AIDC

INTEL AI DEVCON 2018

ADVANCING AI PERFORMANCE WITH INTEL® XEON® SCALABLE SYSTEMS

BANU NAGASUNDARAM & VIKRAM SALETORE

ADVANCING AI PERFORMANCE WITH INTEL® XEON® SCALABLE

TIME TO SOLUTION
FOR PRODUCTION AI

WORKLOAD & SCALABILITY
FLEXIBILITY

MAXIMIZE PERFORMANCE
USE OPTIMIZED SW

JOURNEY TO PRODUCTION AI

DEEP LEARNING IN DATA CENTERS

INTEL AI FOCUS PILLARS

INTERSECTION OF DATA AND COMPUTE GROWTH

	DAILY BY 2020
AVERAGE INTERNET USER	1.5 GB
AUTONOMOUS VEHICLE	4 TB
CONNECTED AIRPLANE	5 TB
SMART FACTORY	1 PB
CLOUD VIDEO PROVIDER	750 PB

BUSINESS INSIGHTS

OPERATIONAL INSIGHTS

SECURITY INSIGHTS

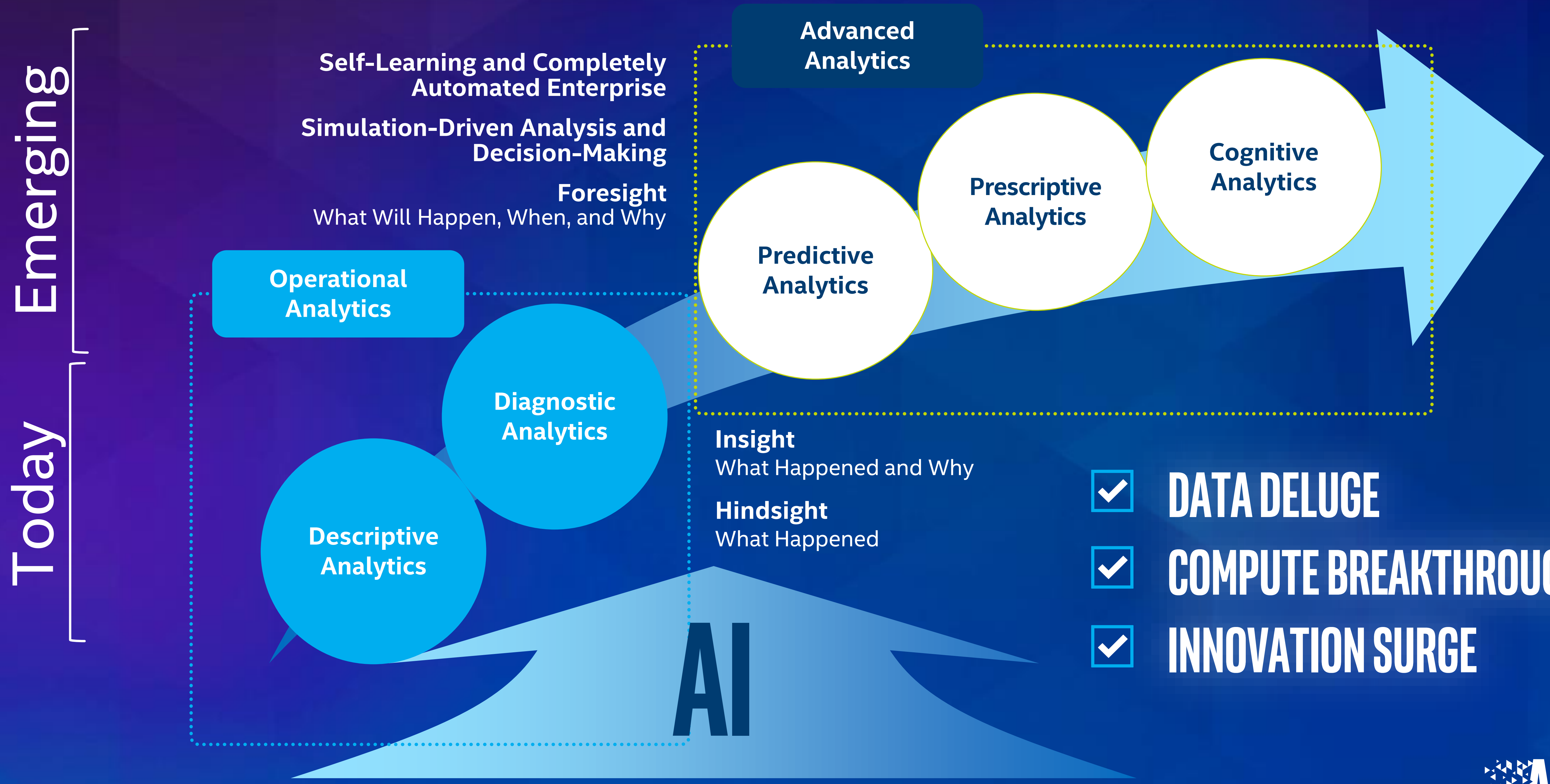
intel ATOM inside

intel CORE inside

intel XEON PLATINUM inside

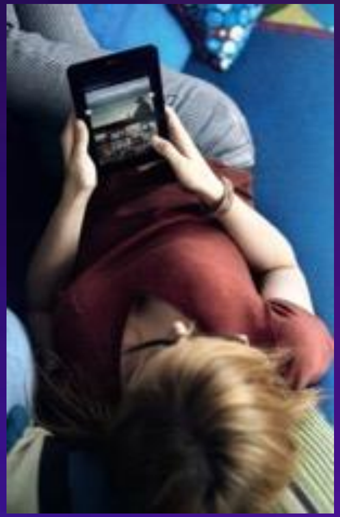
Source: Amalgamation of analyst data and Intel analysis.

DATA ANALYTICS NEEDS AI



- ☑ DATA DELUGE
- ☑ COMPUTE BREAKTHROUGH
- ☑ INNOVATION SURGE

AI WILL TRANSFORM









CONSUMER
- Smart Assistants
- Chatbots
- Search
- Personalization
- Augmented Reality
- Robots

HEALTH
- Enhanced Diagnostics
- Drug Discovery
- Patient Care
- Research
- Sensory Aids

FINANCE
- Algorithmic Trading
- Fraud Detection
- Research
- Personal Finance
- Risk Mitigation

RETAIL
- Support
- Experience
- Marketing
- Merchandising
- Loyalty
- Supply Chain
- Security

GOVERNMENT
- Defense
- Data Insights
- Safety & Security
- Resident Engagement
- Smarter Cities

ENERGY
- Oil & Gas Exploration
- Smart Grid
- Operational Improvement
- Conservation

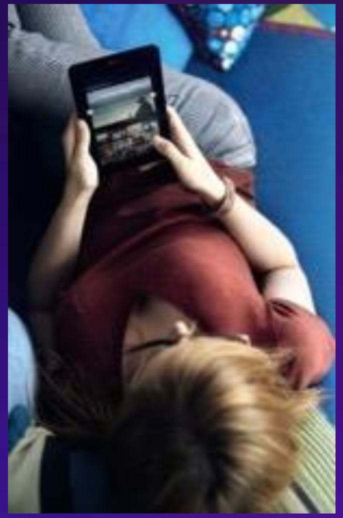
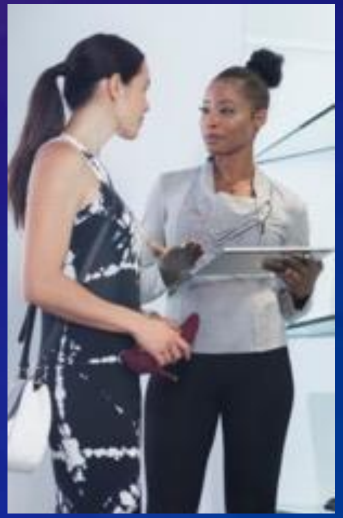
TRANSPORT
- Autonomous Cars
- Automated Trucking
- Aerospace
- Shipping
- Search & Rescue

INDUSTRIAL
- Factory Automation
- Predictive Maintenance
- Precision Agriculture
- Field Automation

OTHER
- Advertising
- Education
- Gaming
- Professional & IT Services
- Telco/Media
- Sports

