An In-depth Performance Characterization of CPU- and GPU-based DNN Training on Modern Architectures

Presentation at MLHPC '17

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

Network Based Computing Laboratory

Dept. of Computer Science and Engineering

The Ohio State University

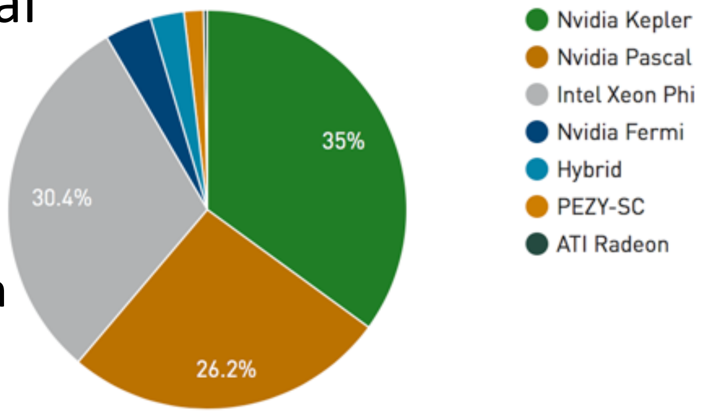
awan.10@osu.edu, {subramon,panda}@cse.ohio-state.edu

CPU based Deep Learning is not as bad as you think!

- **Introduction**
  - CPU-based Deep Learning
  - Deep Learning Frameworks
- Research Challenges
- Design Discussion
- Performance Characterization
- Conclusion

GPUs are great for Deep Learning

- NVIDIA GPUs have been the main driving force for faster training of Deep Neural Networks (DNNs)
- The ImageNet Challenge - (ILSVRC)
  - 90% of the ImageNet teams used GPUs in 2014*
  - DL models like AlexNet, GoogLeNet, and VGG
  - GPUs: A natural fit for DL due to the throughput-oriented nature
  - GPUs are also growing in the HPC arena!

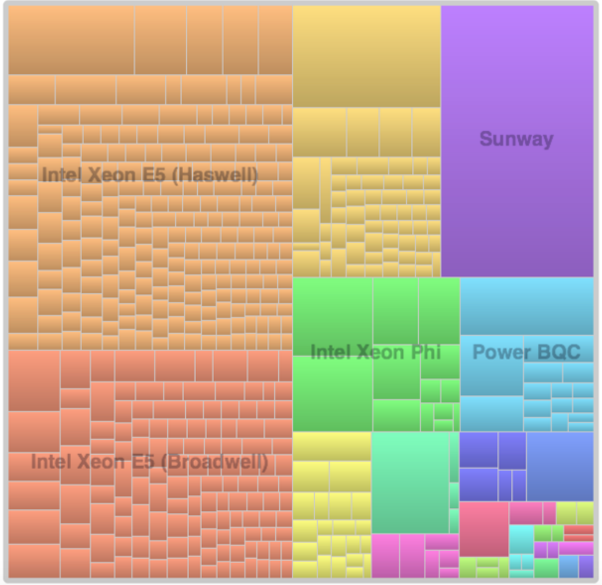


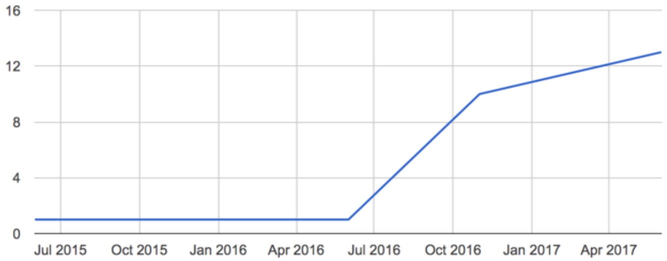
https://www.top500.org/

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

But what about CPUs?

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
- Xeon Phi 1st generation: a many-core co-processor
- Xeon Phi 2nd generation (KNL): a self-hosted many-core processor!
- Usually, we hear CPUs are ***10x – 100x*** slower than GPUs? [1-3]
  - ***But can we do better?***

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

Deep Learning Frameworks – CPUs or GPUs?

- There are several Deep Learning (DL) or DNN Training frameworks
  - Caffe, Cognitive Toolkit, TensorFlow, MXNet, and counting....
- 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 :-)
  - And MKL-DNN, just like cuDNN, has definitely rekindled this!!
  - Coupled with Intel Xeon Phi (Knights Landing or KNL) and MC-DRAM, the landscape for CPU-based DL looks promising..

The DL Framework(s) in discussion: Caffe and friends

- Caffe is a popular and widely used framework; has many forks (friends)
- NVIDIA-Caffe and BVLC-Caffe (Official Caffe) are almost similar
  - NVIDIA-Caffe is cutting edge though! (Tensor cores, Volta, DrivePX, etc.)
- Intel-Caffe is optimized for CPU-based Deep Learning
- OSU-Caffe is a multi-node multi-GPU variant that we have worked on at OSU

Caffe Variant	Multi-GPU Support	Multi-node Support	Multi-node Communication
BVLC-Caffe	Yes	No	N/A
NVIDIA-Caffe	Yes	No	N/A
Intel-Caffe	N/A	Yes	Intel MLSL 2017.1.016 (with Intel MPI 2017)
OSU-Caffe	Yes	Yes	MVAPICH2-GDR 2.2

Agenda

- Introduction
- **Research Challenges**
- Design Discussion
- Performance Characterization
- Conclusion

The Key Question!

Can we provide a holistic yet comprehensive view of DNN training performance for a diverse set of hardware architectures including Intel Xeon Phi (KNL) processors and NVIDIA Pascal GPUs?

