AI Developments at Intel

Dr. Atanas Atanasov
HPC Solution Architect

Legal Disclaimers

Information in this document is provided in connection with Intel® products. No license, express or implied, by estoppels or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel® products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

This document contains information on products in the design phase of development. The information here is subject to change without notice. Do not finalize a design with this information.

Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. Intel may make changes to specifications and product descriptions at any time, without notice.

Intel processors and chipsets may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel® Advanced Vector Extensions (Intel® AVX)* provides higher throughput to certain processor operations. Due to varying processor power characteristics, utilizing AVX instructions may cause a) some parts to operate at less than the rated frequency and b) some parts with Intel® Turbo Boost Technology 2.0 to not achieve any or maximum turbo frequencies. Performance varies depending on hardware, software, and system configuration and you can learn more at http://www.intel.com/go/turbo.

Intel® processors of the same SKU may vary in frequency or power as a result of natural variability in the production process.

Performance estimates or simulated results based on internal Intel analysis or architecture simulation or modeling are provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance

Intel, Intel Xeon, Intel Itanium, and Intel Netburst are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States or other countries.

*Other brands and names may be claimed as the property of others.

Legal Disclaimer & Optimization Notice

INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS". NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO THIS INFORMATION INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

Optimization Notice

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

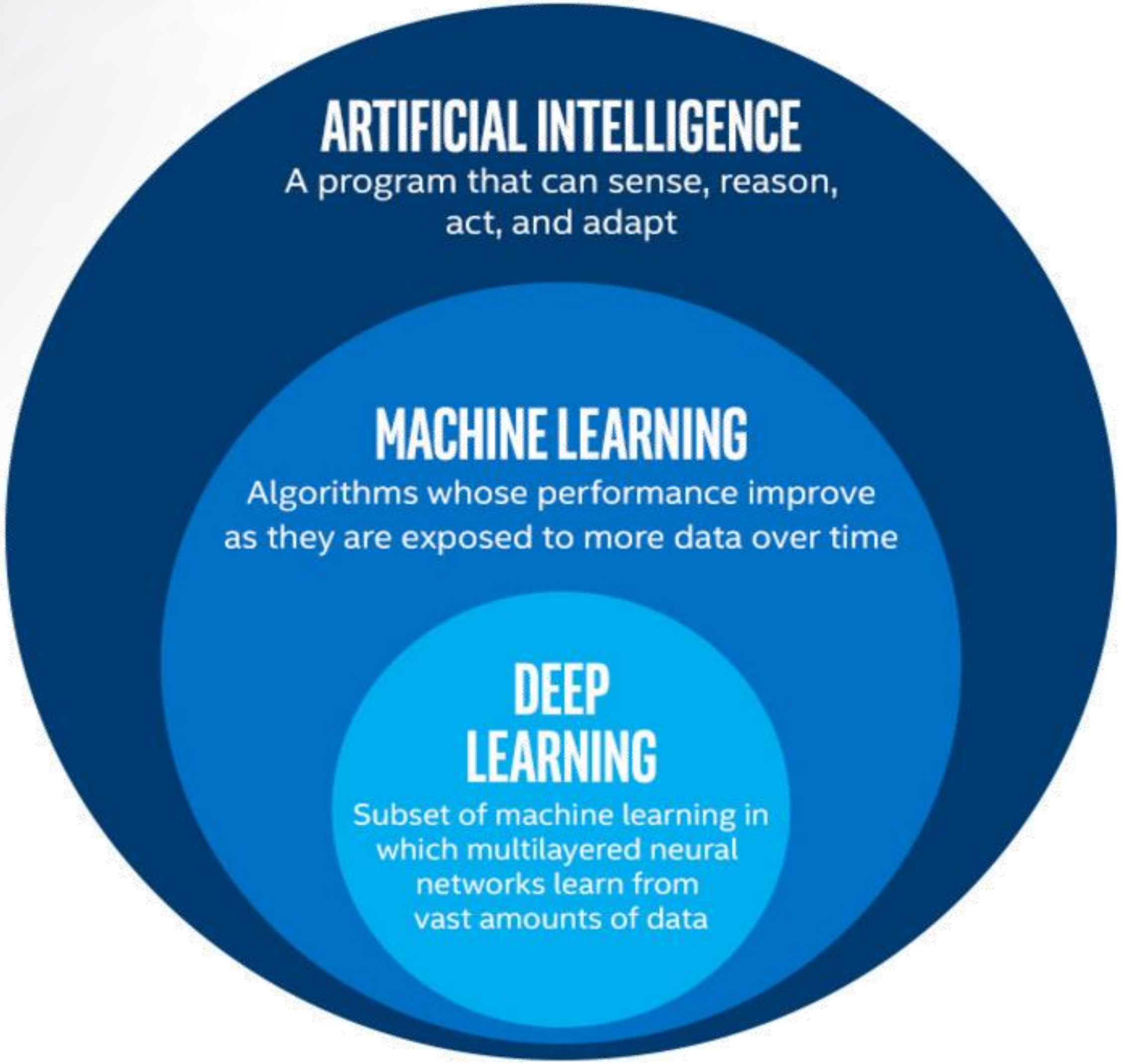
Notice revision #20110804

Agenda

- Brief Deep Learning Training Overview
- AI Optimizations
- Why Multi-node Training?
- Summary

Brief AI Training Overview

ARTIFICIAL INTELLIGENCE



ARTIFICIAL INTELLIGENCE

A program that can sense, reason, act, and adapt

MACHINE LEARNING

Algorithms whose performance improve as they are exposed to more data over time

DEEP LEARNING

Subset of machine learning in which multilayered neural networks learn from vast amounts of data

MACHINE LEARNING

CLASSIC MACHINE LEARNING

How do you engineer the best features?

$N \times N$

$(f_1, f_2, \dots, f_K)$

Roundness of face
Dist between eyes
Nose width
Eye socket depth
Cheek bone structure
Jaw line length
...etc.

CLASSIFIER ALGORITHM

SVM
Random Forest
Naïve Bayes
Decision Trees
Logistic Regression
Ensemble methods

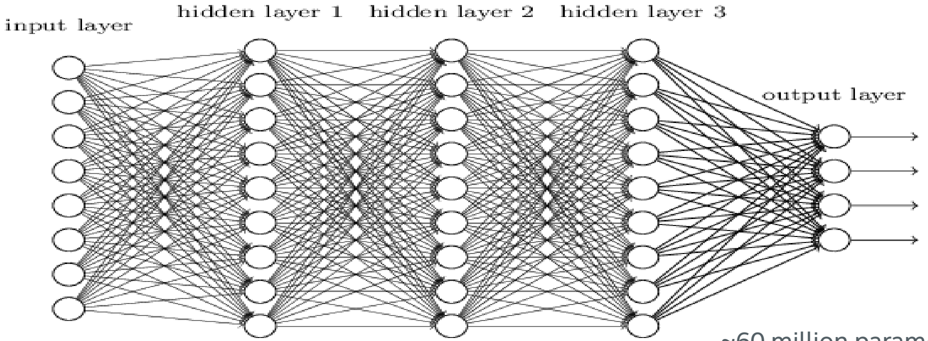
Arjun

DEEP LEARNING

How do you guide the model to find the best features?

$N \times N$

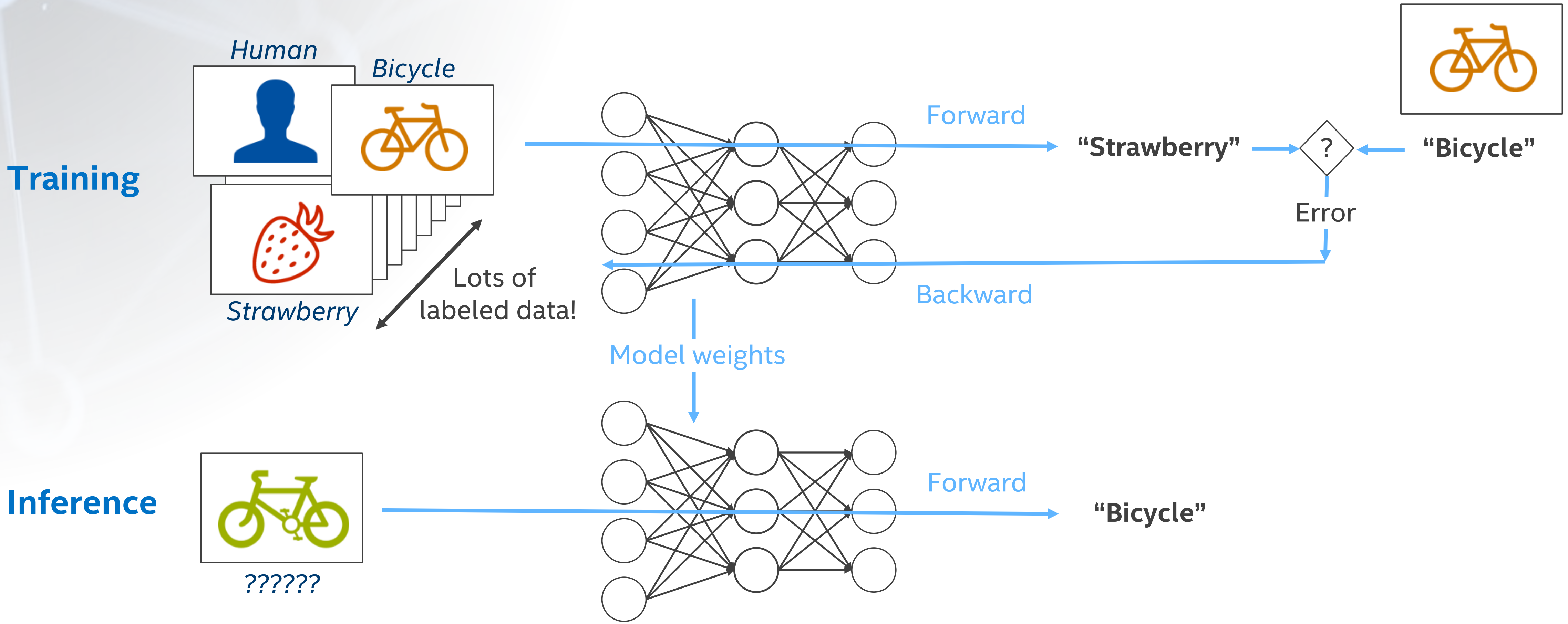
NEURAL NETWORK



~60 million parameters

Arjun

DEEP LEARNING: TRAINING VS. INFERENCE



INTEL® NERVANA™ PORTFOLIO

EXPERIENCES

PLATFORMS

Intel® Nervana™ Cloud & Appliance

Intel® Nervana™ DL Studio

Intel® Computer Vision SDK

Intel® Movidius™ MDK

intel Saffron™

FRAMEWORKS

Apache Spark

MLlib

BigDL

neon

TensorFlow

Microsoft CNTK

mxnet

torch

Caffe

Caffe2

Chainer

theano

LIBRARIES

python Intel Python Distribution

Intel® Data Analytics Acceleration Library (DAAL)

Intel® Nervana™ Graph*

Intel® Math Kernel Library (MKL, MKL-DNN)

HARDWARE

intel XEON inside | * | intel ARRIA 10 inside | intel CORE i7 inside | intel ATOM inside

Compute

Memory & Storage

Networking

INSIDE AI

*Future

Other names and brands may be claimed as the property of others.