Source: Intel forecast

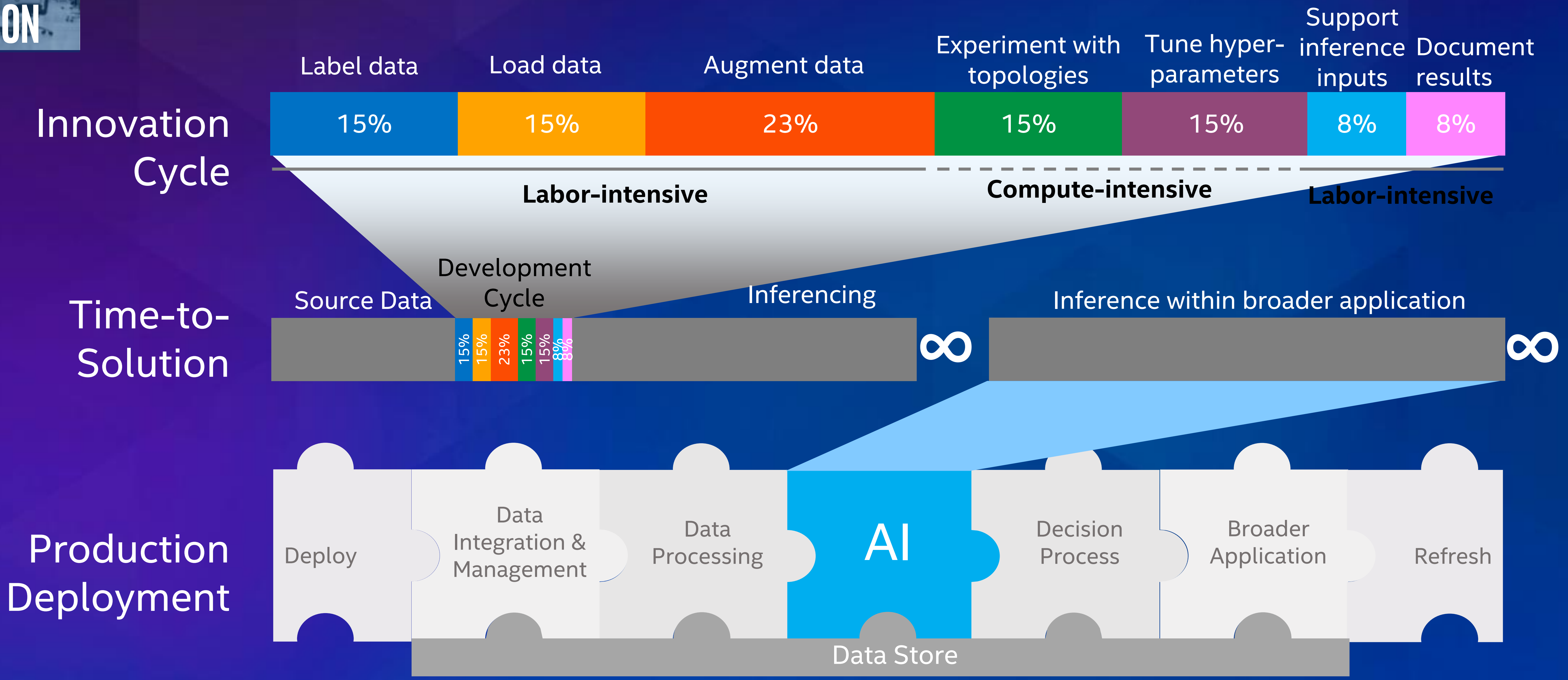
AI WITH INTEL

CONSUMER	HEALTH	FINANCE	RETAIL	GOVERNMENT	ENERGY	TRANSPORT	INDUSTRIAL	OTHER
INSURANCE PROVIDER	KYOTO UNIVERSITY	US MARKET EXCHANGE	JD.COM	NASA	LEADING OIL COMPANY	SERPRO	SOLAR FARM	INTEL PROJECTS
Recommend products to customers	Computational screen drug candidates	Time-based pattern detection	Image recognition	Furthered the search for lunar ice	Detect and classify corrosion levels	Issuing traffic tickets for violations recorded by cameras	Automate increased inspection frequency	Silicon packaging inspection
Increased accuracy [1]	Chose Xeon over GPU, decreased time & cost [2]	10x reduction in search costs [3]	4x gain by switching from GPUs to Xeon [4]	Automating lunar crater detection using a CNN [5]	Full automation [6]	Full automation [7]	Reduced time to train [8]	Significant speedup [9]

1. https://saffrontech.com/wp-content/uploads/2017/07/Personalization-Case-Study_Final-1.pdf 2. https://insidehpc.com/2016/07/superior-performance-commits-kyoto-university-to-cpus-over-gpus 3. Refer slide 60 4. https://software.intel.com/en-us/articles/building-large-scale-image-feature-extraction-with-bigdl-at-jdcom 5. Intel.ly/fdl2017 6. Refer slide 61 7. Refer slide 62 8. https://software.intel.com/en-us/articles/automatic-defect-inspection-using-deep-learning-for-solar-farm 9. https://builders.intel.com/docs/aibuilders/manufacturing-package-fault-detection-using-deep-learning.pdf For more information refer builders.intel.com/ai/solutionslibrary 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: Refer builders.intel.com/ai/solutionslibrary

DEEP LEARNING IN PRACTICE



Time-to-solution is more significant than time-to-train

INTEL AI DEVCON 2018

Source: Intel estimates based on customer data

THE JOURNEY TO PRODUCTION AI

TOTAL TIME TO SOLUTION

1 Frame Opportunity
- Define the problem
- Assess data needs

2 Hypotheses
- Identify software model needs
- Categorize existing resources

3 Data Prep
- Collect data
- Store data
- Extract, transform & load (ETL)
- Label data
- Process data
- Test data

4 Modeling
- Experiment with models
- Develop a proof of concept
- Train model
- Measure performance
- Evaluate system costs

5 Deployment
- Integrate into workflow
- Build organizational and system level capabilities to scale

6 Iteration
- Increase robustness of model with additional data
- Re-training and refine model
- Tie inference to workflow decision making
- Optimize for end-user experience

7 Evaluation
- Measure utilization
- Explore need for additional processing hardware
- Understand bottlenecks to scaling
- Assess additional revenue generation models

AIDC
INTEL AI DEVCON 2018

Source: Intel estimates based on customer data

intel AI PORTFOLIO

SOLUTIONS

Data Scientists | Technical Services | Reference Solutions

PLATFORMS

- Intel® AI Builders
- Intel® Deep Learning System‡
- intel Saffron™ — REASONING

TOOLS

- Intel® Deep Learning Studio‡
- Intel® Deep Learning Deployment Toolkit†
- OpenVino ToolKit
- Intel® Movidius™ Software Development Kit (SDK)

FRAMEWORKS

TensorFlow* Caffe* mxnet* BigDL on Spark* neon Caffe2*‡ PYTORCH*‡ Microsoft CNTK*‡ PaddlePaddle*‡

LIBRARIES

Intel® MKL/MKL-DNN, clDNN, DAAL, Intel Python Distribution, etc.

DIRECT OPTIMIZATION

Intel® nGraph™ Compiler$^{\alpha}$

CPU Transformer† | NNP Transformer‡ | Other

TECHNOLOGY

intel XEON inside | intel NERVANA inside‡ | intel ARRIA 10 inside | intel ATOM inside | intel CORE i7 inside | intel Iris Graphics | intel Movidius