Research Challenges

Various datasets and networks handled differently in DL frameworks

Possible strategies to evaluate the performance of DL frameworks

Performance trends that can be observed for a single node

Performance behavior for hardware features like MCDRAM

Computation and communication characteristics of DL workloads?

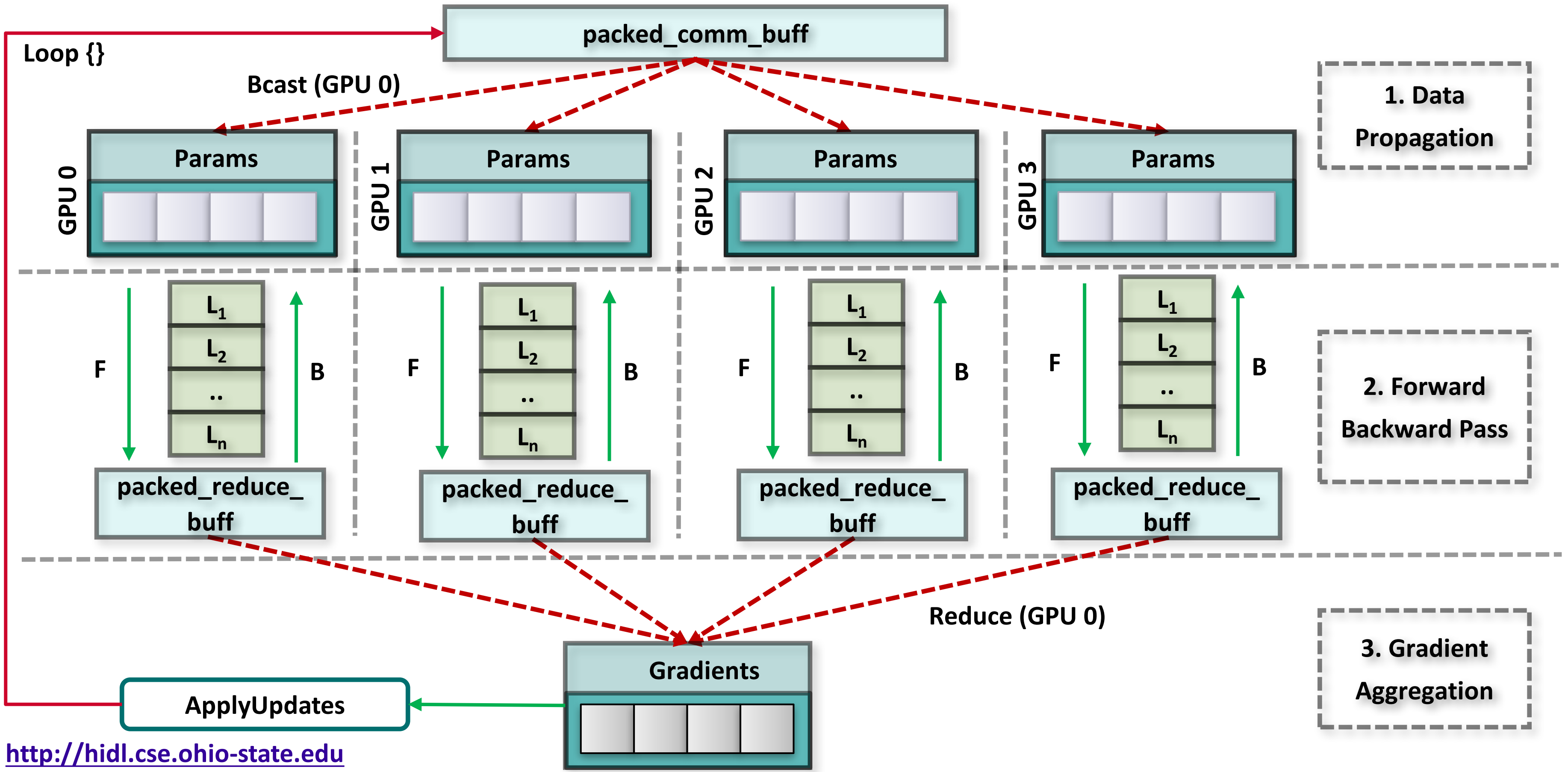
Scale-out of DNN training for CPU-based and GPU-based DNN training

Let us bring HPC and DL “together”!

Agenda

- Introduction
- Research Challenges
- **Design Discussion**
  - Caffe Architecture
  - Understanding the Impact of Execution Environments
  - Multi-node Training: Intel-Caffe, OSU-Caffe, and MPI
- Performance Characterization
- Conclusion