DATACENTER AI

Intel® Stratix® 10 FPGA

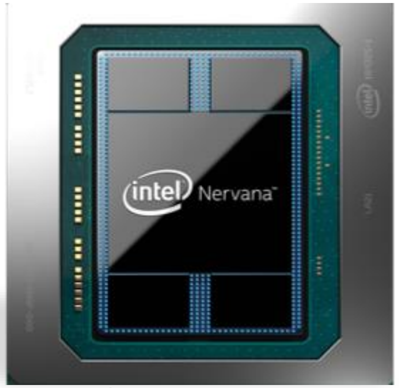
FLEXIBLE ACCELERATION

Accelerate the widest range of AI and other workloads & configurations

Intel® Xeon® Processor Scalable Family

FOUNDATION FOR AI

Begin your journey with the AI you need on the chip you know



Intel® Nervana™ Neural Network Processor*

DEEP LEARNING BY DESIGN

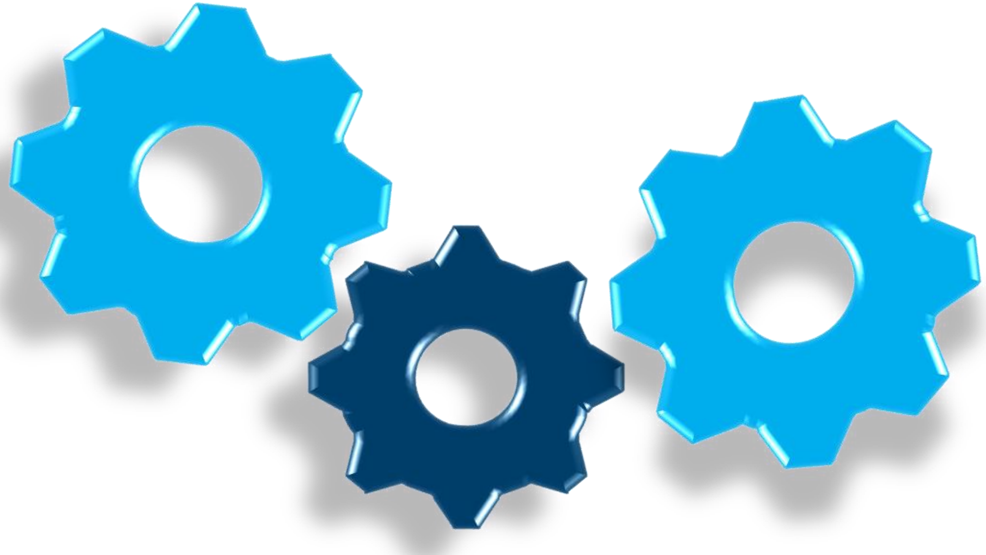
Accelerate the most intensive deep learning deployments with this custom-built processor

*Future product that was formerly codenamed the "Crest family"
All performance positioning claims are relative to other processor technologies in Intel's AI datacenter portfolio
All products, computer systems, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.

AI Optimizations

Performance Drivers for AI Workloads

Compute



Bandwidth

SW Optimizations

```
a.length (x=a[i]) x.oSrc
d.MM_p) d.MM_p=new Array();
preloadImages.arguments; for(i=0; i<a.length
d.MM_p[j]=new Image; d.MM_p[j++].src=a[i]
n.indexOf("?"))>0&&parent.frames.length
].document; n=n.substring(0,p);}
for (i=0;!x&&i<d.forms.length;i++) x=d.forms
layers.length;i++) x=MM_findObj(n,d.layers[i].document
getElementById(n); return x;}
arguments; document.MM_sr=new Array; for(i=0;i<a.length
```

Hardware Optimized Libraries and Frameworks

Scaling

- Improve load balancing
- Reduce synchronization events, all-to-all comms

Utilize all the cores

- OpenMP, MPI
- Reduce synchronization events, serial code
- Improve load balancing

Vectorize/SIMD

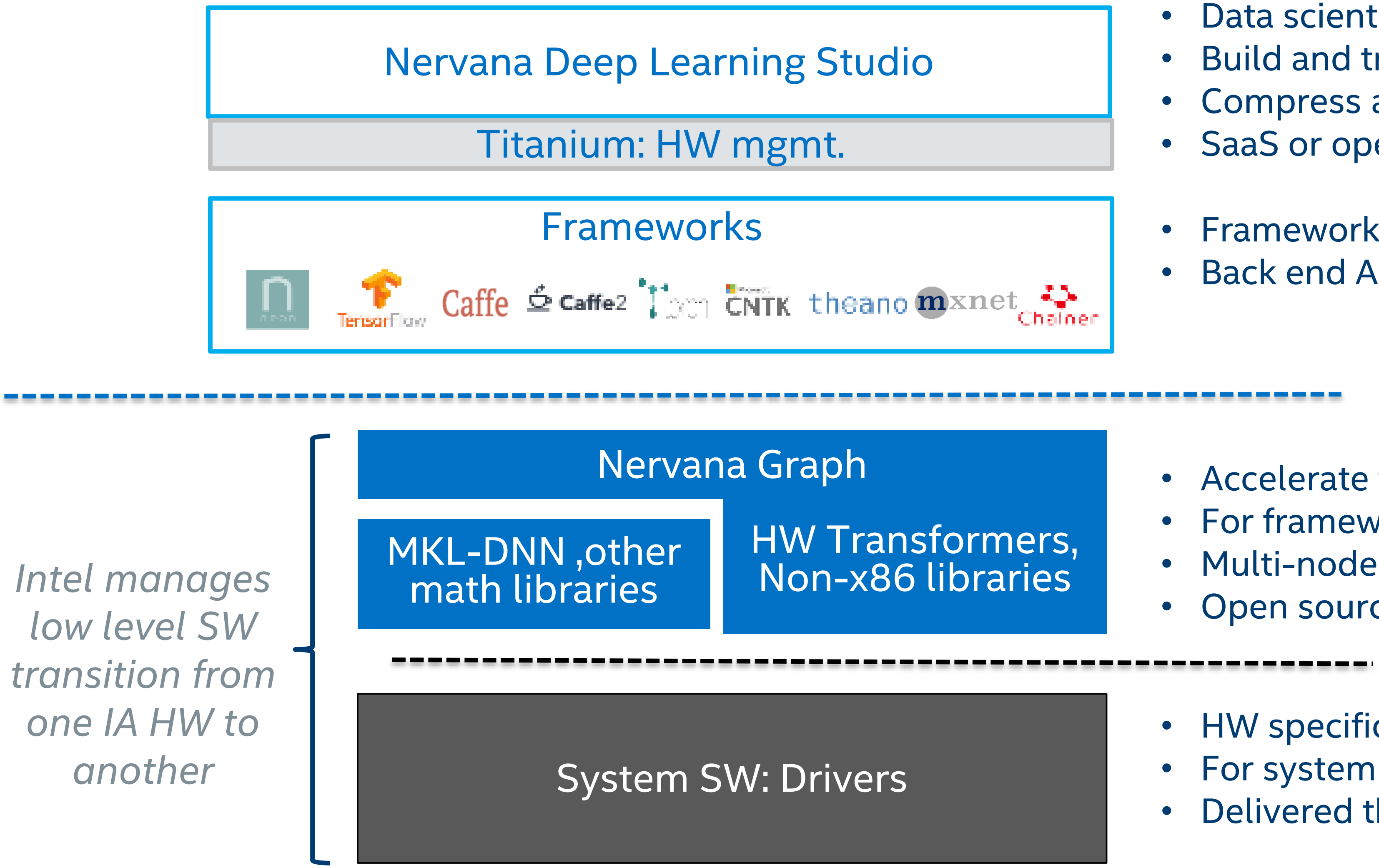
- Unit strided access per SIMD lane
- High vector efficiency
- Data alignment

Efficient memory/cache use

- Blocking
- Data reuse
- Prefetching
- Memory allocation

Important to use optimized software frameworks and libraries for best AI workload performance

Deep Learning Software Summary



Nervana Deep Learning Studio

Titanium: HW mgmt.

Frameworks

Nervana Graph

MKL-DNN ,other math libraries

HW Transformers, Non-x86 libraries

System SW: Drivers

Intel manages low level SW transition from one IA HW to another

- Data scientist and developer DL productivity tools
- Build and train models
- Compress and export modes to end points
- SaaS or open source

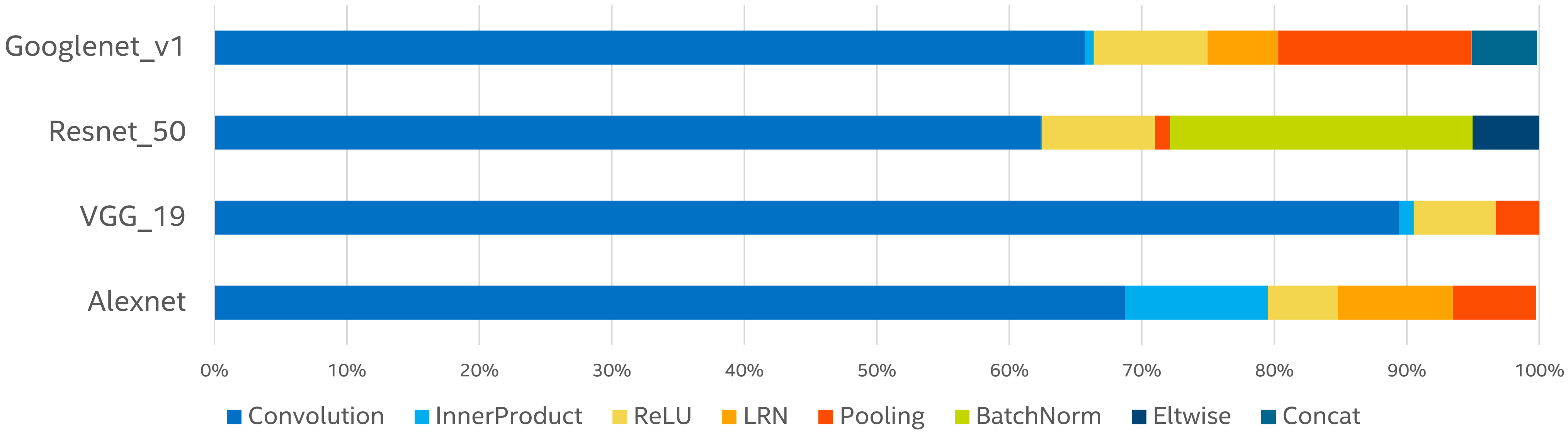
- Frameworks for developers
- Back end APIs to Nervana Graph