END-TO-END COMPUTE

SYSTEMS & COMPONENTS

$^{\alpha}$Alpha available
†Beta available
‡ Future
*Other names and brands may be claimed as the property of others.
All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

intel AI COMPUTE

Multi-purpose to purpose-built compute for AI workloads from cloud to device

GENERAL AI

FOUNDATION FOR AI

ACCELERATED DEEP LEARNING

TRAINING

INTENSIVE TRAINING

+

MAINSTREAM TRAINING

INFERENCE

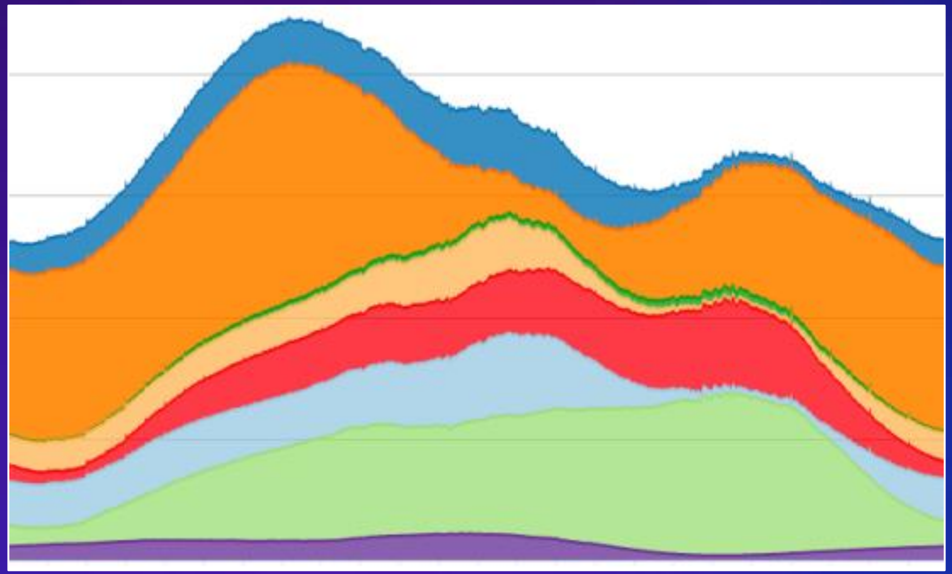
TARGETED INFERENCE

+

MAINSTREAM INFERENCE

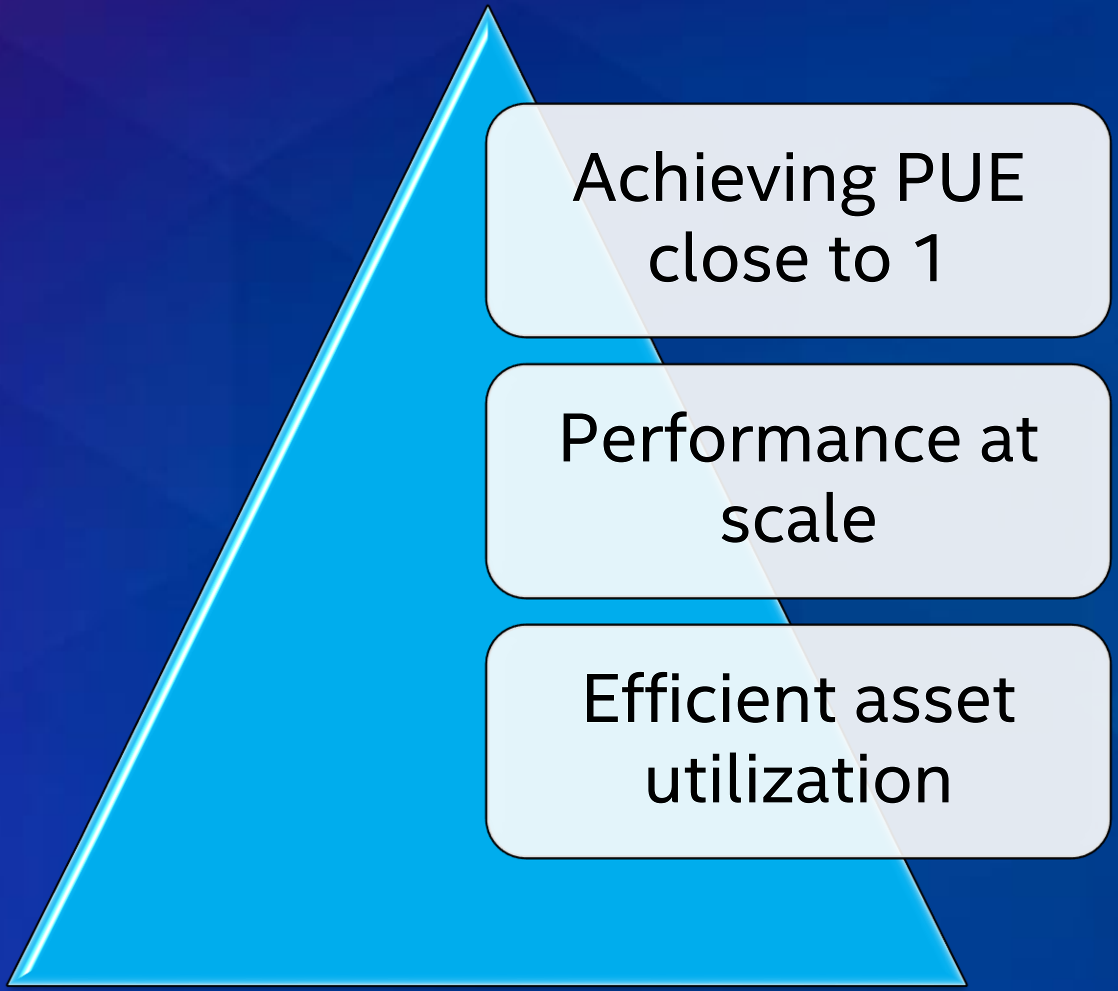
DEEP LEARNING IN DATA CENTERS

DATA CENTERS - FLEXIBILITY AND SCALABILITY

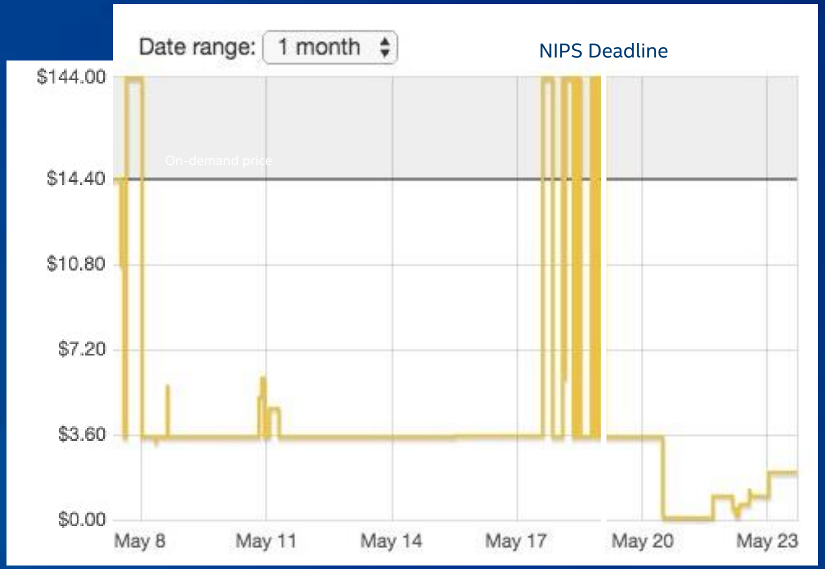


Demand load across Facebook's fleet over a 24 hour period on 19 September 2017[2]