Caffe Architecture

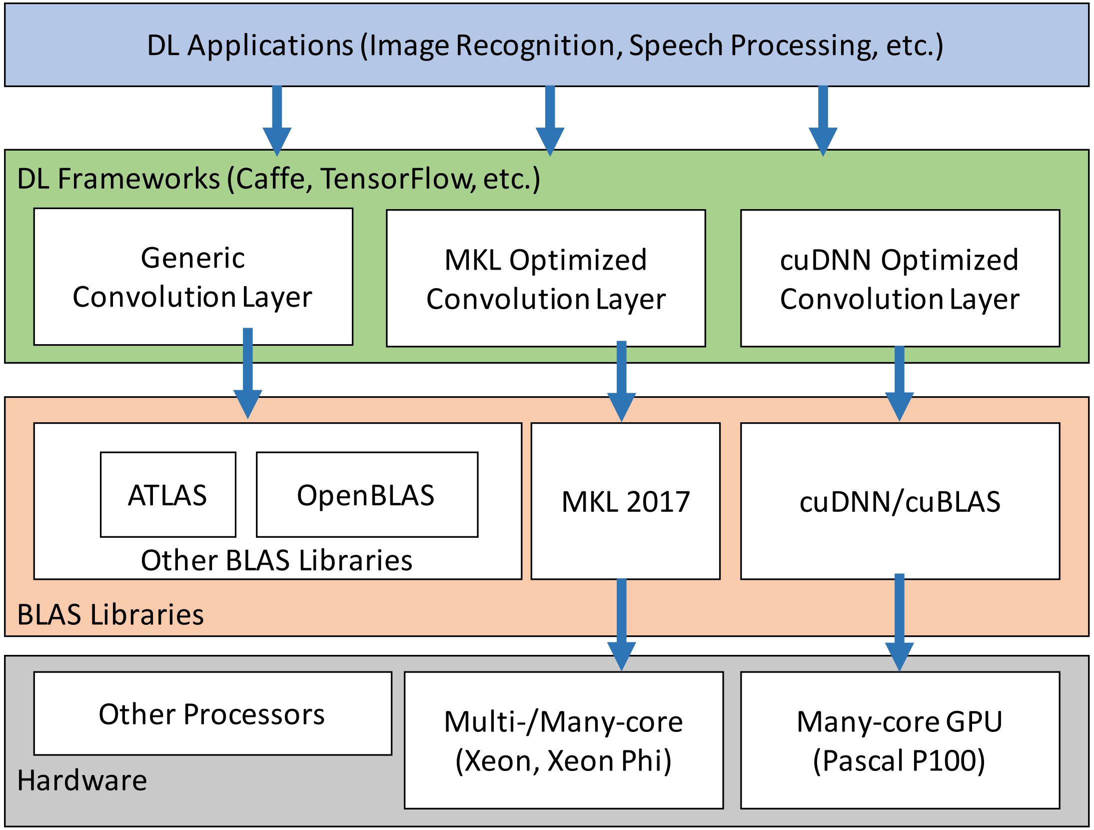


http://hidl.cse.ohio-state.edu

Understanding the Impact of Execution Environments

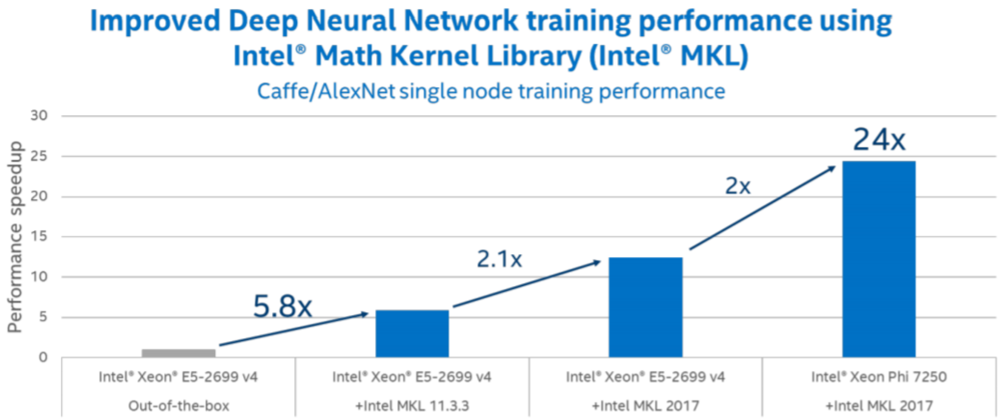
Performance is dependent on:

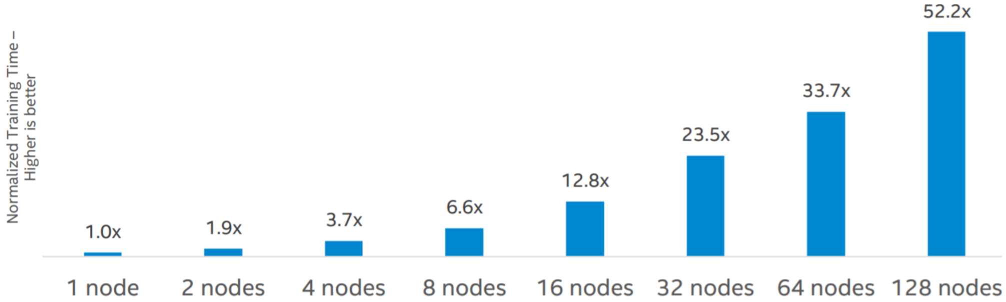
1. Hardware Architectures
   - GPUs
   - Multi-/Many-core CPUs
2. Software Libraries
   - cuDNN (for GPUs)
   - MKL-DNN/MKL 2017 (for CPUs)
3. Hardware/Software co-design
   - Software libraries optimized for one platform will not help the other!
   - cuDNN vs. MKL-DNN



Intel-Caffe and Intel MKL

- MKL-DNN: The key performance difference for CPU-based DNN training!
- Does that really work in practice?
- Intel MKL claims to offer much better performance
- Intel MLSL promises multi-node training





