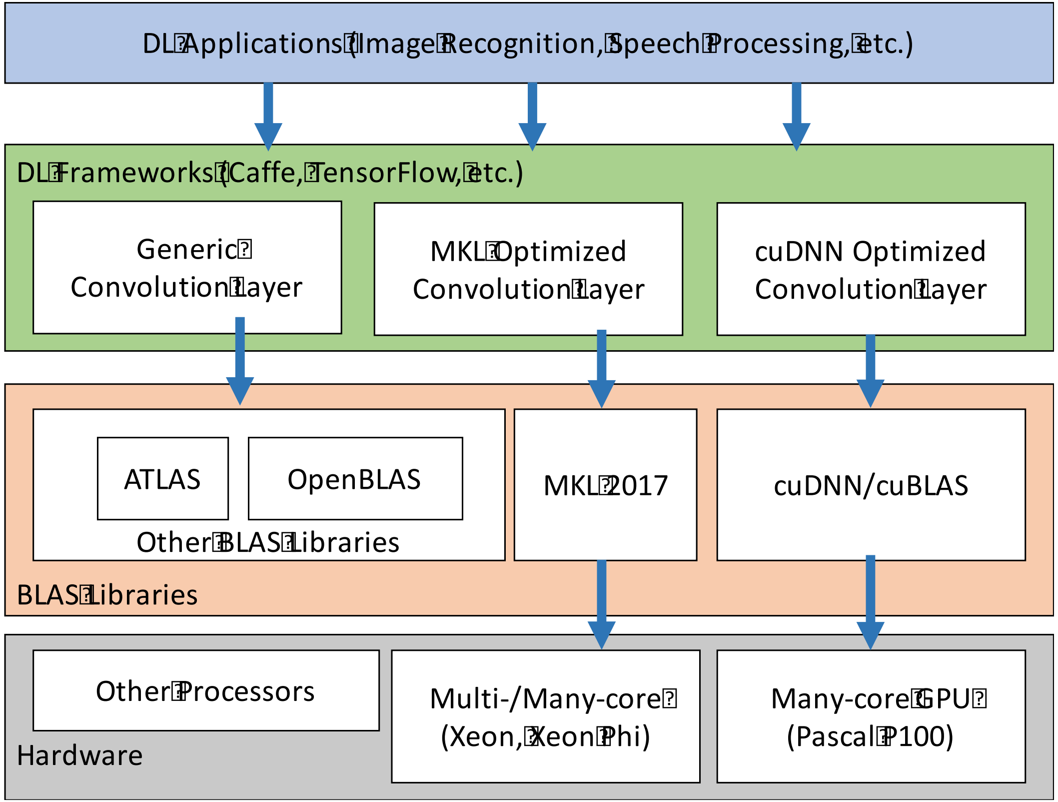


VGG Training with CNTK

A. A. Awan, C-H. Chu, H. Subramoni, and D. K. Panda. Optimized Broadcast for Deep Learning Workloads on Dense-GPU InfiniBand Clusters: MPI or NCCL?, EuroMPI '18

Understanding the Impact of Execution Environments

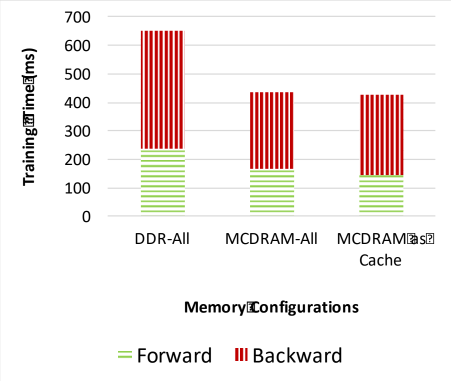
- Performance depends on many factors
- Hardware Architectures
  - GPUs
  - Multi-/Many-core CPUs
  - Software Libraries: cuDNN (for GPUs), MKL-DNN/MKL 2017 (for CPUs)
- Hardware and Software co-design
  - Software libraries optimized for one platform will not help the other!
  - cuDNN vs. MKL-DNN