Re-provision resources when AI developers do not need system access



Data Center Priorities



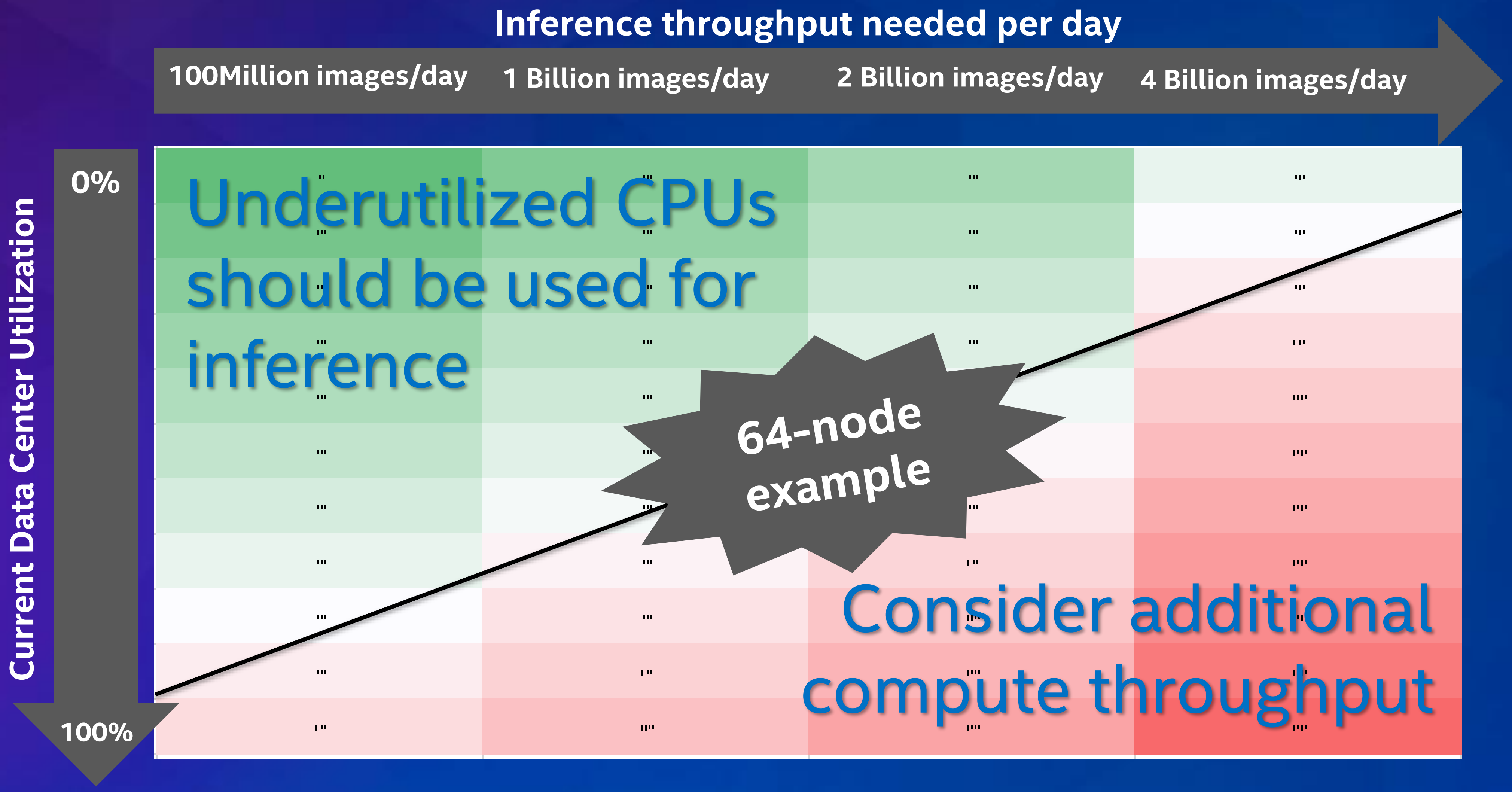
AWS Pricing Trends for p2.16xLarge Instance[4]

Provide access to multiple nodes through scalable performance when compute needs come in

Source: Facebook, Inc. Paper – Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective
The Register® https://www.theregister.co.uk/2017/05/22/cloud_providers_ai_researchers/ AWS Pricing Analysis – Screen Capture 23 May 2017

BUILT-IN ROI WITH INTEL® XEON® CLUSTERS

Workload Flexibility with Multi-Purpose CPU



Utilization estimated on 64-node cluster with Estimated performance on Caffe Resnet 50 inference throughput with 2S Intel® Xeon® Scalable Platinum 8180 Processor 1012 images/second with Int8, Refer Configuration Details 1 Performance measurements were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.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. 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 April 2018.

CPU FOR INFERENCE & TRAINING AT FACEBOOK

The abundance of readily-available CPU capacity makes it a useful platform for both training and inference. This is especially true during the off-peak portions of the diurnal cycle where CPU resources would otherwise sit idle.

Services	Ranking Algorithm	Photo Tagging	Photo Text Generation	Search	Language translation	Spam Flagging	Speech Recognition
Models	MLP	SVM,CNN	CNN	MLP	RNN	GBDT	RNN
Inference Resource	CPU	CPU	CPU	CPU	CPU	CPU	CPU
Training Resource	CPU	GPU & CPU	GPU	Depends	GPU	CPU	GPU
Training Frequency	Daily	Every N photos	Multi-Monthly	Hourly	Weekly	Sub-Daily	Weekly
Training Duration	Many Hours	Few Seconds	Many Hours	Few Hours	Days	Few Hours	Many Hours

Source: https://research.fb.com/wp-content/uploads/2017/12/hpca-2018-facebook.pdf

INTEL® XEON® SCALABLE PROCESSORS

Scalable performance for widest variety of AI & other datacenter workloads – including deep learning

intel®
XEON®
PLATINUM
inside™

THE FOUNDATION FOR AI

BUILT-IN ROI

Start your AI journey today using existing, familiar infrastructure

POTENT PERFORMANCE

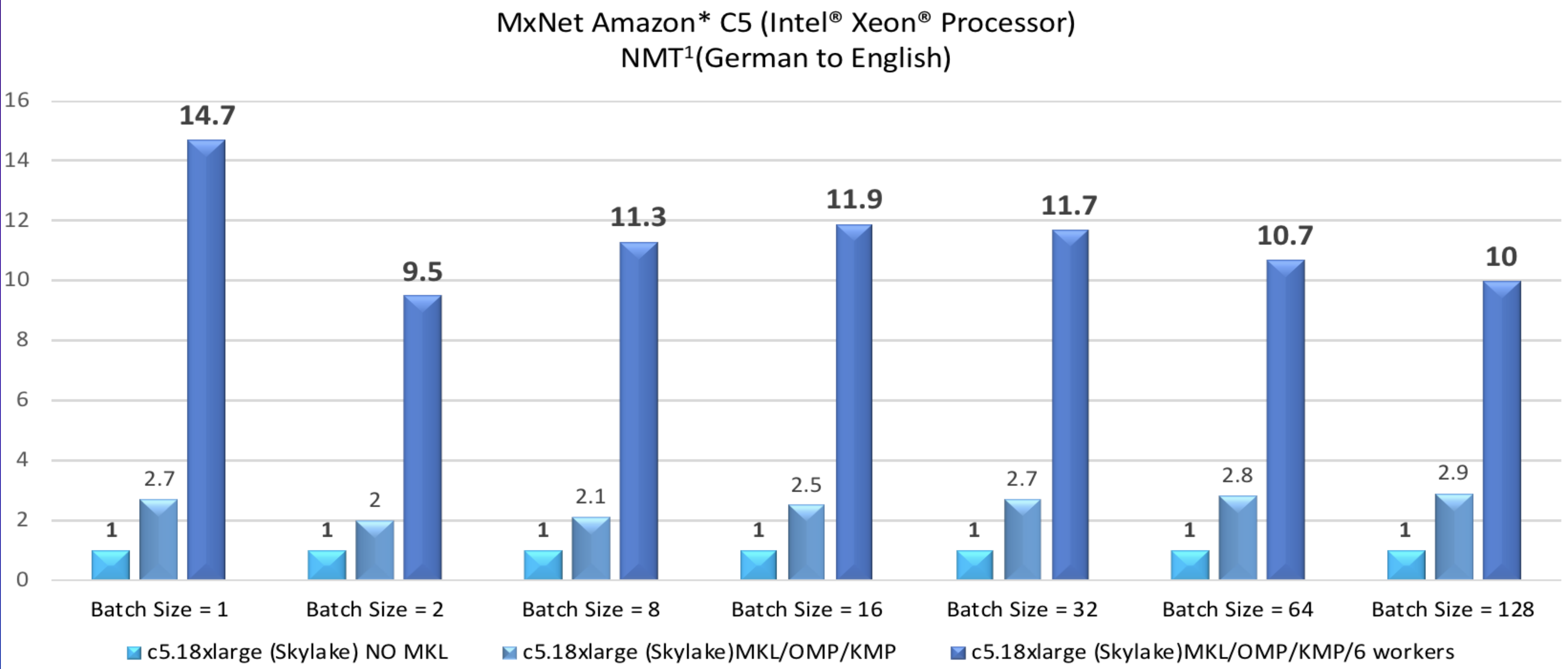
DL training in ~~days~~ HOURS;
198X[1] DL perf on 3 year HW SW refresh