Multi-Node scaling using Intel Omni-Path on AlexNet

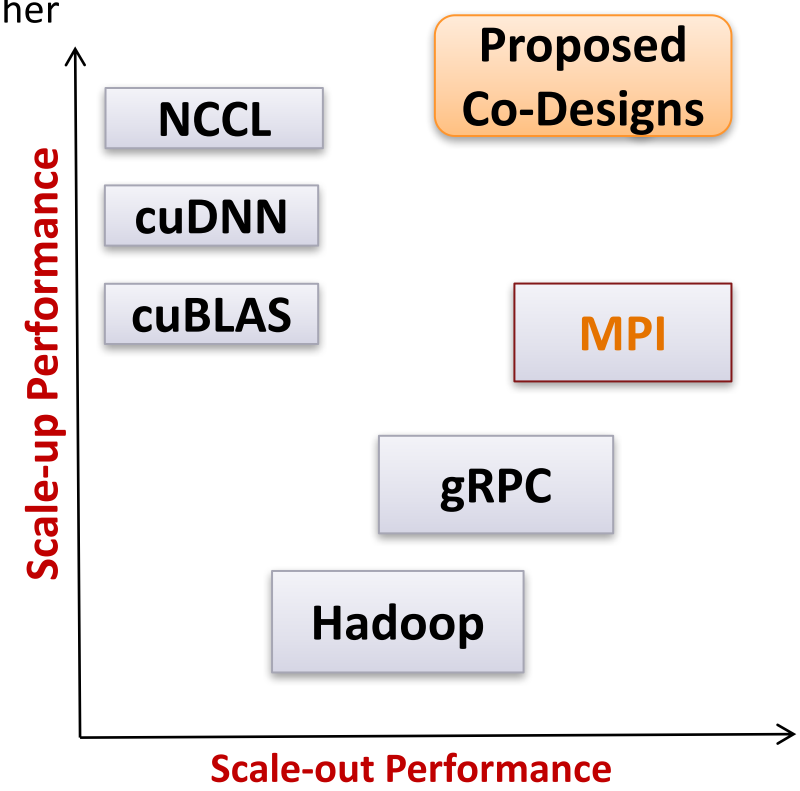
Courtesy: http://www.techenablement.com/accelerating-python-deep-learning/

So what to use for Scale-out with Intel-Caffe?

- We need a communication library for Scale-out?
  - Message Passing Interface (MPI) libraries like MVAPICH, Intel MPI, etc.
  - NVIDIA NCCL, Facebook Gloo, Baidu-allreduce, etc.
  - Intel Machine Learning Scaling Library (higher level library built on top of MPI)
- How to choose?
  - For GPU-based frameworks, CUDA-Aware MPI, NCCL, and Gloo
  - For CPU-based frameworks, any MPI library will do
    - MLSL offers something more
    - MLSL is sort of a DL framework API – can be used inside the framework
    - But can be used in a stand-alone format too!

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

- Deep Learning frameworks are a different game altogether
  - Unusually large message sizes (order of megabytes)
  - Most communication based on GPU buffers
- State-of-the-art
  - cuDNN, cuBLAS, NCCL --> **scale-up** performance
  - CUDA-Aware MPI --> **scale-out** performance
    - For small and medium message sizes only!
- 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**?



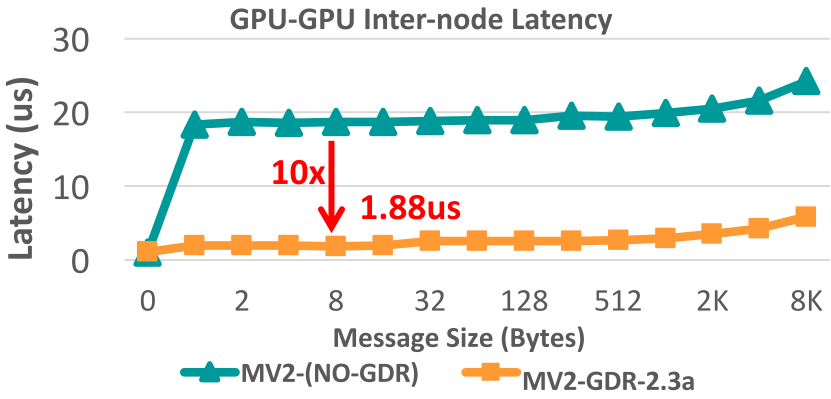
A. A. Awan, K. Hamidouche, J. M. Hashmi, and D. K. Panda, S-Caffe: Co-designing MPI Runtimes and Caffe for Scalable Deep Learning on Modern GPU Clusters. In *Proceedings of the 22nd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming* (PPoPP '17)