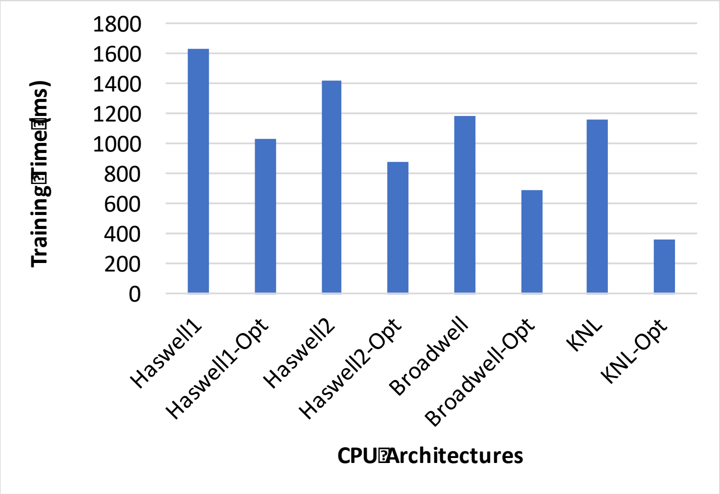
A. A. Awan, H. Subramoni, D. Panda, "An In-depth Performance Characterization of CPU- and GPU-based DNN Training on Modern Architectures" 3rd Workshop on Machine Learning in High Performance Computing Environments, held in conjunction with SC17, Nov 2017.

Impact of MKL engine and MC-DRAM for Intel-Caffe

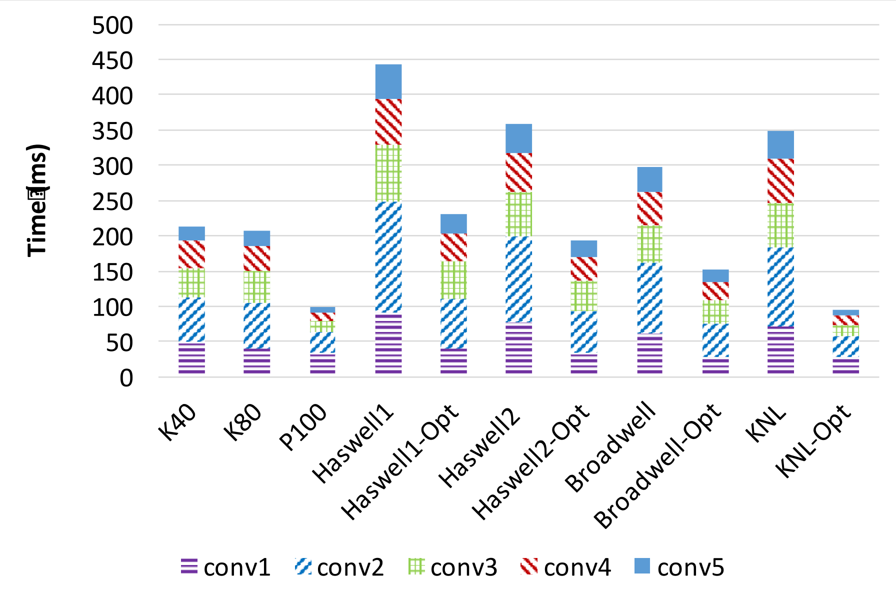
- We use ***MCDRAM as Cache*** for all the subsequent results
- On average, DDR-All is up to ***1.5X slower*** than MCDRAM