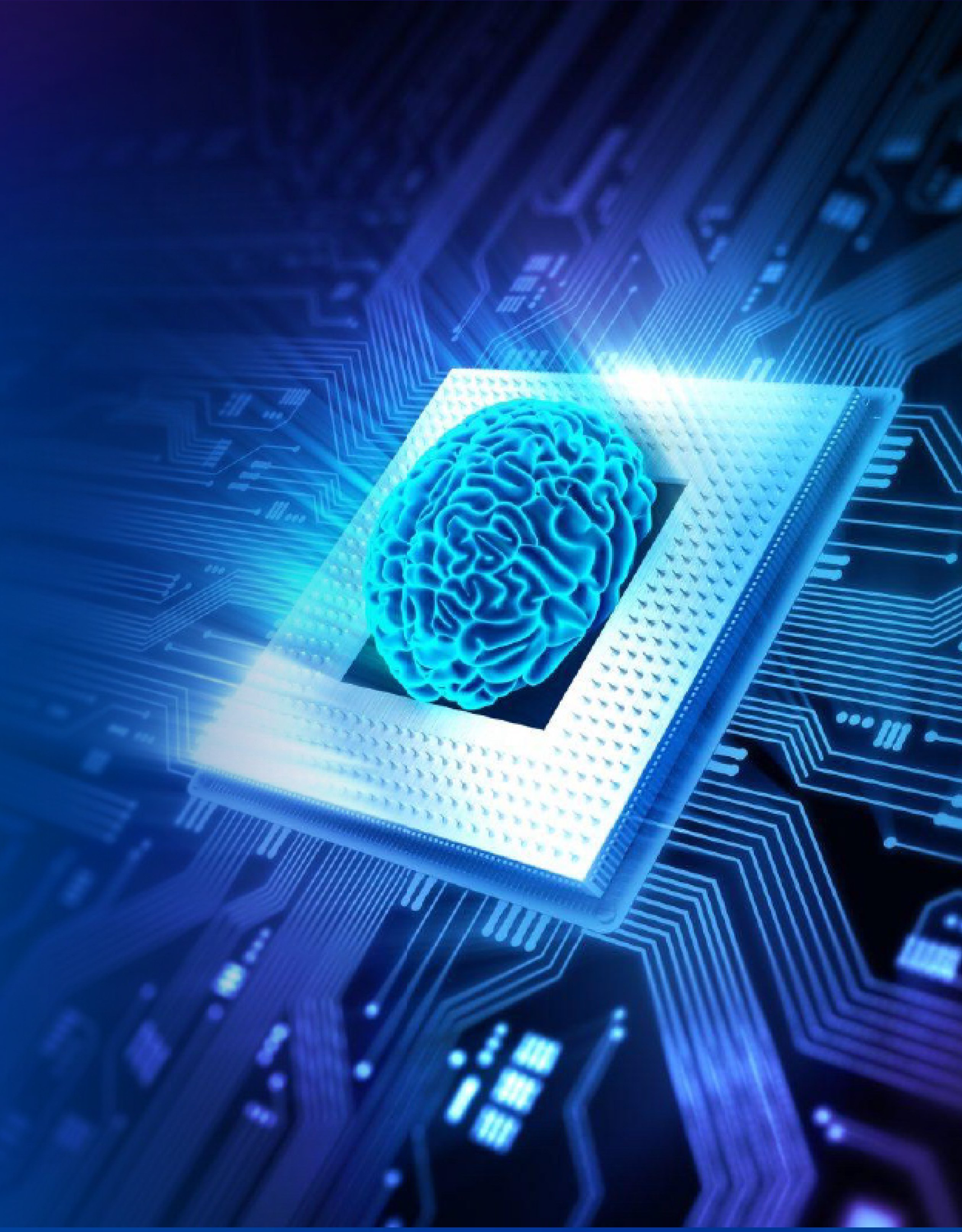
PRODUCTION-READY

Robust support for full range of AI deployments

Config 1, 14, 15 Performance estimates were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system. 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. 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 or estimated as of November 2017.

14X HIGHER INFERENCE PERFORMANCE ON INTEL® XEON® SCALABLE PROCESSORS ON NEURAL MACHINE TRANSLATION





Configuration Details 34 35 Performance measurements were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.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. 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 May 2018.

DAWNBENCH V1 BENCHMARK RESULTS

BEST INFERENCE RESULTS

FROM **INTEL** - DELIVERED IN

10MS OR LESS COMPARED TO NVIDIA P100 & K80

'For ImageNet inference, Intel submitted the best result in both cost and latency. Using an Intel optimized version of Caffe on high performance AWS instances, they reduced per image latency to 9.96 milliseconds and processed 10,000 images for $0.02'

DAWNBench

An End-to-End Deep Learning Benchmark and Competition

https://dawn.cs.stanford.edu/

https://arxiv.org/pdf/1705.07538.pdf

intel AI FOCUS PILLARS

MAXIMIZE PERFORMANCE

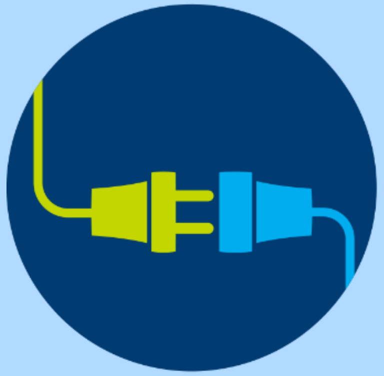
Through continuous software optimizations to libraries and frameworks

INNOVATE HARDWARE SOLUTIONS

By architecting innovative solutions to improve underlying hardware capabilities for AI

ACCELERATE DEPLOYMENTS

Speed up customer deployments through turn-key solutions for AI applications



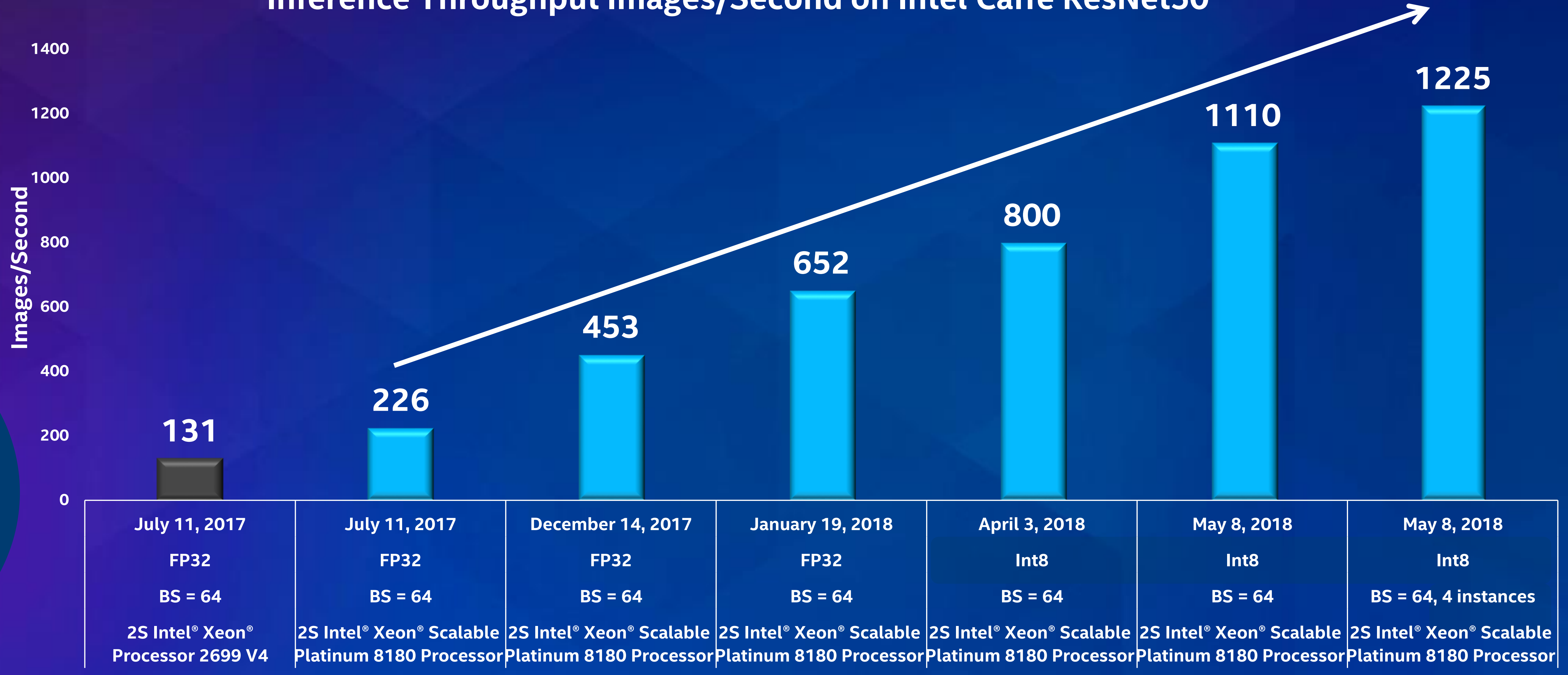
ECOSYSTEM ENABLEMENT

Partner with customers & developers on their AI journey to develop end to end solutions from edge to cloud