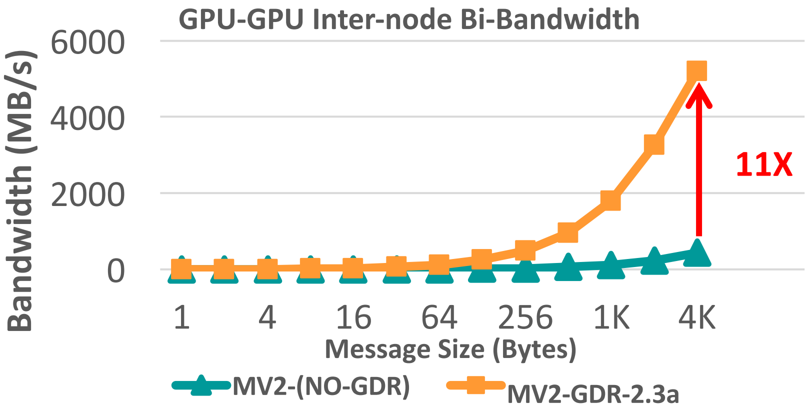
Overview of the MVAPICH2 Project

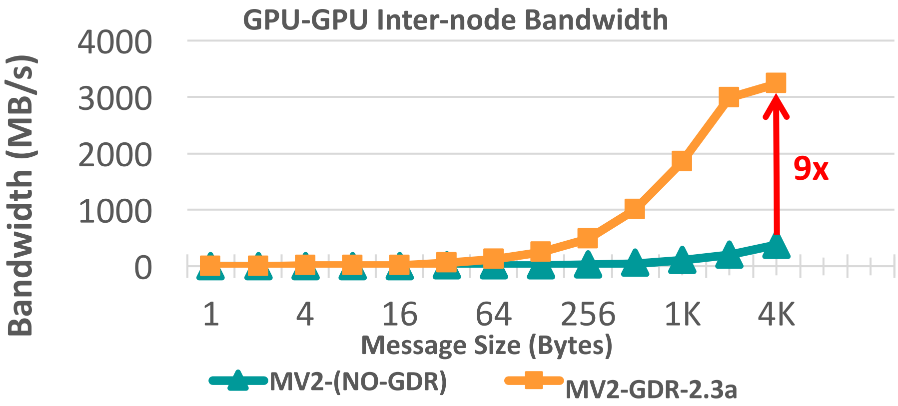
- High Performance open-source MPI Library for InfiniBand, Omni-Path, Ethernet/iWARP, and RDMA over Converged Ethernet (RoCE)
  - MVAPICH (MPI-1), MVAPICH2 (MPI-2.2 and MPI-3.0), Started in 2001, First version available in 2002
  - MVAPICH2-X (MPI + PGAS), Available since 2011
  - ***Support for GPGPUs (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 2,825 organizations in 85 countries**
  - **More than 432,000 (> 0.4 million) downloads from the OSU site directly**
  - Empowering many TOP500 clusters (June '17 ranking)
    - **1st, 10,649,600-core (Sunway TaihuLight) at National Supercomputing Center in Wuxi, China**
    - 15th, 241,108-core (Pleiades) at NASA
    - 20th, 462,462-core (Stampede) at TACC
  - 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
  - System-X from Virginia Tech (3rd in Nov 2003, 2,200 processors, 12.25 TFlops) ->
  - Sunway TaihuLight (1st in Jun'17, 10M cores, 100 PFlops)

Scale-out for GPU-based Training







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

MVAPICH2-GDR: Performance that meets Deep Learning requirements!

Agenda

- Introduction
- Research Challenges
- Design Discussion
- **Performance Characterization**
  - Single-node Performance
  - Multi-node Performance
- Conclusion

Performance Characterization

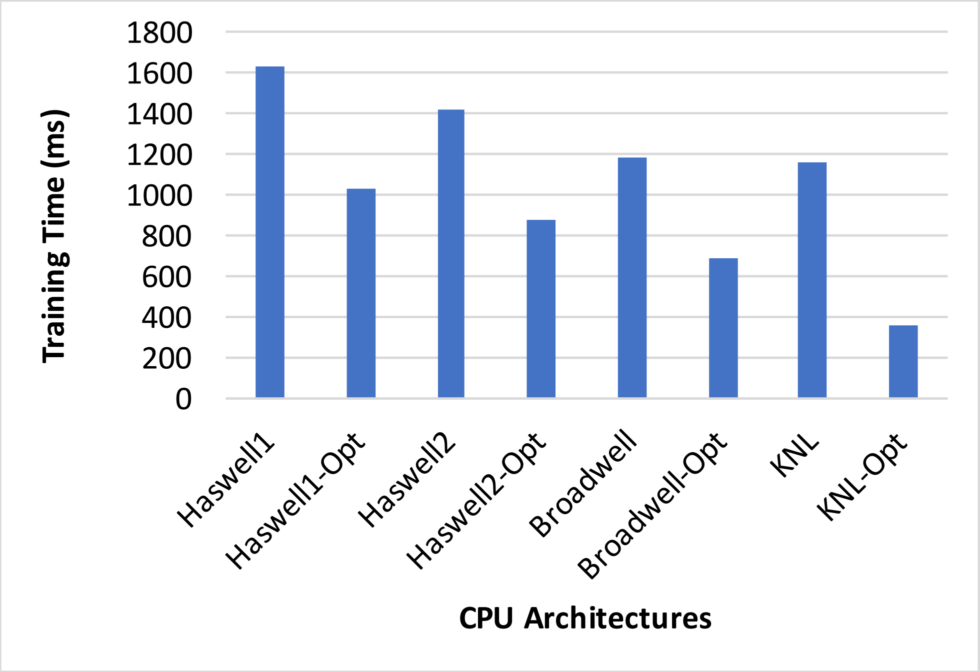
- Several GPU generations and CPU architectures
- Single-node Results for AlexNet and ResNet-50
  - Impact of MKL engine
  - Impact of MC-DRAM
  - Layer-wise breakdown
  - P100 vs. KNL
- Multi-node results using Intel-Caffe and OSU-Caffe
  - Weak scaling
  - ResNet-50 and AlexNet

Performance Characterization: Various Architectures

Name (Label)	Processor Architecture (Description)	No. of Cores	No. of Sockets
Haswell1	Intel Xeon CPU E5-2660 v3 @ 2.60 GHz	20 (2*10)	2
Haswell2	Intel Xeon CPU E5-2687 v3 @ 3.10 GHz	20 (2*10)	2
Broadwell	Intel Xeon CPU E5-2680 v4 @ 2.40 GHz	28 (2*14)	2
KNL	Intel Xeon Phi CPU 7250 @ 1.40 GHz	68 (1*68)	1
K40	NVIDIA Tesla K40 11.8GB @ 0.75 GHz	2880 CUDA Cores	N/A
K80	NVIDIA Tesla K80 11.8GB @ 0.82 GHz	2496 CUDA Cores	N/A
P100	NVIDIA Tesla P100-PCIE 1 6GB @ 1.33 GHz	3584 CUDA Cores	N/A