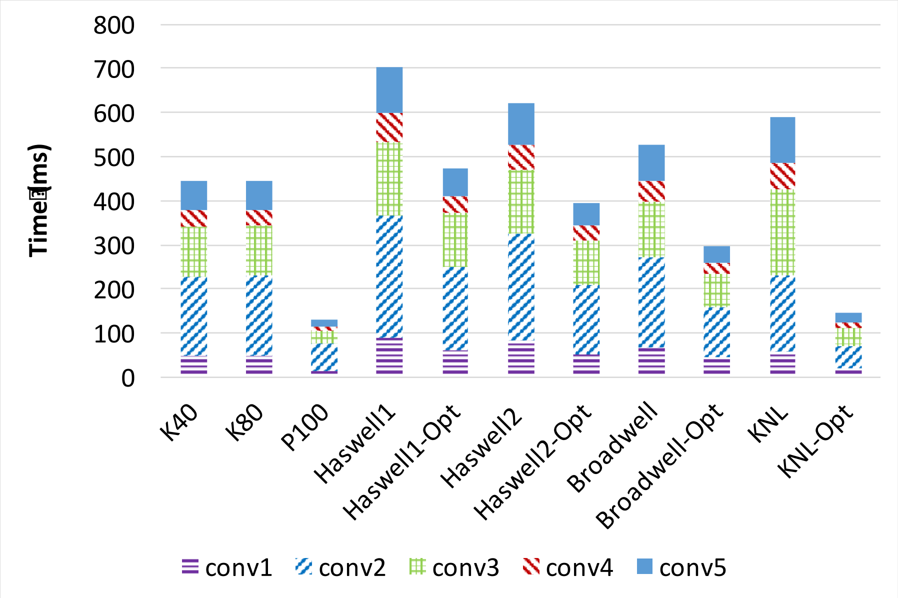


- MKL engine is up to ***3X better*** than default Caffe engine
- ***Biggest*** gains for ***Intel Xeon Phi*** (many-core) architecture
- Both Haswell and Broadwell architectures get significant speedups (***up to 1.5X***)



The Full Landscape for AlexNet Training on CPU/GPU

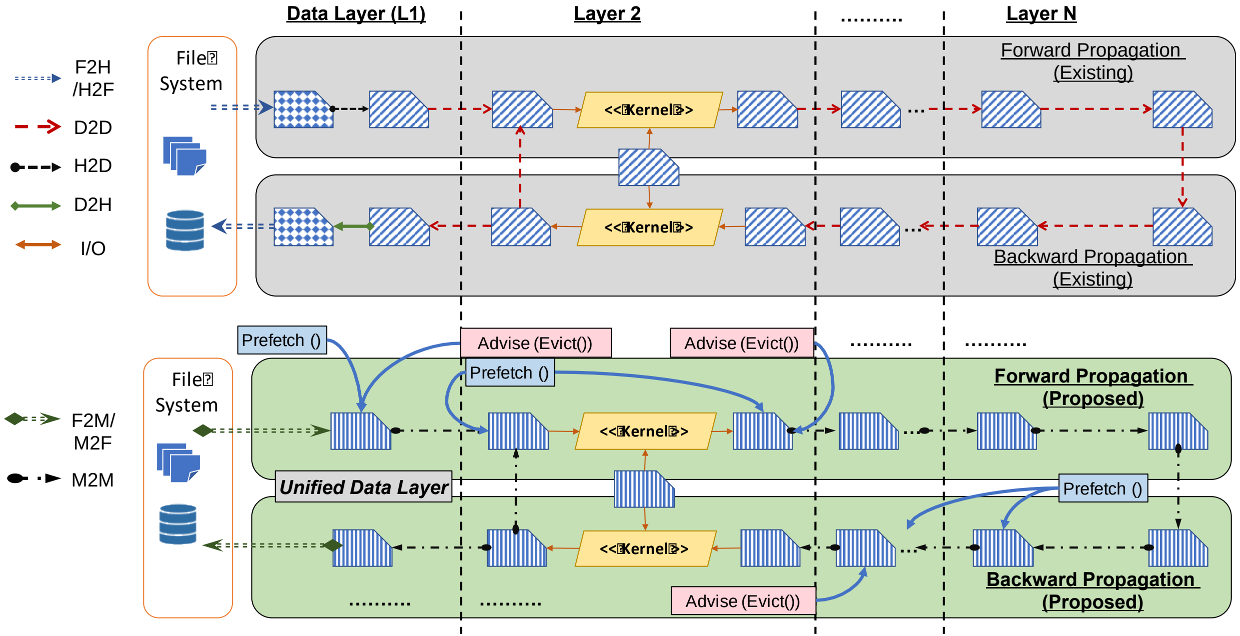




- Convolutions in the Forward and Backward Pass
- ***Faster Convolutions → Faster Training***
- Most performance gains are based on ***conv2*** and ***conv3***.