MAXIMIZE PERFORMANCE

INTEL® XEON® PROCESSOR AI PERFORMANCE

Inference Throughput Images/Second on Intel Caffe ResNet50



5.4X(1)

In 10 months

since Intel® Xeon® Scalable Processor launch

(1) Up to 5.4X performance improvement with software optimizations on Caffe Resnet-50 in 10 months with 2 socket Intel® Xeon® Scalable Processor, Configuration Details 1, 14, 15, 41. Performance measurements were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

Performance measurements were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

USE OPTIMIZED MKL-DNN LIBRARY

OPTIMIZED SOFTWARE : MKL-DNN LIBRARY

2D & 3D Convolution

2D & 3D Inner Product

Pooling

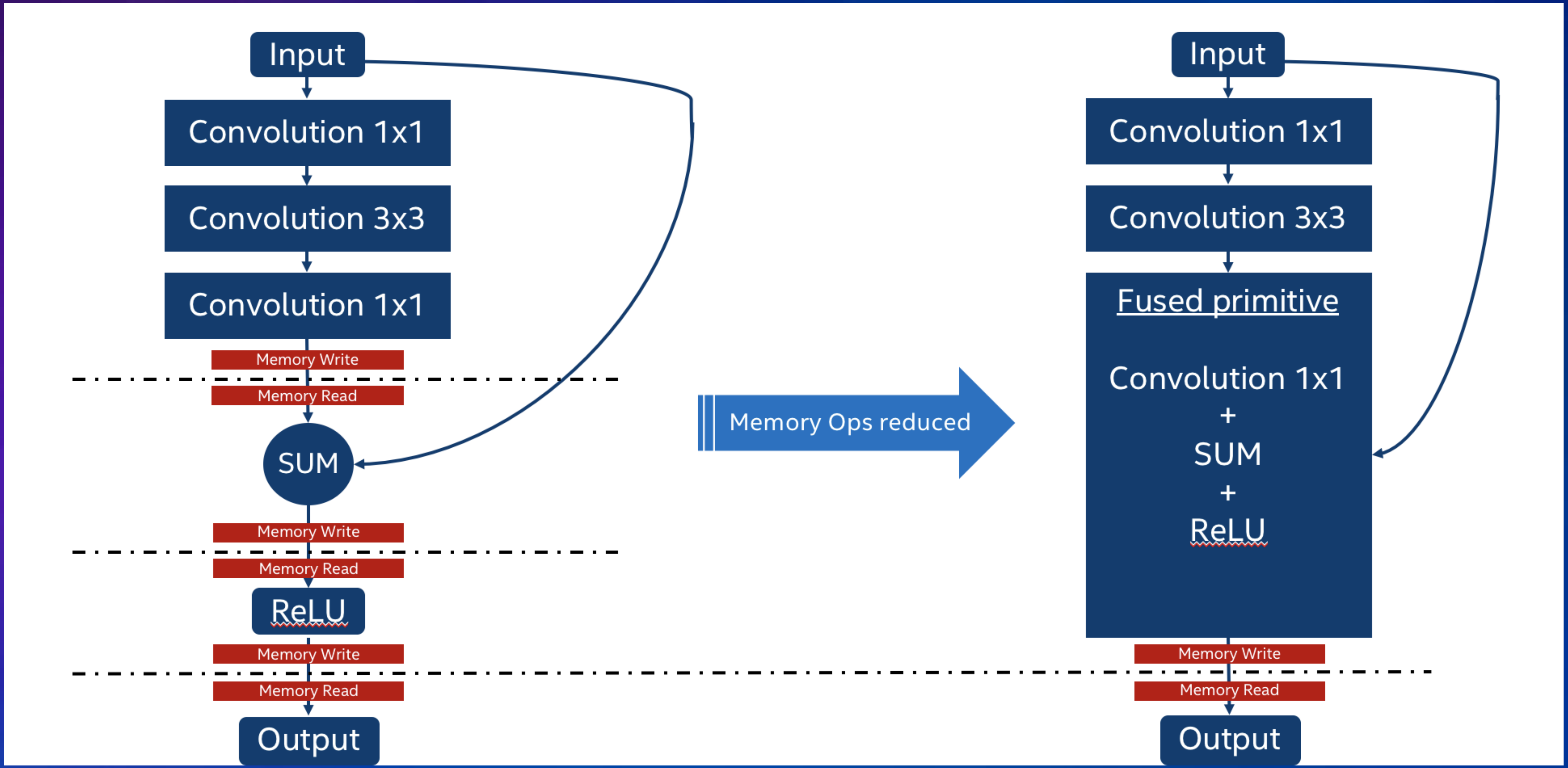
Normalization

Activation:
- ReLU (Training)
- Tanh, Logistic Regression, Softmax (Inference)

Data Manipulation: Reorder, Sum, Concat

OPTIMIZED SOFTWARE : MKL-DNN LIBRARY

Layer Fusion Example



AIDC INTEL AI DEVCON 2018

OPTIMIZED SOFTWARE : MKL-DNN LIBRARY

Winograd Algorithm to improve matrix multiply performance

Normal Matrix Multiply
6 multiplications

Y0 = i0*F0 + i1*F1 + i2*F2
Y1= i1*F1 + i2*F2 + i3*F3

$$\begin{bmatrix} i0 & i1 & i2 \\ i1 & i2 & i3 \end{bmatrix} \times \begin{bmatrix} F0 \\ F1 \\ F2 \end{bmatrix} = \begin{bmatrix} Y0 \\ Y1 \end{bmatrix}$$

Winograd Matrix
Multiply
4 multiplications

Y0 = X0 + X1 + X2
Y1= X1 - X2 - X3

X0 = (i0-i2)*F0
X1 = (i1+i2)*(F0+F1+F2)/2
X2 = (i1-i3)*F2
X3 = (i2-i1)*(F0-F1+F2)/2

Compute Operations decreased,
memory accesses increased

USE OPTIMIZED MKL-DNN LIBRARY

+

USE OPTIMIZED FRAMEWORKS

FRAMEWORK OPTIMIZATIONS

Scaling

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

Utilize all the cores

- OpenMP
- 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

Example: Load Balancing:

TensorFlow graphs offer opportunities for parallel execution. Threading model

1. **inter_op_parallelism_threads**: Max number of operators that can be executed in parallel
2. **intra_op_parallelism_threads**: Max number of threads to use for executing an operator
3. **OMP_NUM_THREADS**: MKL-DNN equivalent of intra_op_parallelism_threads

OPTIMIZED FRAMEWORK INSTALLATION

Framework	How to Access Optimized Framework
TensorFlow	Install Intel optimized wheel, see tensorflow.org page for CPU optimization instructions
MXNet	Intel optimizations in main branch via experimental path, available here
Caffe2	Will upstream to master branch in Q2
PaddlePaddle	Paddle Paddle master branch
PyTorch	Intel optimizations available in this branch
Caffe	Intel optimized version of Caffe
CNTK	Will upstream to master branch in Q2