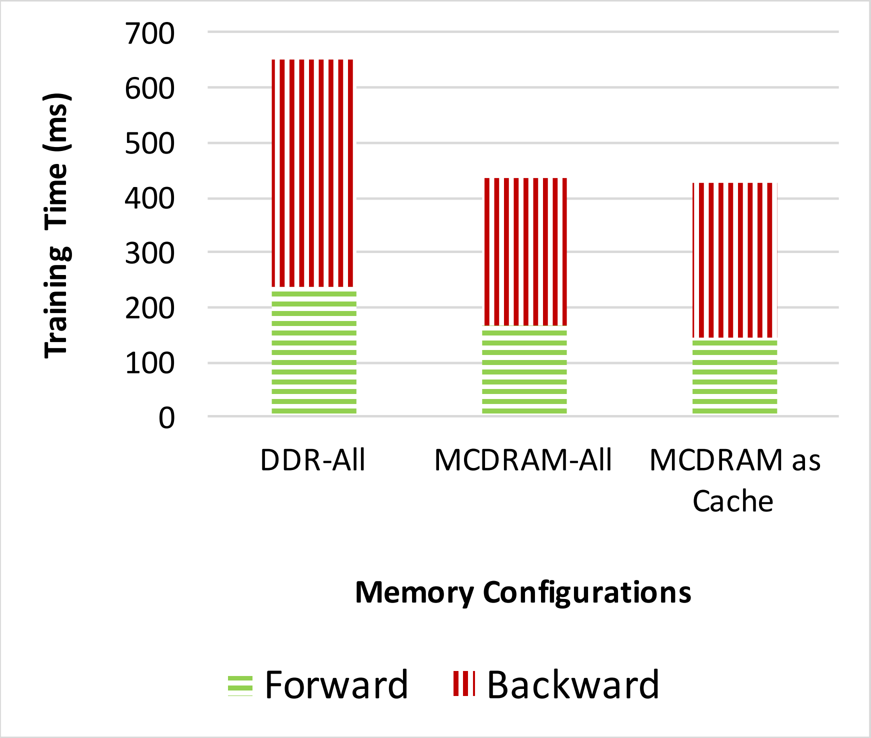
Single-node: Impact of MKL engine in Intel-Caffe

- The comparison of optimized MKL engine and the default Caffe engine
- 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***)

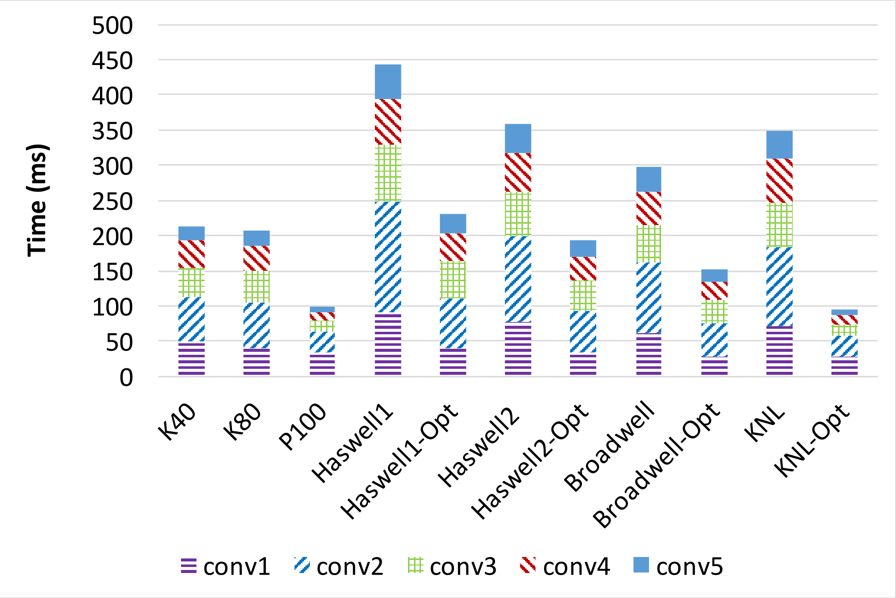


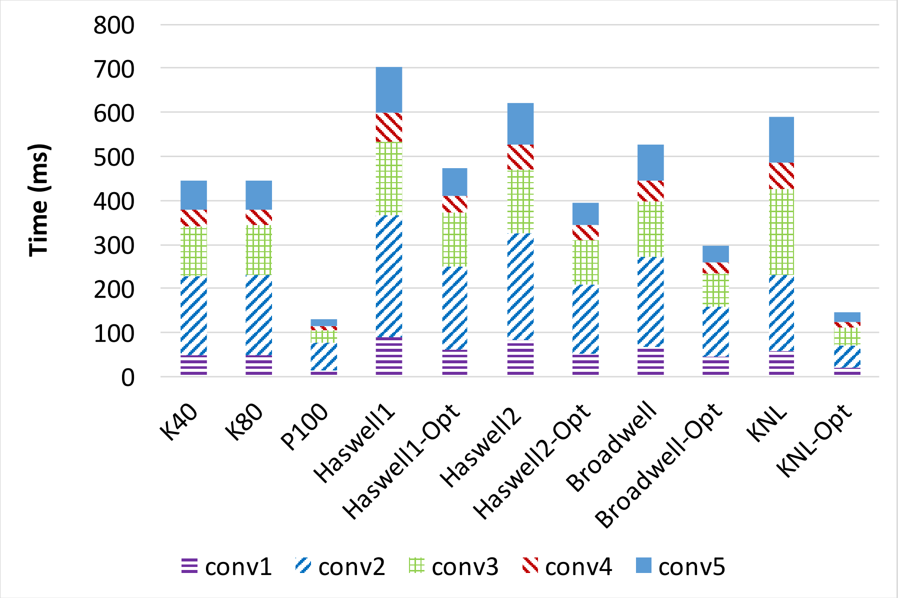
Single-node: Impact of Utilizing MCDRAM

- "MCDRAM as Cache" and "MCDRAM-All" offer very similar performance
- We chose to use ***MCDRAM as Cache*** for all the subsequent results
- On average, DDR-All is up to ***1.5X slower*** than MCDRAM