- Accelerate framework optimization on IA
- For framework developers & Intel
- Multi-node optimizations
- Open source

- HW specific SW
- For system developers
- Delivered through HW sales

Majority of Cycles Spent in Convolution

Percentage of time spent on individual layers of different network topologies running inference with Caffe* framework on Intel® Xeon® Processor E5-2699 v4



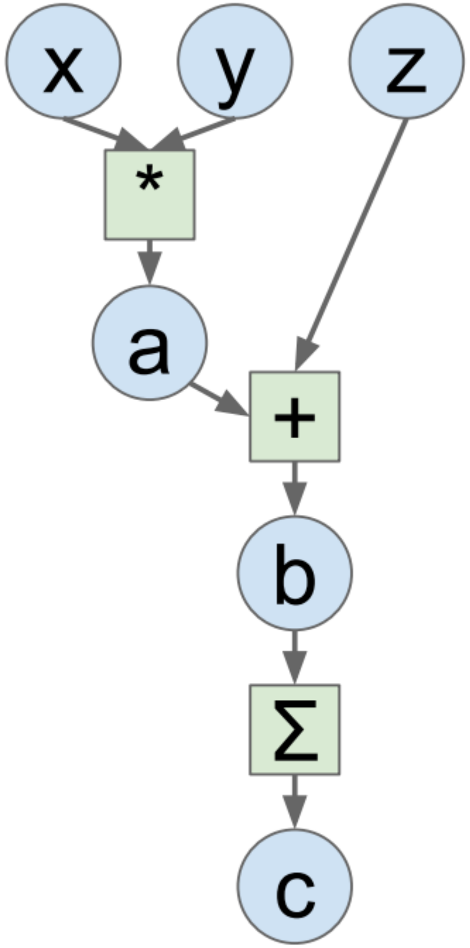
60 to 85% Time Spent in Convolution - AI workloads such as image recognition are compute heavy

Batch Sizes AlexNet:256 VGG-19: 64 ResNet-50: 50 GoogleNet-V1: 96 Configuration Details on Slide: 27
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit: http://www.intel.com/performance Source: Intel measured as of June 2017. *other names and brands may be claimed as property of others.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Building blocks (primitives) from mkl-dnn

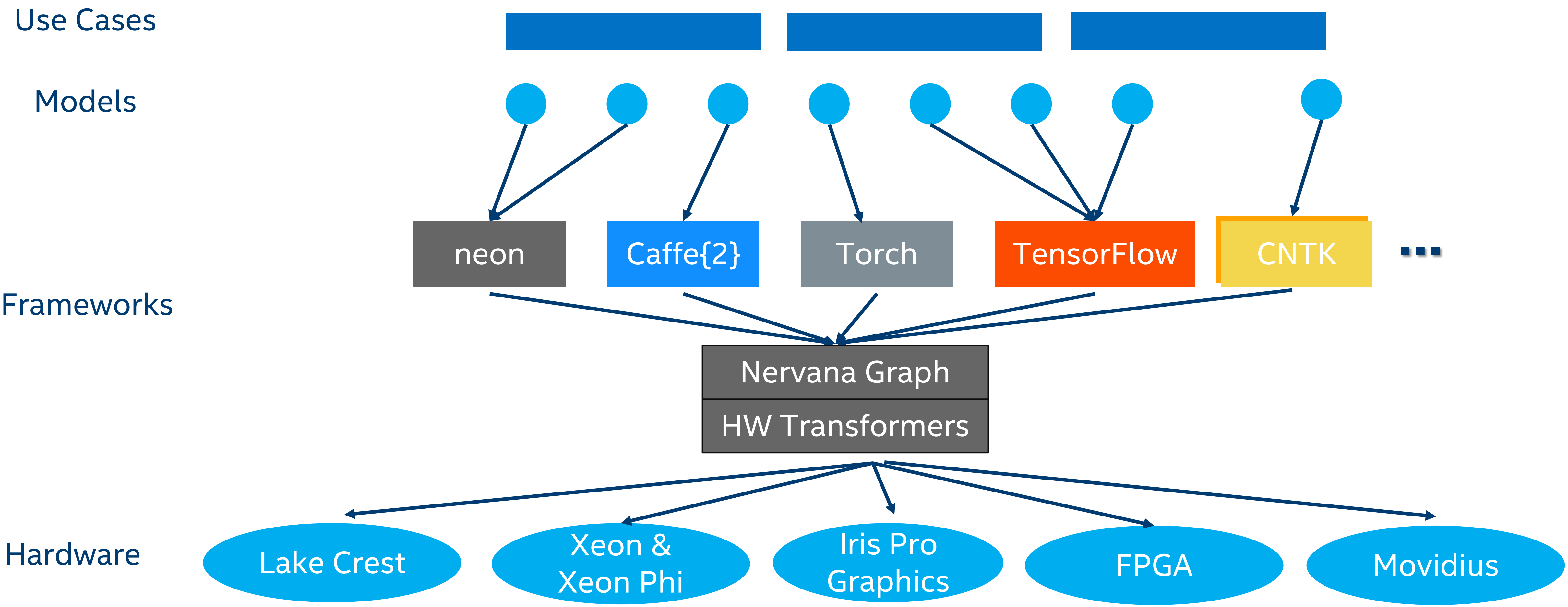
1. Matrix multiplication
2. Convolution and direct batched convolution
3. Inner product
4. Pooling: maximum, minimum, average
5. Normalization
6. Activation: rectified linear unit (ReLU)
7. Data manipulation: multi-dimensional transposition



https://software.intel.com/en-us/articles/tensorflow-optimizations-on-modern-intel-architecture

Intel® Nervana™ graph

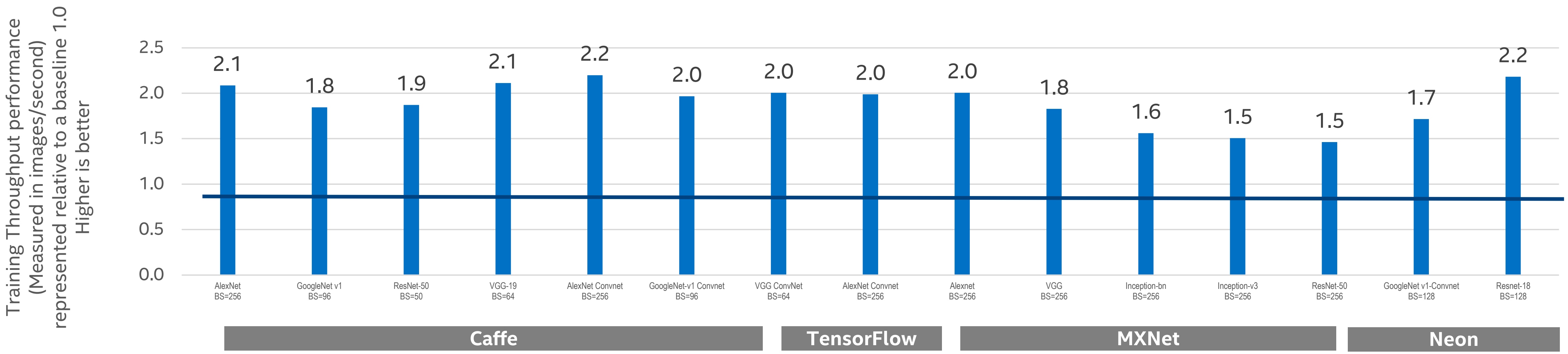
High-Performance Execution Graph for Neural Networks



Up to 2.2x Higher Training Throughput

on Intel® Xeon® Platinum 8180 Processor

Intel® Xeon® Platinum 8180 Processor Training throughput over Intel® Xeon® Processor E5-2699 v4



Intel® Xeon® Platinum Processor delivers high Training throughput performance across different frameworks

Configuration Details on Slide:24,25

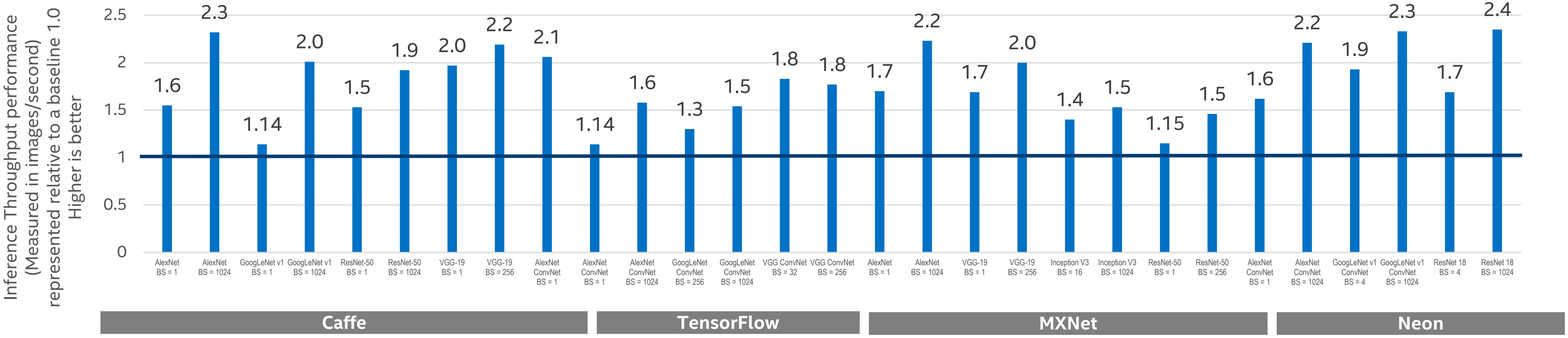
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit: http://www.intel.com/performance Source: Intel measured as of June 2017