INFERENCE

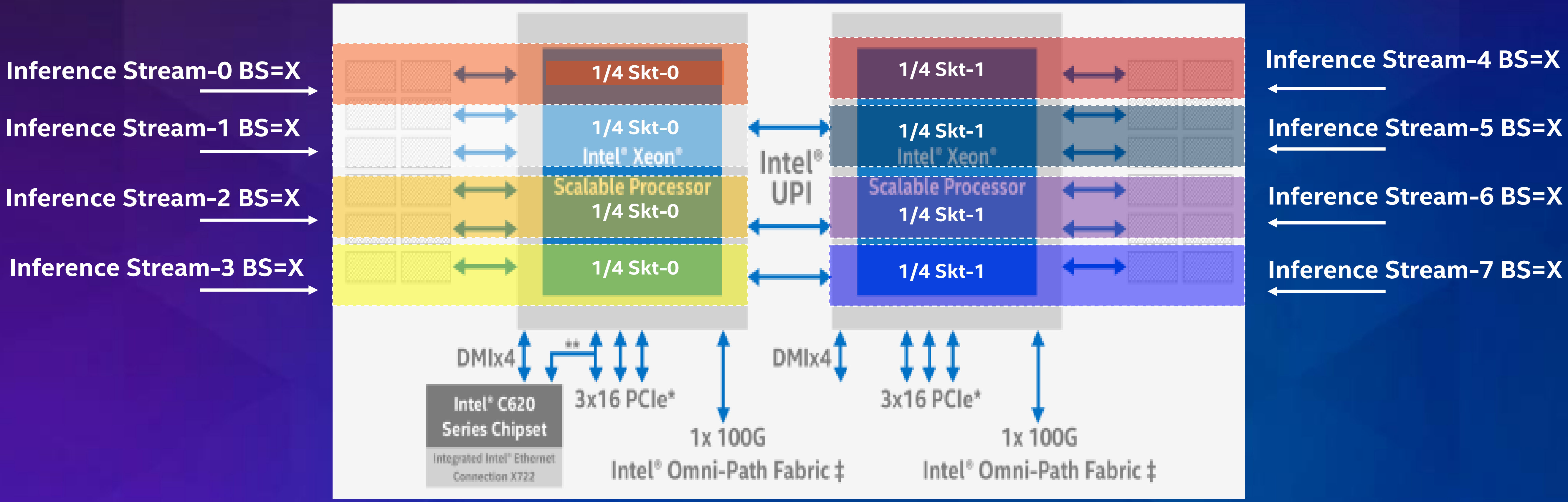
USE OPTIMIZED MKL-DNN LIBRARY

USE OPTIMIZED FRAMEWORKS

+

ENABLE MULTIPLE STREAMS

MULTIPLE INFERENCE STREAMS



Recommend using multiple framework instances
Each framework instance is pinned to a separate NUMA domain
Each instance processes a separate Inference Stream

Optimizations at run time without framework code change

Best Known Methods: https://ai.intel.com/accelerating-deep-learning-training-inference-system-level-optimizations/

INFERENCE EXAMPLE : MULTI-STREAM FOR TENSORFLOW

For 2S Intel Xeon® Platinum 8170 processor-based systems, sub-socket with 8 inference streams:

- common_args: “--model resnet50 --batch_size 64 --data_format NCHW --num_batches 100 --distortions=True --mkl=True --num_warmup_batches 10 --device cpu --data_dir ~/tensorflow/TF_Records --data_name imagenet --display_every 10“
- WK_HOST= “hostname”
- worker_env:”export OMP_NUM_THREADS=6”
- inf_args: “$common_args --num_intra_threads 6 --num_inter_threads 2“

To start 4 inference streams on Socket-0:

- ssh $WK_HOST; $worker_env; nohup unbuffer numactl -m 0 python tf_cnn_benchmarks.py --forward_only True $inf_args --kmp_affinity="granularity=thread,proclist=[0-5,52-57],explicit,verbose" &
- ssh $WK_HOST; $worker_env; nohup unbuffer numactl -m 0 python tf_cnn_benchmarks.py --forward_only True $inf_args --kmp_affinity="granularity=thread,proclist=[6-12,58-64],explicit,verbose" &
- ssh $WK_HOST; $worker_env; nohup unbuffer numactl -m 0 python tf_cnn_benchmarks.py --forward_only True $inf_args --kmp_affinity="granularity=thread,proclist=[13-18,65-70],explicit,verbose" &
- ssh $WK_HOST; $worker_env; nohup unbuffer numactl -m 0 python tf_cnn_benchmarks.py --forward_only True $inf_args --kmp_affinity="granularity=thread,proclist=[19-25,71-77],explicit,verbose" &

To start 4 inference streams on Socket-1:

- ssh $WK_HOST; $worker_env; nohup unbuffer numactl -m 1 python tf_cnn_benchmarks.py --forward_only True $inf_args --kmp_affinity="granularity=thread,proclist=[26-31,78-83],explicit,verbose" &
- ssh $WK_HOST; $worker_env; nohup unbuffer numactl -m 1 python tf_cnn_benchmarks.py --forward_only True $inf_args --kmp_affinity="granularity=thread,proclist=[32-38,84-90],explicit,verbose" &
- ssh $WK_HOST; $worker_env; nohup unbuffer numactl -m 1 python tf_cnn_benchmarks.py --forward_only True $inf_args --kmp_affinity="granularity=thread,proclist=[39-44,91-96],explicit,verbose" &
- ssh $WK_HOST; $worker_env; nohup unbuffer numactl -m 1 python tf_cnn_benchmarks.py --forward_only True $inf_args --kmp_affinity="granularity=thread,proclist=[45-51,96-102],explicit,verbose" &

3X HIGHER INFERENCE PERFORMANCE

BY USING MKL-DNN LIBRARIES + OPTIMIZED FRAMEWORK

On MxNet Amazon* C5 (Intel® Xeon® Scalable Processor) running NMT[1](German to English) with and without MKL DNN libraries

+

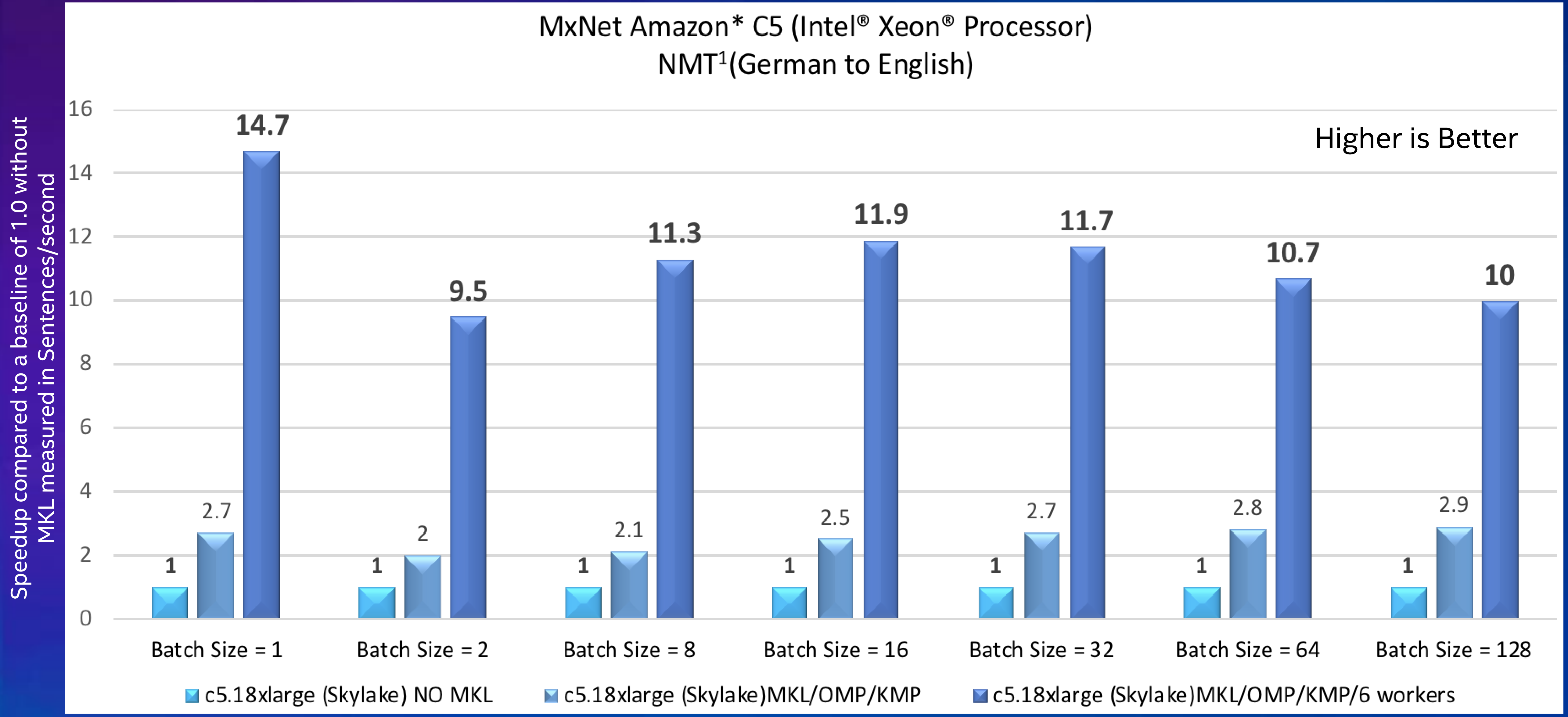
5X HIGHER INFERENCE PERFORMANCE

BY USING MULTIPLE STREAMS

On MxNet Amazon* C5 (Intel® Xeon® Scalable Processor) running NMT[1](German to English) with MKL DNN libraries comparing with and without multiple streams

[1]sockeye https://github.com/awslabs/sockeye. Measured by Intel on AWS instance. Measured as of May 2018 Configurations:34,35 Performance measurements were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system. 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. 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 April 2018.