Diving Deeper: Layer-wise Breakdown

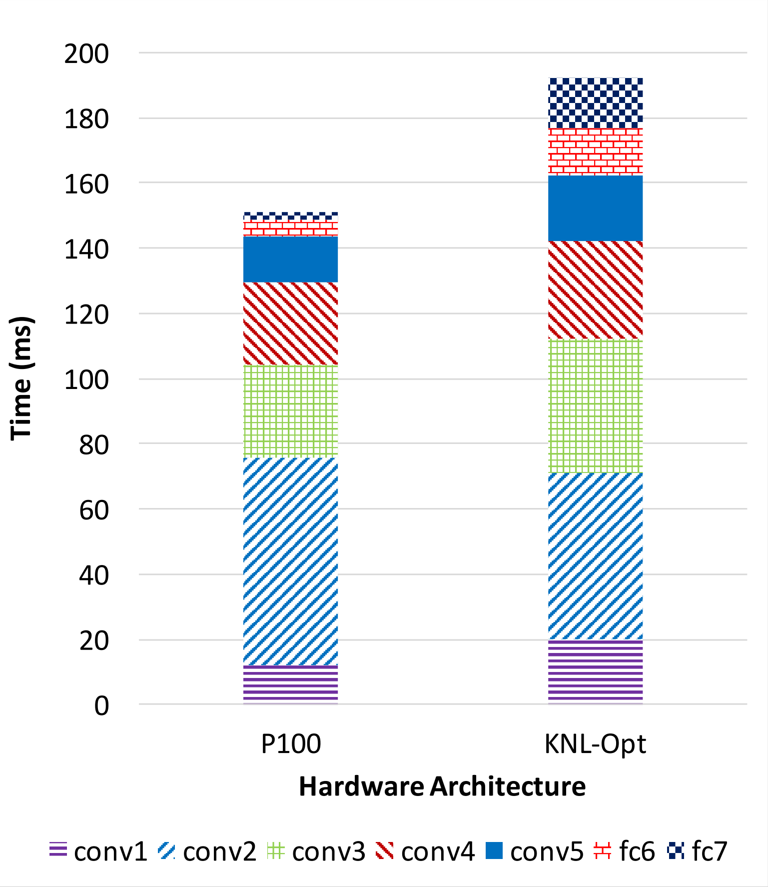




- The full landscape for AlexNet: Forward and Backward Pass
- ***Faster Convolutions → Faster Training***
- Most performance gains are based on ***conv2*** and ***conv3*** for AlexNet

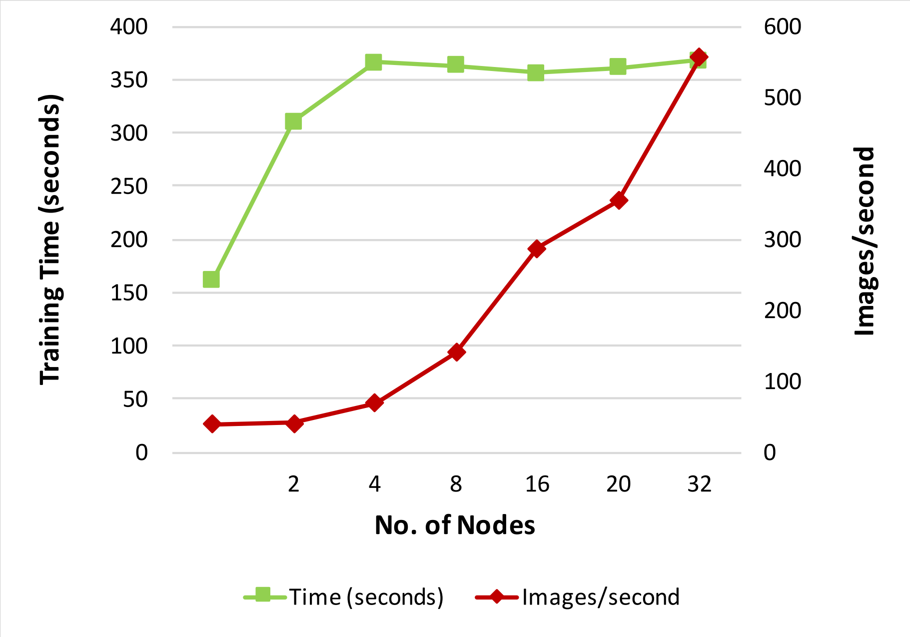
Diving Deeper: P100 vs. KNL (AlexNet)

- Fully connected layers are much slower on KNL compared to P100
- ***conv1*** and ***conv3*** also contribute to degradation on KNL
- ***conv2*** is faster on KNL compared to P100
- ResNet-50 has some surprises (*not shown on this slide*)
  - KNL performs ***significantly better*** than P100
  - Difficult to visualize as there are several layers in ResNet-50