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Up to 2.4x Higher Inference Throughput

on Intel® Xeon® Platinum 8180 Processor

Intel® Xeon® Platinum 8180 Processor Inference throughput over Intel® Xeon® Processor E5-2699 v4



Intel® Xeon® Platinum Processor delivers high Inference throughput performance across different frameworks

INFERENCE using FP32 Configuration Details on Slide:24,25

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit: http://www.intel.com/performance Source: Intel measured as of June 2017

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

INTEL® XEON® PROCESSOR PLATFORM PERFORMANCE

Hardware plus optimized software

INFERENCE THROUGHPUT

Up to

138x

Intel® Xeon® Platinum 8180 Processor
higher Intel optimized Caffe GoogleNet v1 with Intel® MKL
inference throughput compared to
Intel® Xeon® Processor E5-2699 v3 with BVLC-Caffe

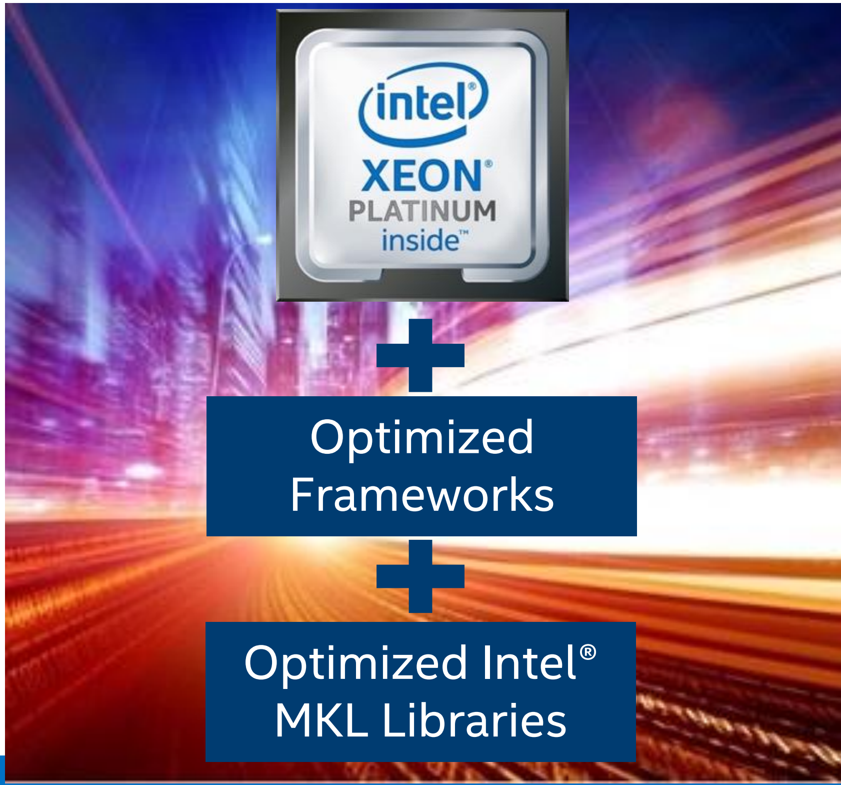
TRAINING THROUGHPUT

Up to

113x

Intel® Xeon® Platinum 8180 Processor
higher Intel Optimized Caffe AlexNet with Intel® MKL
training throughput compared to
Intel® Xeon® Processor E5-2699 v3 with BVLC-Caffe

Inference and training throughput measured with FP32 instructions. Inference with INT8 will be higher.



Deliver significant AI performance with hardware and software optimizations on Intel® Xeon® Scalable Processors

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit: http://www.intel.com/performance Source: Intel measured as of June 2017 Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

Why Multi-node Training?

Accuracy and reduced time to train:

The goal to drive innovation

Properties of Neural Networks[1]:

Results get better with

- More Data
- Bigger models
- More computation.

Training is not a one time effort.

- Many operational neural networks as part of different applications
- Each neural network may be trained with domain specific training sets
- Evolving input data sets drives the need for re-training

Google Brain's Jeff Dean quantifies benefits of reducing the time to train:

Minutes, hours:

- Interactive research
- Instant gratification of results

1-4 Days:

- Tolerable
- Interactivity replaced by running many experiments in parallel

1-4 weeks:

- High value experiments only
- Progress stalls

>1 Month:

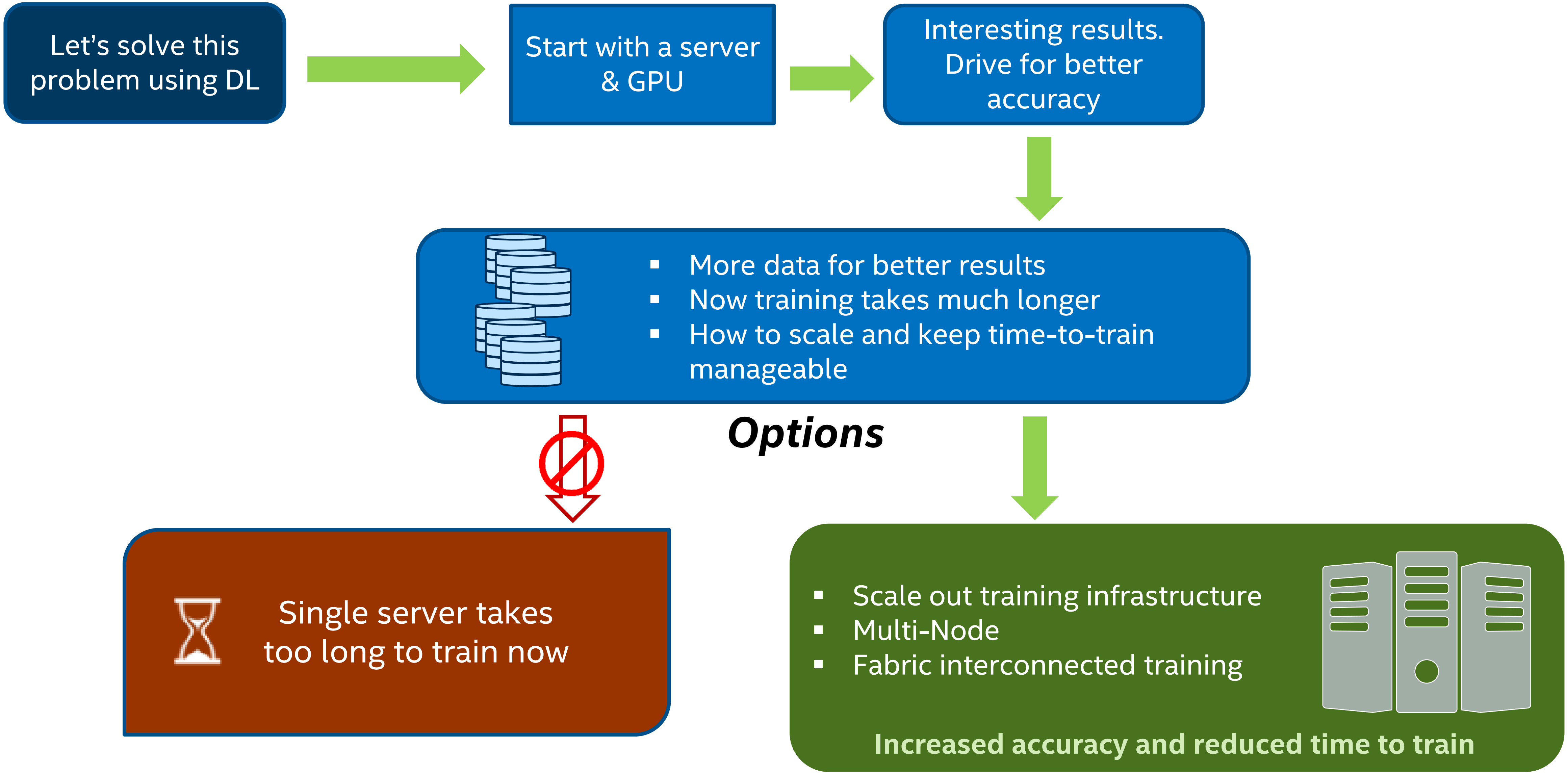
- Don't even try.

Jeff Dean: Large-Scale Deep Learning for Intelligent Computer Systems (GoogleBrain)

Striving for interactive research drives need for more computational power and multi-node training options.

[1] https://static.googleusercontent.com/media/research.google.com/en//people/jeff/BayLearn2015.pdf

Progression towards Multi-node Training



Let's solve this problem using DL

Start with a server & GPU

Interesting results. Drive for better accuracy

- More data for better results
- Now training takes much longer
- How to scale and keep time-to-train manageable