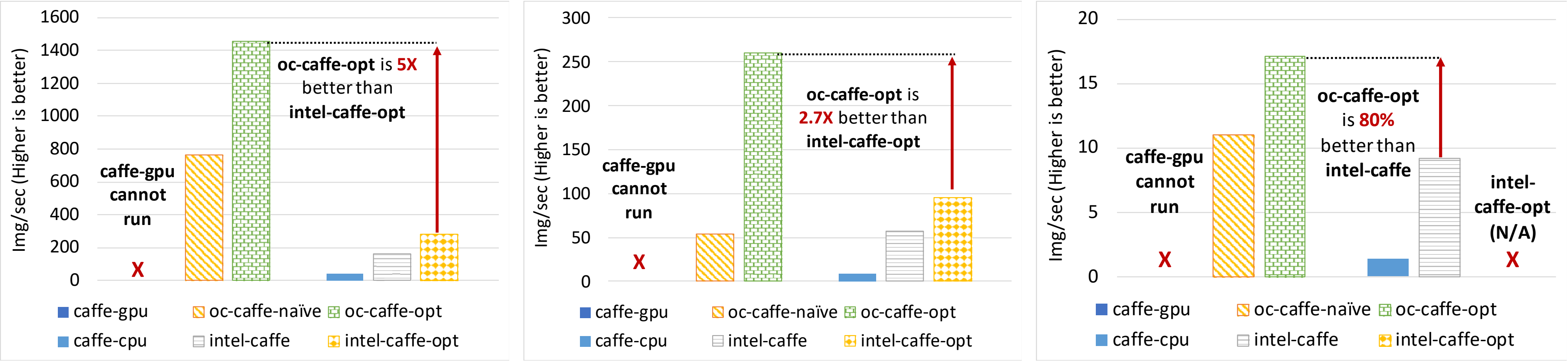
Out-of-core DNN Training

- What if your Neural Net is bigger than the GPU memory (out-of-core)?
  - Use our proposed Unified Memory solution called OC-DNN :-)



A. A. Awan, C-H Chu, X. Lu, H. Subramoni, D.K. Panda, "OC-DNN: Exploiting Advanced Unified Memory Capabilities in CUDA 9 and Volta GPUs for Out-of-Core DNN Training", 25th IEEE International Conference on High Performance Computing, Data, and Analytics (HiPC) 2018.

Out-of-core DNN Training



AlexNet **GoogLeNet** **ResNet-50**

- We exploit Unified Memory designs in CUDA 9 and hardware support in Volta GPU
  - *Better performance than CPU-based solutions for all state-of-the-art Image models*

A. A. Awan, C-H Chu, X. Lu, H. Subramoni, D.K. Panda, “OC-DNN: Exploiting Advanced Unified Memory Capabilities in CUDA 9 and Volta GPUs for Out-of-Core DNN Training”, 25th IEEE International Conference on High Performance Computing, Data, and Analytics (HiPC) 2018.

Agenda

- Introduction
- Research Challenges: Exploiting HPC for Deep Learning
- Proposed Solutions
- **Conclusion**

Summary

- Deep Learning is on the rise
  - Rapid advances in software, hardware, and availability of large datasets
- Single node or single GPU is not enough for Deep Learning workloads
- We need to focus on distributed Deep Learning but there are many challenges
- MPI offers a great abstraction for communication in DL Training tasks
- A co-design of Deep Learning frameworks and communication runtimes will be required to make DNN Training scalable

Thank You!

awan.10@osu.edu

http://web.cse.ohio-state.edu/~awan.10

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

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

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

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