Multi-node Results: ResNet-50

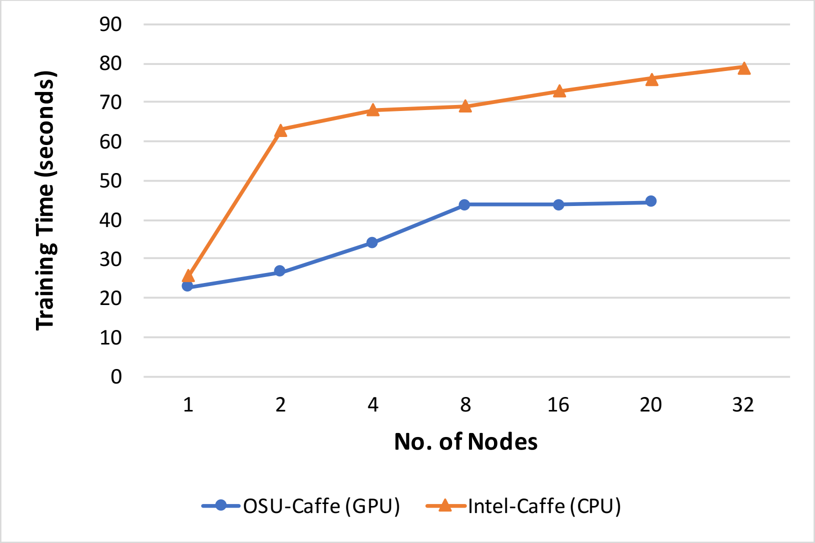
- All results are ***weak scaling***
  - The batch size remains constant/solver
  - But increases overall by:
  - *batch-size * (#nodes or #gpus)*
- Images/second is a derived metric but more meaningful for understanding scalability
- Efficiency is another story [1]
  - ***Larger DNN architectures → Less scalability due to communication overhead***

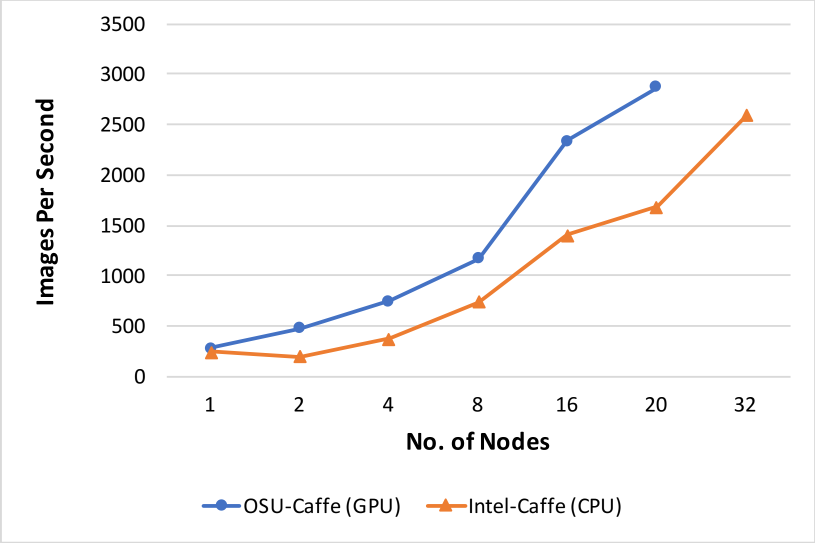


ResNet-50 Intel-Caffe

1. *Experiences of Scaling TensorFlow On Up to 512 Nodes On CORI Supercomputer*, Intel HPC Dev. Con., https://www.intel.com/content/www/us/en/events/hpcdevcon/overview.html

Multi-node Results: AlexNet Comparison





- OSU-Caffe vs. Intel-Caffe
  - Different frameworks so not directly comparable
  - A rough comparison can still help in understanding scalability trends
  - Design of framework can affect performance for distributed training
    - ***MPI (or the communication runtime) can cause a marked difference***

Agenda

- Introduction
- Research Challenges
- Design Comparisons
- Performance Characterization
- **Conclusion**

Conclusion

- CPU is very comparable to GPU for DNN Training workloads if appropriate optimizations are exploited
- GPUs are still faster than CPUs in general
- KNL beats P100 for one case but P100 beats KNL for most cases
- Evaluating the performance of a DL framework
  - The hardware architecture matters
  - But software stack has a higher and more significant impact than hardware
  - The full execution environment and communication runtime needs to be evaluated to ensure fairness in comparisons

Future Work

- Evaluate with upcoming architectures
  - Volta GPUs
  - DGX-1V System
  - Intel Nervana Neural Network Processor
- Verify the hypothesis using other DL frameworks
  - TensorFlow
  - Intel Neon
  - Nervana Graph
- Investigate new designs with MVAPICH2 and other MPI stacks to support faster DNN training

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/

HiDL
High-Performance
Deep Learning

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

MVAPICH
MPI, PGAS and Hybrid MPI+PGAS Library

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

Please join us for other events at SC '17

- Workshops
  - ESPM2 2017: Third International Workshop on Extreme Scale Programming Models and Middleware
- Tutorials
  - InfiniBand, Omni-Path, and High-Speed Ethernet for Dummies
  - InfiniBand, Omni-Path, and High-Speed Ethernet: Advanced Features, Challenges in Designing, HEC Systems and Usage
- BoFs
  - MPICH BoF: MVAPICH2 Project: Latest Status and Future Plans
- ACM SRC Posters
  - Co-designing MPI Runtimes and Deep Learning Frameworks for Scalable Distributed Training on GPU Clusters
  - High-Performance and Scalable Broadcast Schemes for Deep Learning on GPU Clusters
- Booth Talks
  - The MVAPICH2 Project: Latest Developments and Plans Towards Exascale Computing
  - Exploiting Latest Networking and Accelerator Technologies for MPI, Streaming, and Deep Learning: An MVAPICH2-Based Approach
  - Accelerating Deep Learning with MVAPICH
  - MVAPICH2-GDR Library: Pushing the Frontier of HPC and Deep Learning

Please refer to http://mvapich.cse.ohio-state.edu/talks/ for more details