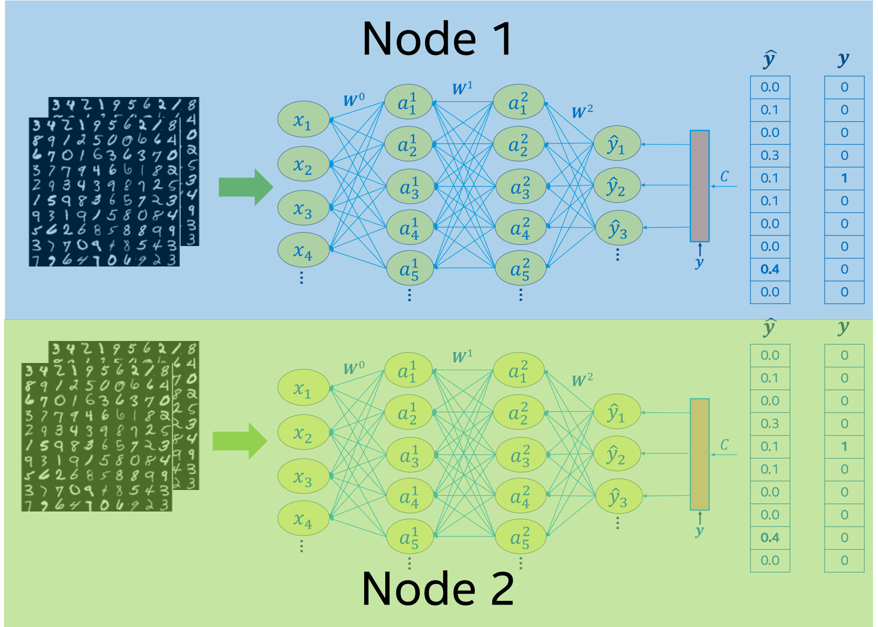
Options

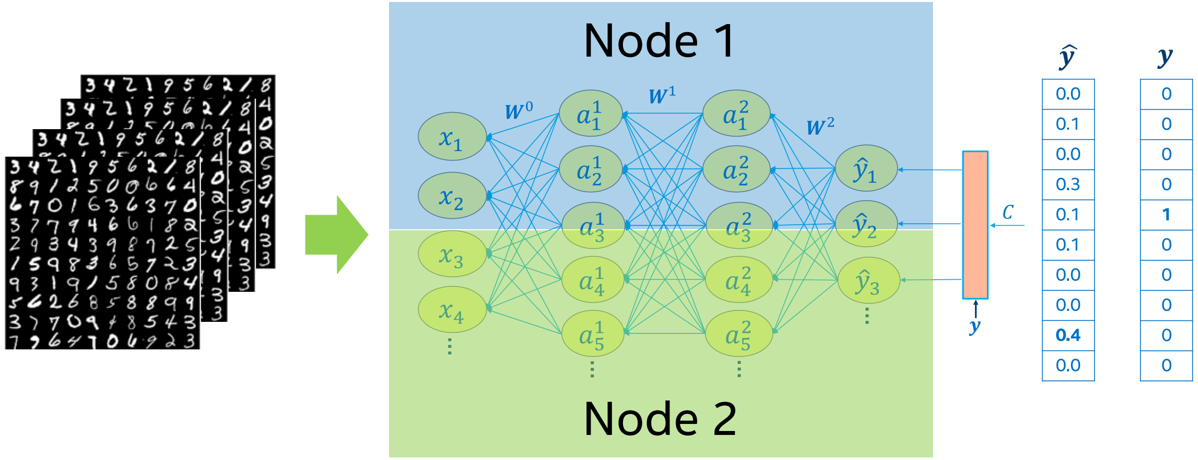
Single server takes too long to train now

- Scale out training infrastructure
- Multi-Node
- Fabric interconnected training

Increased accuracy and reduced time to train

How can Deep Learning Training be Parallelized?


Node 1
x_1 x_2 x_3 x_4
W^0 W^1 W^2
a^1_1 a^1_2 a^1_3 a^1_4 a^1_5
a^2_1 a^2_2 a^2_3 a^2_4 a^2_5
$\hat{y}_1$ $\hat{y}_2$ $\hat{y}_3$
C
y
$\hat{y}$: 0.0 0.1 0.0 0.3 0.1 0.1 0.0 0.0 0.4 0.0
y: 0 0 0 0 1 0 0 0 0 0
Node 2


Node 1
x_1 x_2 x_3 x_4
W^0 W^1 W^2
a^1_1 a^1_2 a^1_3 a^1_4 a^1_5
a^2_1 a^2_2 a^2_3 a^2_4 a^2_5
$\hat{y}_1$ $\hat{y}_2$ $\hat{y}_3$
C
y
$\hat{y}$: 0.0 0.1 0.0 0.3 0.1 0.1 0.0 0.0 0.4 0.0
y: 0 0 0 0 1 0 0 0 0 0
Node 2

Data parallelism

- Each system runs on its own dataset
- Communication occurs at each iteration, but less frequently as model parallelism
- Fast, non-blocking communication best to insure computation is not waiting on data.

Model parallelism

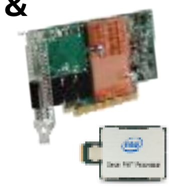
- Share the Neural Network across many nodes
- Communication occurs for layers in each iteration; creates lots of communication
- Only as fast as slowest machine due to interactivity of code

Intel® Omni-Path Architecture & AI

World-Class Interconnect Solution for Shorter Time to Train

HFI Adapters & Integrated Processors

Single port
x8 and x16



Edge Switches

1U Form Factor
24 and 48 port

Director Switches

QSFP-based
192 and 768 port

Software

Open Source
Host Software and Fabric Manager

Cables

Third Party Vendors
Passive Copper
Active Optical

Fabric interconnect for breakthrough performance on scale-out workloads like deep learning training

OMNI-PATH BENEFITS FOR AI

- High bandwidth (100Gbps), low latency and high message rate
- Improve performance, reliability & QoS:
  - ➢ Traffic Flow Optimization to maximize QoS in mixed traffic
  - ➢ Packet Integrity Protection for rapid and transparent recovery of transmission errors
  - ➢ Dynamic lane scaling to maintain link continuity
  - ➢ Switches offered with redundancy and hot swappable FRUs
- Heterogeneous cluster support
- Excellent price/performance & price/port, 48 radix
- Dispersive routing
- Multi-Rail support to provide high injection rate
- CPU/OPA integration for increased performance, lower power & cost

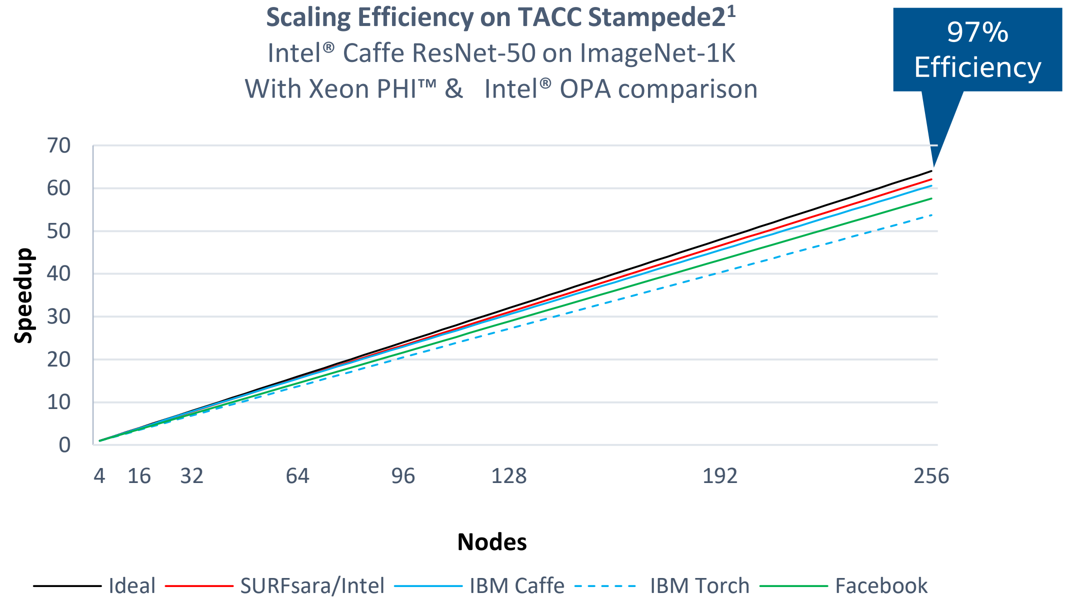
BREAKTHROUGH PERFORMANCE

- ImageNet-1K training in **less than** 40 minutes with Intel® Caffe[2]
- 90%+ efficiency with Resnet-50 training to 256 nodes and beyond on Xeon® and Xeon® Phi™
- Reduce time to train
- #1 Green500 (June 2017) cluster designed with AI in mind
- Reduced communication latency compared to InfiniBand EDR[1]:
  - ➢ Up to **21%** Higher Performance, lower latency at scale
  - ➢ Up to **17%** higher messaging rate
  - ➢ Up to **9%** higher application performance

[1] Intel® Xeon® Processor E5-2697A v4 dual-socket servers with 2133 MHz DDR4 memory. Intel® Turbo Boost Technology and Intel® Hyper Threading Technology enabled. BIOS: Early snoop disabled, Cluster on Die disabled, IOU non-posted prefetch disabled, Snoop hold-off timer=9. Red Hat Enterprise Linux Server release 7.2 (Maipo). Intel® OPA testing performed with Intel Corporation Device 24f0 – Series 100 HFI ASIC (B0 silicon). OPA Switch: Series 100 Edge Switch – 48 port (B0 silicon). Intel® OPA host software 10.1 or newer using Open MPI 1.10.x contained within host software package. EDR IB* testing performed with Mellanox EDR ConnectX-4 Single Port Rev 3 MCX455A HCA. Mellanox SB7700 - 36 Port EDR Infiniband switch. EDR tested with MLNX_OFED_Linux-3.2.x. OpenMPI 1.10.x contained within MLNX HPC-X. Message rate claim: Ohio State Micro Benchmarks v. 5.0. osu_mbw_mr, 8 B message (uni-directional), 32 MPI rank pairs. Maximum rank pair communication time used instead of average time, average timing introduced into Ohio State Micro Benchmarks as of v3.9 (2/28/13). Best of default, MXM_TLS=self,rc, and -mca pml yalla tunings. All measurements include one switch hop. Latency claim: HPCC 1.4.3 Random order ring latency using 16 nodes, 32 MPI ranks per node, 512 total MPI ranks. Application claim: GROMACS version 5.0.4 ion_channel benchmark. 16 nodes, 32 MPI ranks per node, 512 total MPI ranks. Intel® MPI Library 2017.0.064. Additional configuration details available upon request. 2. https://blog.surf.nl/en/imagenet-1k-training-on-intel-xeon-phi-in-less-than-40-minutes

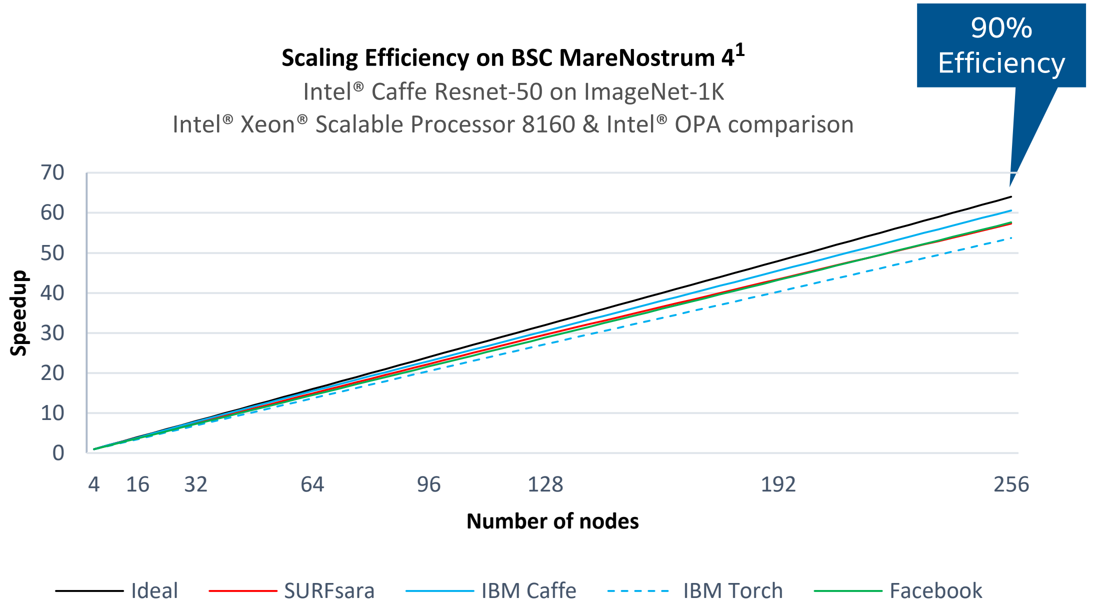
Multi-Node Intel® Caffe Resnet-50 on ImageNet-1K:

High Scaling Efficiency with Intel® OPA



Scaling Efficiency on TACC Stampede2[1]
Intel® Caffe ResNet-50 on ImageNet-1K
With Xeon PHI™ & Intel® OPA comparison

97% Efficiency



Scaling Efficiency on BSC MareNostrum 4[1]
Intel® Caffe Resnet-50 on ImageNet-1K
Intel® Xeon® Scalable Processor 8160 & Intel® OPA comparison

90% Efficiency

- **TACC Stampede 2**
  - **97% scaling efficiency** from 4 to 256 Intel® Xeon Phi™ 7250 nodes interconnected with Intel® OPA
  - Convergence with Top1/5 > **74%/92%**
  - 4 - 256 node runs: batch size of 16 per node, scaling efficiency of 97% in **63 minutes**

- **BSC MareNostrum 4**
  - Convergence with Top1/5 > **74%/92%**
  - 4 - 256 node runs: Batch size of 32 per node, 90% scaling efficiency, Total time to train: 70 Minutes

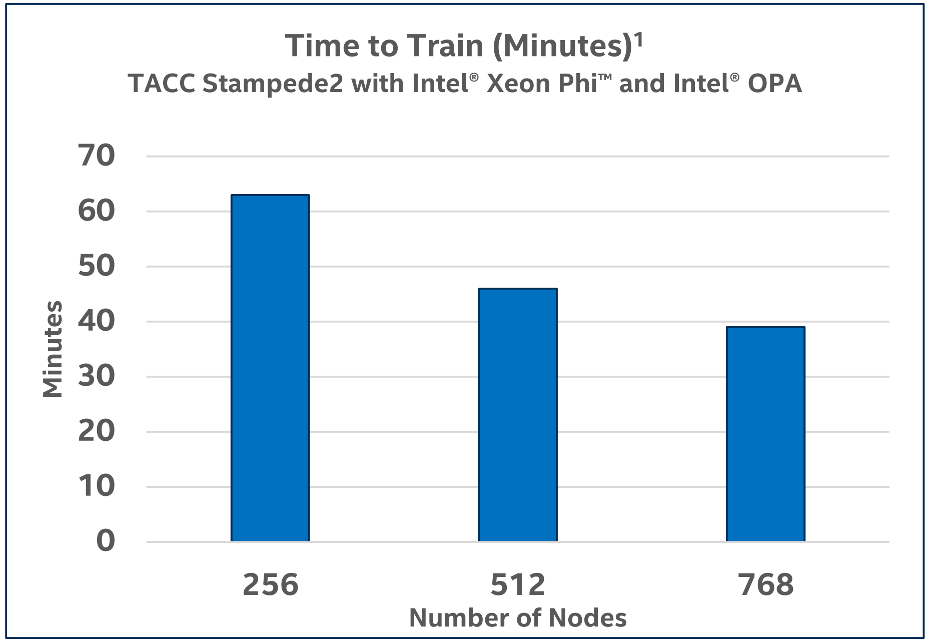
Strong multi-node training, with high accuracy with Intel® OPA

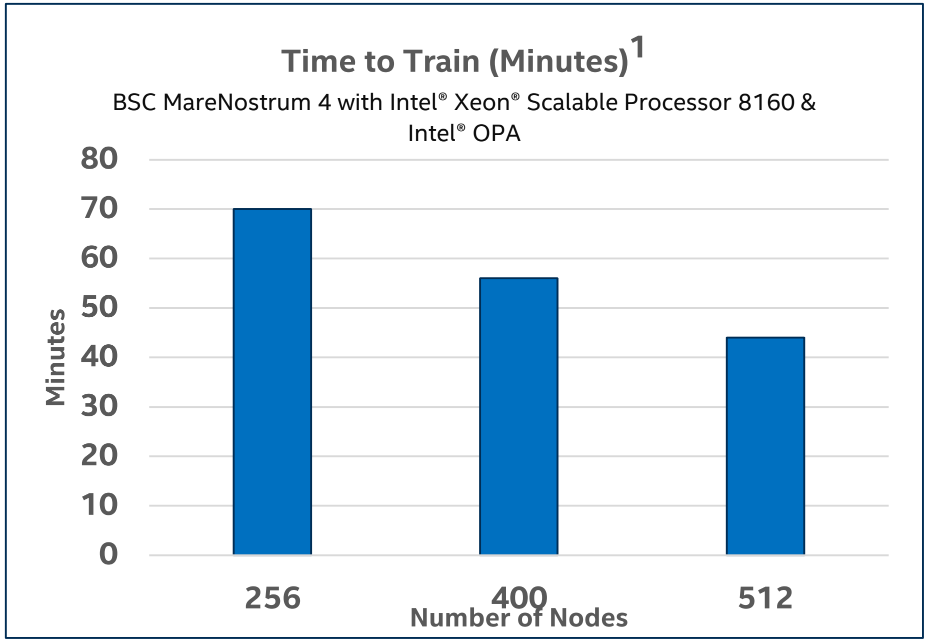
[1] https://blog.surf.nl/en/imagenet-1k-training-on-intel-xeon-phi-in-less-than-40-minutes/

More Information
- https://www.bsc.es/user-support/mn4.php
- http://portal.tacc.utexas.edu/user-guides/stampede2
- Goyal, Priya, et al. “Accurate, Larg Minibatch SGD: Training ImagNetin 1 Hour.” arXiv preprint arXiv:1706.02677 (2017)
- Cho, Minsik, et al. “powerAI DDL.” arXiv preprint arXiv:1708.02188 (2017)
- IBM claims 95% scaling efficiency and Facebook claims 89%

Multi-Node Intel® Caffe Resnet-50 on ImageNet-1K:

Time to Train on Intel® Xeon® and Xeon Phi™ with Intel® OPA





- Scaling continues, up to 768 nodes
- Efficiency reduced due to number of iterations (epochs) needed to achieve 74%/92% Top1/Top5 accuracy

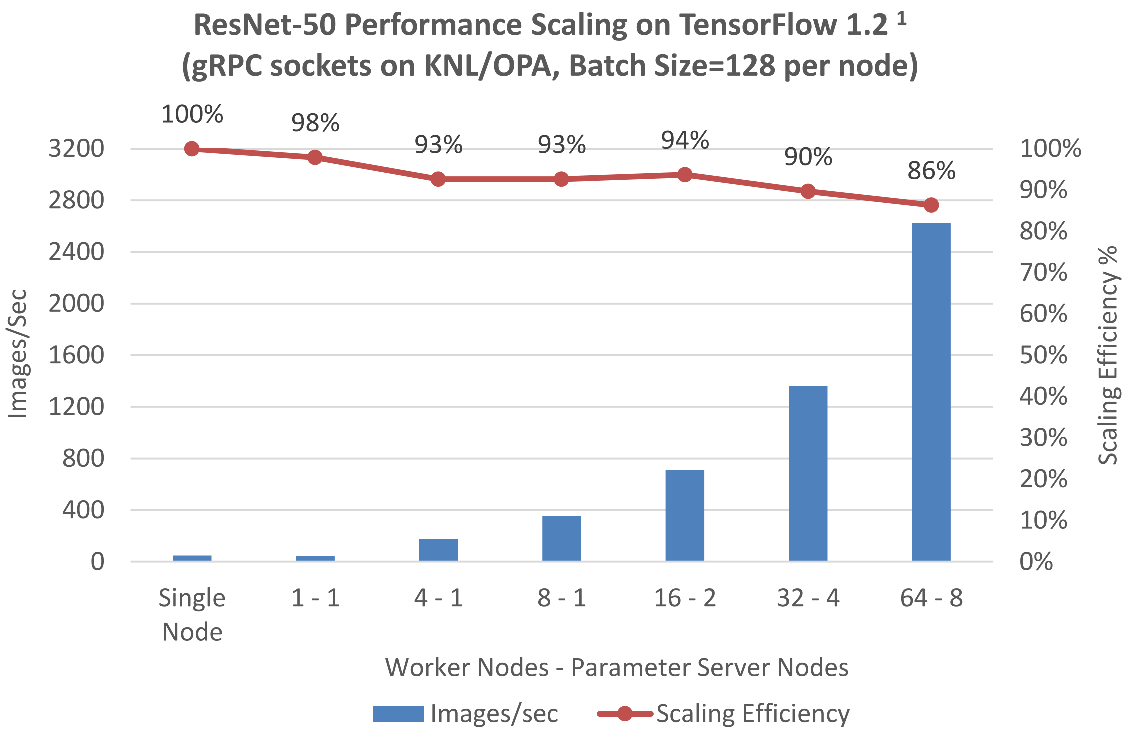
1. https://blog.surf.nl/en/imagenet-1k-training-on-intel-xeon-phi-in-less-than-40-minutes/

More information

- https://www.bsc.es/user-support/mn4.php
- http://portal.tacc.utexas.edu/user-guides/stampede2

TensorFlow on OPA

- OPA based systems support TensorFlow with any of gRPC, Verbs and MPI communications interfaces
- Preliminary multi-node measurements with Resnet-50 show good scaling with TensorFlow (using gRPC & sockets)
- Measurement and tuning efforts with various models and MPI are making good progress
- MPI Multi-End Point capability for TensorFlow in PoC stage

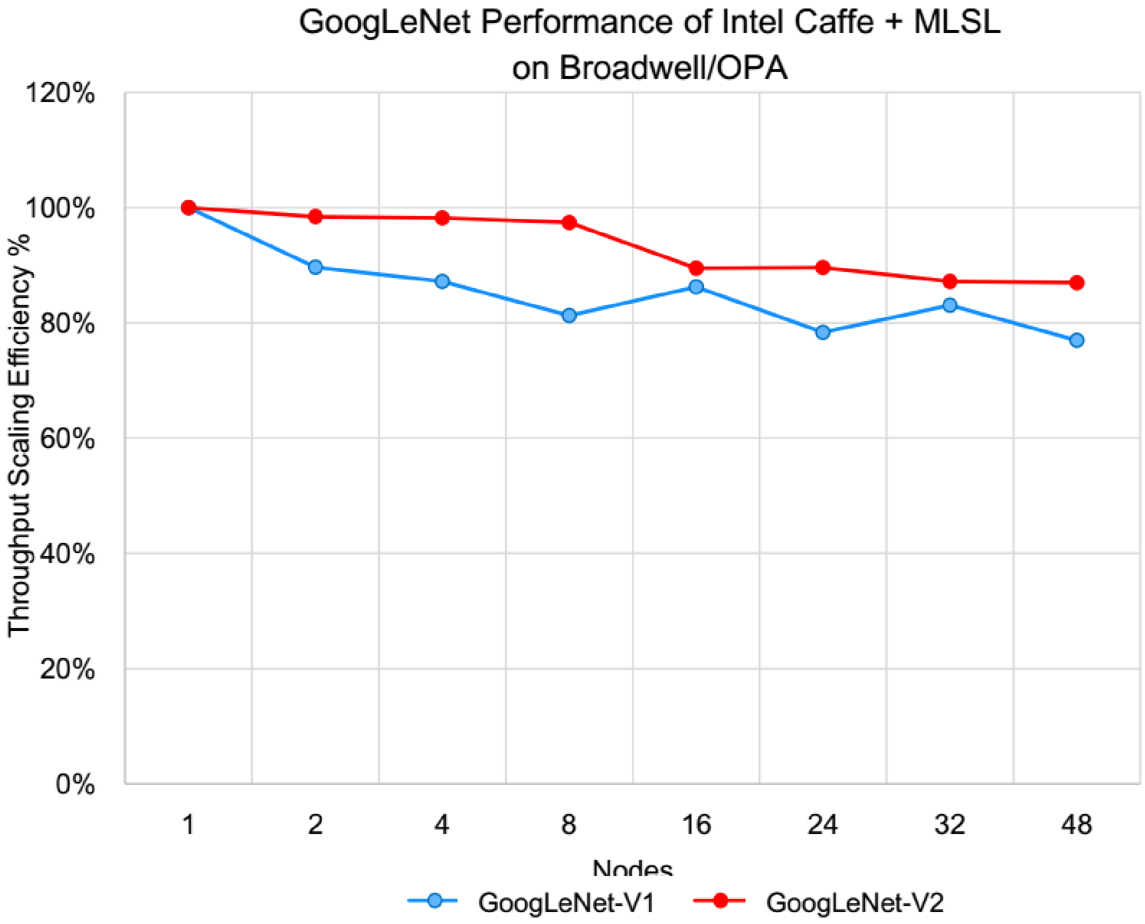


Source: Intel, Aug 2017

[1] Pre-released Intel Xeon Phi Processor codenamed Knights Landing-Fabric (KNL-F) w/integrated Omni-Path Architecture. B0 Stepping. QS upto 68 cores. 1.40 GHz. 98GB (6x16B) DDR4-2400 RDIMMS. OmniPath (OPA) Si 100 series. 48 port OPA switch with dual leaf switches per rack. 48 nodes per rack, 24 spine switches. Oracle Linux Server release 7.3. Kernel:3.10.0-514.6.2.01.el7_x86_64.knl1.

Multi-node GoogLeNet comparison with Intel® Caffe on Intel® Omni-Path Architecture (Intel® OPA)

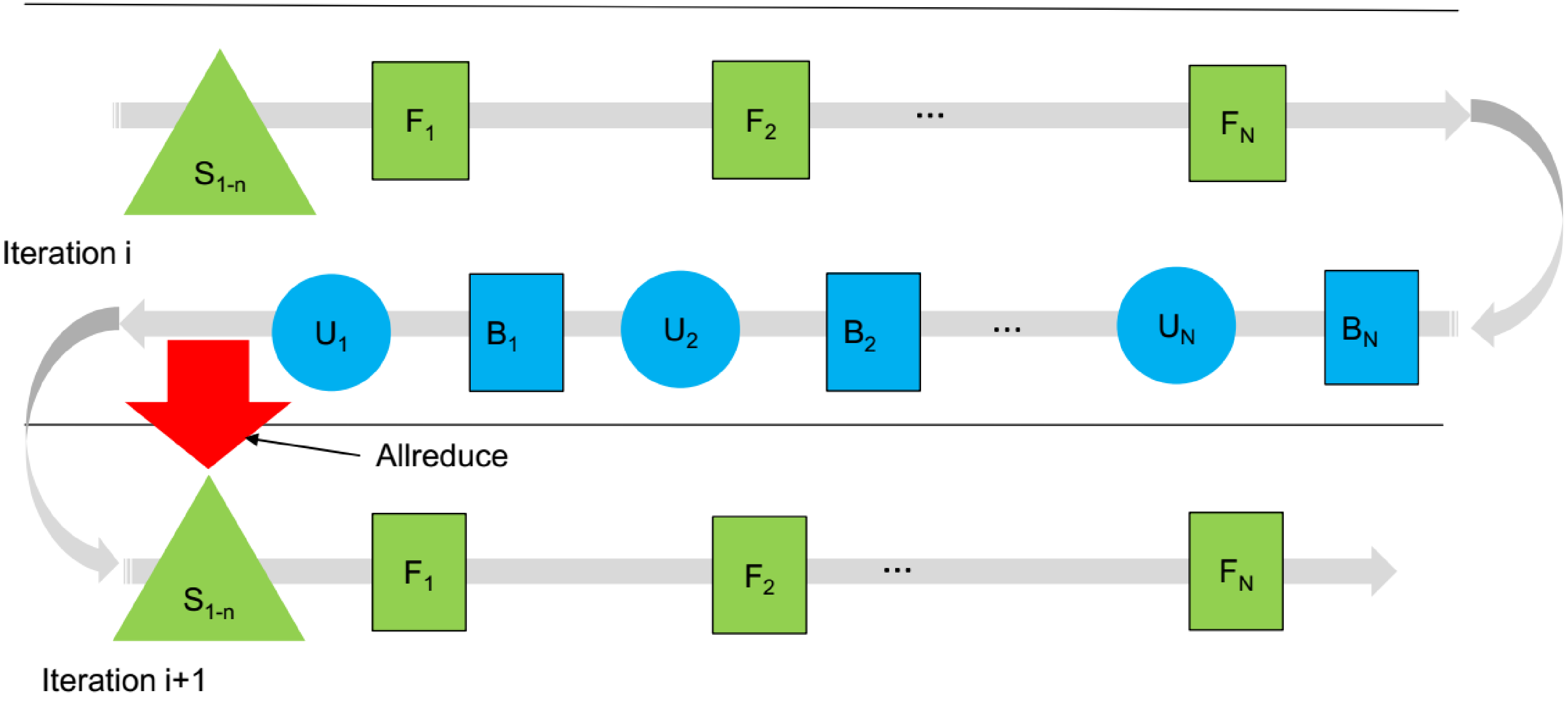
- Intel® Caffe and MLSL have been optimized for Intel® Xeon® processors & Intel® OPA
- Parallelization using Machine Learning Scaling Library (MLSL)
  - MLSL abstracts communication patterns and supports data/model/hybrid parallelism
  - Leverages Intel MPI optimizations for OPA for communication but could use other runtimes/message layers
  - MLSL API is being designed to be applicable to variety of popular Frameworks (e.g. Caffe, Torch, Theano, etc)
  - MLSL also provides statistical data collection to monitor time spent on different operations; including computation and communication.



Intel® Xeon® Processor E5-2697A v4 dual-socket servers with 64GB 2133 MHz DDR4 memory per node. 2 cores for MLSL and 30 MPI ranks per node. Intel® Turbo Boost and Hyper-Threading technology enabled. Red Hat Enterprise Linux* Server release 7.2 (Maipo). Intel® Parallel Studio XE 2017.4.056, Intel MPI 2017.3.196 MLSL 2017 Update 1 https://github.com/intel/MLSL/releases/tag/v2017.1-Preview Intel Caffe : https://github.com/intel/caffe

Naïve Multi-Node Implementation

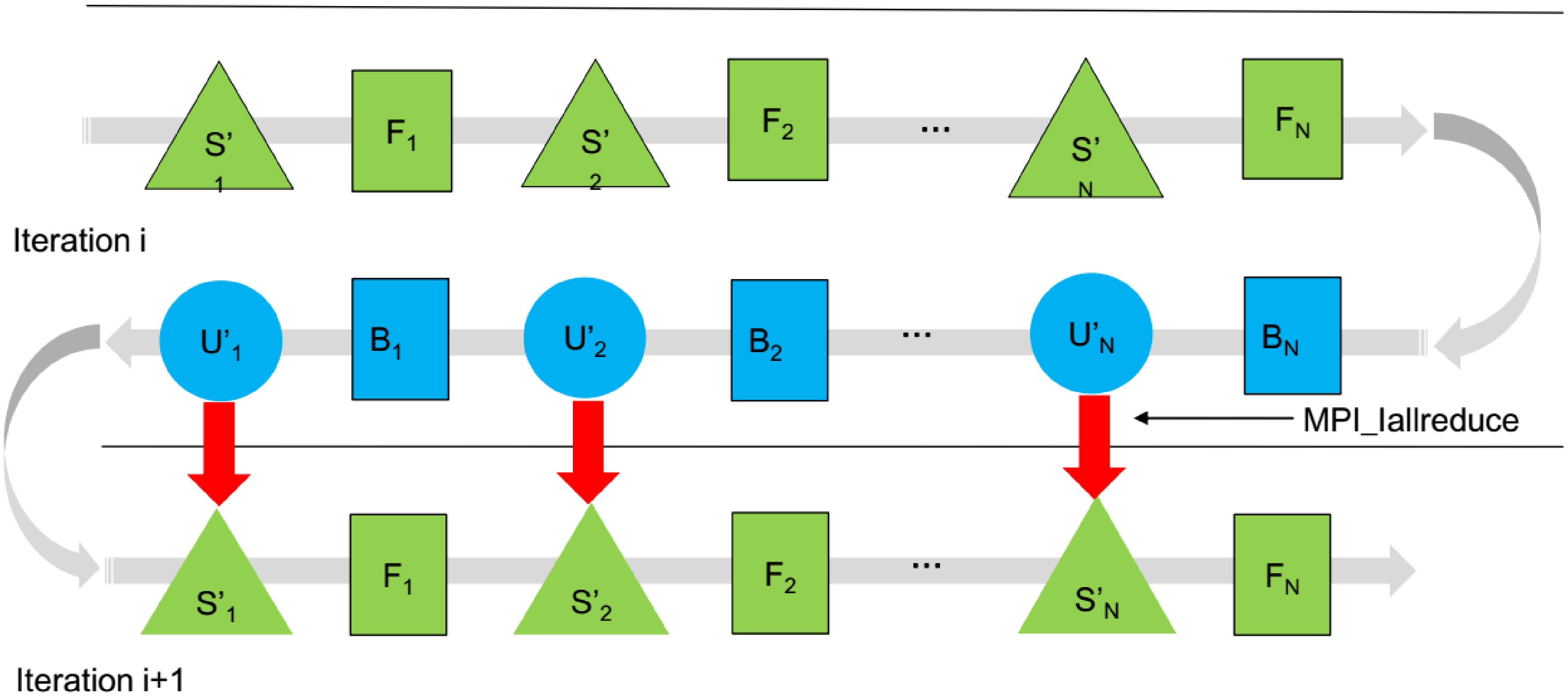
- Global averaging (usually Allreduce) of weights for all layers performed across nodes either :
  - At beginning of iteration (e.g. before F_1) .
  - After weight update U_k
- No overlapping of computation & communication



F_k – Forward propagation computation for layer k
B_k – Back propagation computation for layer k
U_k – Local weight update for layer k
$S_{1\text{-}N}$ – Allreduce of weights for all layers

MLSL – Multi-node communication technique

- Overlap computation & communication by using Non-blocking Allreduce (MPI_Iallreduce) of weights after back propagation for each layer
- Note for first layer (F_1) communication cannot be hidden but is partially/totally hidden for subsequent layers



F_k – Forward propagation computation for layer k
B_k – Back propagation computation for layer k
U'_k – Weight update & MLSL servers perform non-blocking Allreduce of weights for layer k
S'_k – Synchronization (Waits) for completion of MPI_Iallreduce from U'_k

RESOURCES

INTEL® NERVANA™ AI ACADEMY

For developers, students, instructors and startups

LEARN

- Online tutorials
- Webinars
- Student kits
- Support forums

DEVELOP

- Intel Optimized Frameworks
- Exclusive access to Intel® Nervana™ DevCloud

TEACH

- Comprehensive courseware
- Hands-on labs
- Cloud compute
- Technical Support

SHARE

- Project showcase opportunities at
- Intel Developer Mesh
- Industry & Academic events

software.intel.com/ai

INTEL® NERVANA™ DEVCLOUD

Free AI cloud access for Intel® Nervana™ AI Academy members

Intel® Nervana™ DevCloud

Preregister for this free cloud compute, powered by Intel® Xeon® Scalable rocessors, for your machine learning and deep learning training and infer compute needs.

- ✓ Get started for FREE
- ✓ 4 weeks access to remote cluster of Intel® Xeon® Scalable processors
- ✓ 200 GB file storage
- ✓ Pre-configured libraries & frameworks[†]

software.intel.com/ai/DevCloud

[†] neon™ framework, Intel® Optimization for Theano, Intel® Optimization for TensorFlow*, Intel® Optimization for Caffe*, Intel® Distribution for Python* (including NumPy, SciPy, and scikit-learn*), Keras* library*

**Other names and brands may be claimed as the property of others.*

FIND OUT MORE

LEARN

Find out more at www.intelnervana.com

EXPLORE

Use Intel's performance-optimized libraries & frameworks

ENGAGE

Contact your Intel representative for help and POC opportunities

BACKUP

Configuration Details

Platform: 2S Intel® Xeon® Platinum 8180 CPU @ 2.50GHz (28 cores), HT disabled, turbo disabled, scaling governor set to “performance” via intel_pstate driver, 384GB DDR4-2666 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86_64. SSD: Intel® SSD DC S3700 Series (800GB, 2.5in SATA 6Gb/s, 25nm, MLC).

Performance measured with: Environment variables: KMP_AFFINITY='granularity=fine, compact‘, OMP_NUM_THREADS=56, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance

Deep Learning Frameworks:

- Caffe: (http://github.com/intel/caffe/), revision f96b759f71b2281835f690af267158b82b150b5c. Inference measured with “caffe time --forward_only” command, training measured with “caffe time” command. For “ConvNet” topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from https://github.com/intel/caffe/tree/master/models/intel_optimized_models (GoogLeNet, AlexNet, and ResNet-50), https://github.com/intel/caffe/tree/master/models/default_vgg_19 (VGG-19), and https://github.com/soumith/convnet-benchmarks/tree/master/caffe/imagenet_winners (ConvNet benchmarks; files were updated to use newer Caffe prototxt format but are functionally equivalent). Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2018.0.20170425. Caffe run with “numactl -l“.

- TensorFlow: (https://github.com/tensorflow/tensorflow), commit id 207203253b6f8ea5e938a512798429f91d5b4e7e. Performance numbers were obtained for three convnet benchmarks: alexnet, googlenetv1, vgg(https://github.com/soumith/convnet-benchmarks/tree/master/tensorflow) using dummy data. GCC 4.8.5, Intel MKL small libraries version 2018.0.20170425, interop parallelism threads set to 1 for alexnet, vgg benchmarks, 2 for googlenet benchmarks, intra op parallelism threads set to 56, data format used is NCHW, KMP_BLOCKTIME set to 1 for googlenet and vgg benchmarks, 30 for the alexnet benchmark. Inference measured with --caffe time -forward_only -engine MKL2017option, training measured with --forward_backward_only option.

- MxNet: (https://github.com/dmlc/mxnet/), revision 5efd91a71f36fea483e882b0358c8d46b5a7aa20. Dummy data was used. Inference was measured with “benchmark_score.py”, training was measured with a modified version of benchmark_score.py which also runs backward propagation. Topology specs from https://github.com/dmlc/mxnet/tree/master/example/image-classification/symbols. GCC 4.8.5, Intel MKL small libraries version 2018.0.20170425.

- Neon: ZP/MKL_CHWN branch commit id:52bd02acb947a2adabb8a227166a7da5d9123b6d. Dummy data was used. The main.py script was used for benchmarking , in mkl mode. ICC version used : 17.0.3 20170404, Intel MKL small libraries version 2018.0.20170425.

Configuration Details

Platform: 2S Intel® Xeon® CPU E5-2699 v4 @ 2.20GHz (22 cores), HT enabled, turbo disabled, scaling governor set to “performance” via acpi-cpufreq driver, 256GB DDR4-2133 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.10.2.el7.x86_64. SSD: Intel® SSD DC S3500 Series (480GB, 2.5in SATA 6Gb/s, 20nm, MLC).

Performance measured with: Environment variables: KMP_AFFINITY='granularity=fine, compact,1,0‘, OMP_NUM_THREADS=44, CPU Freq set with cpupower frequency-set -d 2.2G -u 2.2G -g performance

Deep Learning Frameworks:

- Caffe: (http://github.com/intel/caffe/), revision f96b759f71b2281835f690af267158b82b150b5c. Inference measured with “caffe time --forward_only” command, training measured with “caffe time” command. For “ConvNet” topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from https://github.com/intel/caffe/tree/master/models/intel_optimized_models (GoogLeNet, AlexNet, and ResNet-50), https://github.com/intel/caffe/tree/master/models/default_vgg_19 (VGG-19), and https://github.com/soumith/convnet-benchmarks/tree/master/caffe/imagenet_winners (ConvNet benchmarks; files were updated to use newer Caffe prototxt format but are functionally equivalent). GCC 4.8.5, Intel MKL small libraries version 2017.0.2.20170110.

- TensorFlow: (https://github.com/tensorflow/tensorflow), commit id 207203253b6f8ea5e938a512798429f91d5b4e7e. Performance numbers were obtained for three convnet benchmarks: alexnet, googlenetv1, vgg(https://github.com/soumith/convnet-benchmarks/tree/master/tensorflow) using dummy data. GCC 4.8.5, Intel MKL small libraries version 2018.0.20170425, interop parallelism threads set to 1 for alexnet, vgg benchmarks, 2 for googlenet benchmarks, intra op parallelism threads set to 44, data format used is NCHW, KMP_BLOCKTIME set to 1 for googlenet and vgg benchmarks, 30 for the alexnet benchmark. Inference measured with --caffe time -forward_only -engine MKL2017option, training measured with --forward_backward_only option.

- MxNet: (https://github.com/dmlc/mxnet/), revision e9f281a27584cdb78db8ce6b66e648b3dbc10d37. Dummy data was used. Inference was measured with “benchmark_score.py”, training was measured with a modified version of benchmark_score.py which also runs backward propagation. Topology specs from https://github.com/dmlc/mxnet/tree/master/example/image-classification/symbols. GCC 4.8.5, Intel MKL small libraries version 2017.0.2.20170110.

- Neon: ZP/MKL_CHWN branch commit id:52bd02acb947a2adabb8a227166a7da5d9123b6d. Dummy data was used. The main.py script was used for benchmarking , in mkl mode. ICC version used : 17.0.3 20170404, Intel MKL small libraries version 2018.0.20170425.

Configuration Details

Platform: 2S Intel® Xeon® CPU E5-2697 v2 @ 2.70GHz (12 cores), HT enabled, turbo enabled, scaling governor set to “performance” via intel_pstate driver, 256GB DDR3-1600 ECC RAM. CentOS Linux release 7.3.1611 (Core), Linux kernel 3.10.0-514.21.1.el7.x86_64. SSD: Intel® SSD 520 Series 240GB, 2.5in SATA 6Gb/s, 25nm, MLC.

Performance measured with: Environment variables: KMP_AFFINITY='granularity=fine, compact,1,0‘, OMP_NUM_THREADS=24, CPU Freq set with cpupower frequency-set -d 2.7G -u 3.5G -g performance

Deep Learning Frameworks:

- Caffe: (http://github.com/intel/caffe/), revision b0ef3236528a2c7d2988f249d347d5fdae831236. Inference measured with “caffe time --forward_only” command, training measured with “caffe time” command. For “ConvNet” topologies, dummy dataset was used. For other topologies, data was stored on local storage and cached in memory before training. Topology specs from https://github.com/intel/caffe/tree/master/models/intel_optimized_models (GoogLeNet, AlexNet, and ResNet-50), https://github.com/intel/caffe/tree/master/models/default_vgg_19 (VGG-19), and https://github.com/soumith/convnet-benchmarks/tree/master/caffe/imagenet_winners (ConvNet benchmarks; files were updated to use newer Caffe prototxt format but are functionally equivalent). GCC 4.8.5, Intel MKL small libraries version 2017.0.2.20170110.

intel®