INTEL® XEON® SCALABLE PROCESSORS PERFORMANCE ON NEURAL MACHINE TRANSLATION



[1]sockeye https://github.com/awslabs/sockeye. Measured by Intel on AWS instance. Measured as of May 2018 Configurations:34,35 Performance measurements were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system. 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. 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 April 2018.

INTEL AI DEVCON 2018

TRAINING

USE OPTIMIZED MKL-DNN LIBRARY

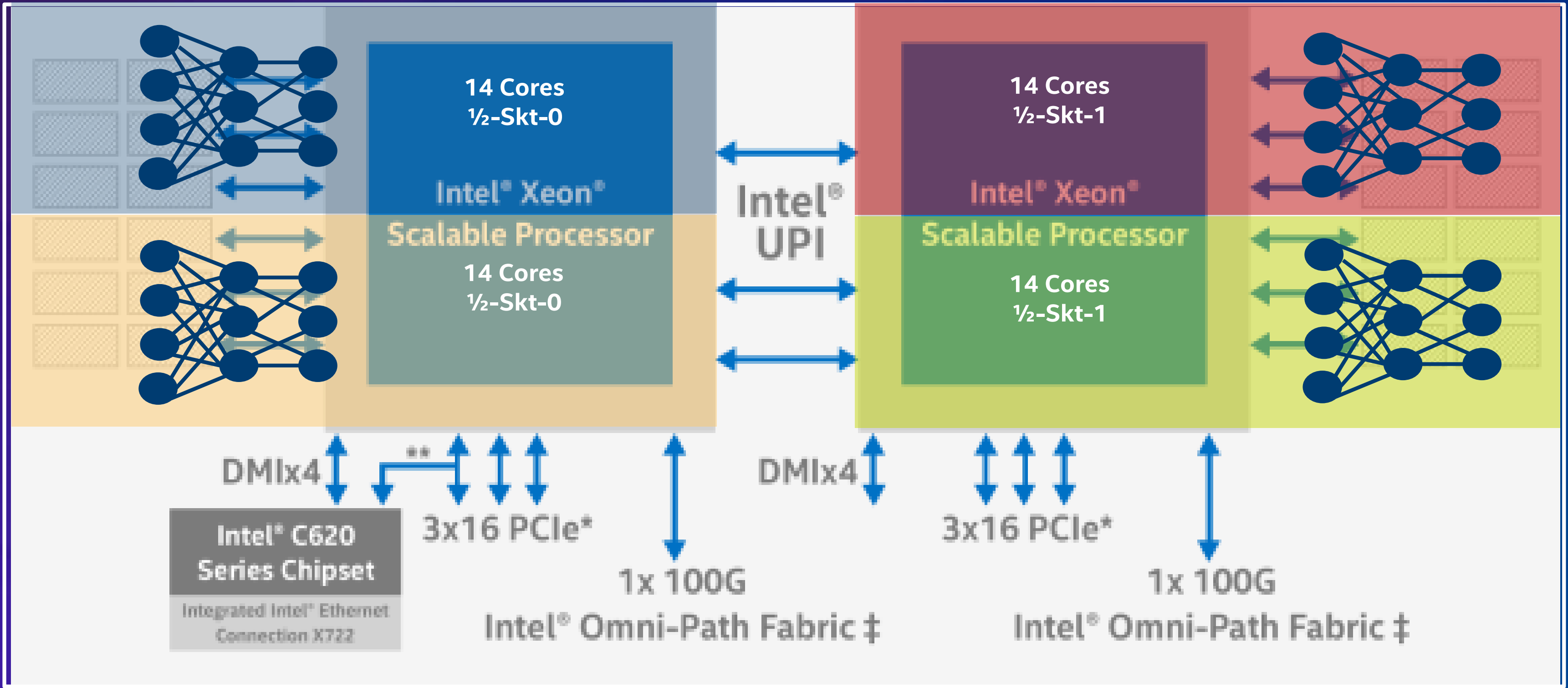
USE OPTIMIZED FRAMEWORKS

ENABLE MULTIPLE STREAMS

+

SCALEOUT TO MULTIPLE NODES

TRAINING: MULTI-WORKERS PER SOCKET



Each framework instance is pinned to a separate NUMA domain

Each CPU running 1 or more workers/node

Uses optimized MPI library for gradient updates over shared memory

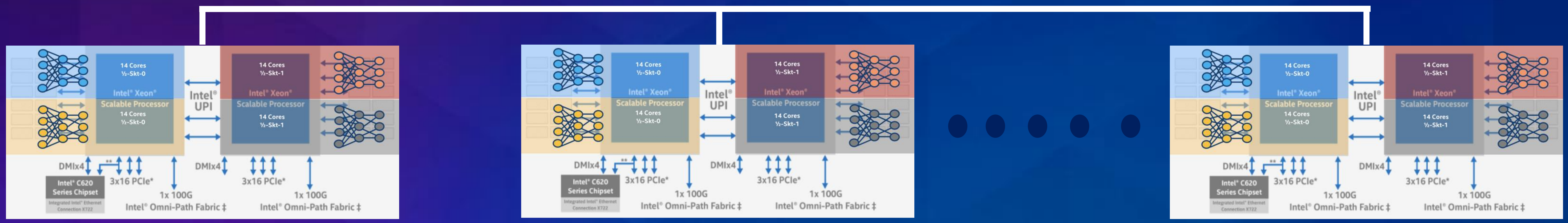
- Caffe – Use Optimized Intel® MPI ML Scaling Library (ML-SL)
- TensorFlow – Uber horovod MPI Library

Optimizations at run time <u>without</u> framework code change

Intel Best Known Methods: https://ai.intel.com/accelerating-deep-learning-training-inference-system-level-optimizations/

SCALEOUT TRAINING: MULTI-WORKERS & MULTI-NODES

Interconnect Fabric (Intel® OPA or Ethernet)



Node 1 Node 2 Node N

Distributed Deep Learning Training Across Multiple nodes

Each node running multiple workers/node

Uses optimized MPI library for gradient updates over network fabric

Caffe – Use Optimized Intel® MPI ML Scaling Library (ML-SL)

TensorFlow – Uber horovod MPI Library

Intel Best Known Methods: https://ai.intel.com/accelerating-deep-learning-training-inference-system-level-optimizations/

TRAINING EXAMPLE : MULTI WORKER/NODE FOR TENSORFLOW

For 4-Node 2S 28 Core Intel Xeon® Platinum 8180 processor based cluster with 4 Workers/Node
Total of 16 TensorFlow Workers using horovod MPI Communication Library:

OMP_NUM_THREADS=14

To start Distributed Training:

mpiexec --machinefile <hostfile> -genv **-np 16 -ppn 4** –genv OMP_NUM_THREADS $OMP_NUM_THREADS \
-genv I_MPI_PIN_DOMAIN 28:compact -genv HOROVOD_FUSION_THRESHOLD 134217728 \
python <path>/tf_cnn_benchmarks/tf_cnn_benchmarks.py --batch_size=64 --model=resnet50 \
--num_inter_threads 2 --num_intra_threads $OMP_NUM_THREADS \
--num_batches 100 --display_every 10 --data_format NCHW --optimizer momentum --device cpu --mkl=true \
--variable_update horovod --horovod_device cpu --local_parameter_device cpu \
--kmp_blocktime=1 --enable_layout_optimizer=TRUE --data_dir=<path-to-TFRecords> \
--data_name=<dataset_name>

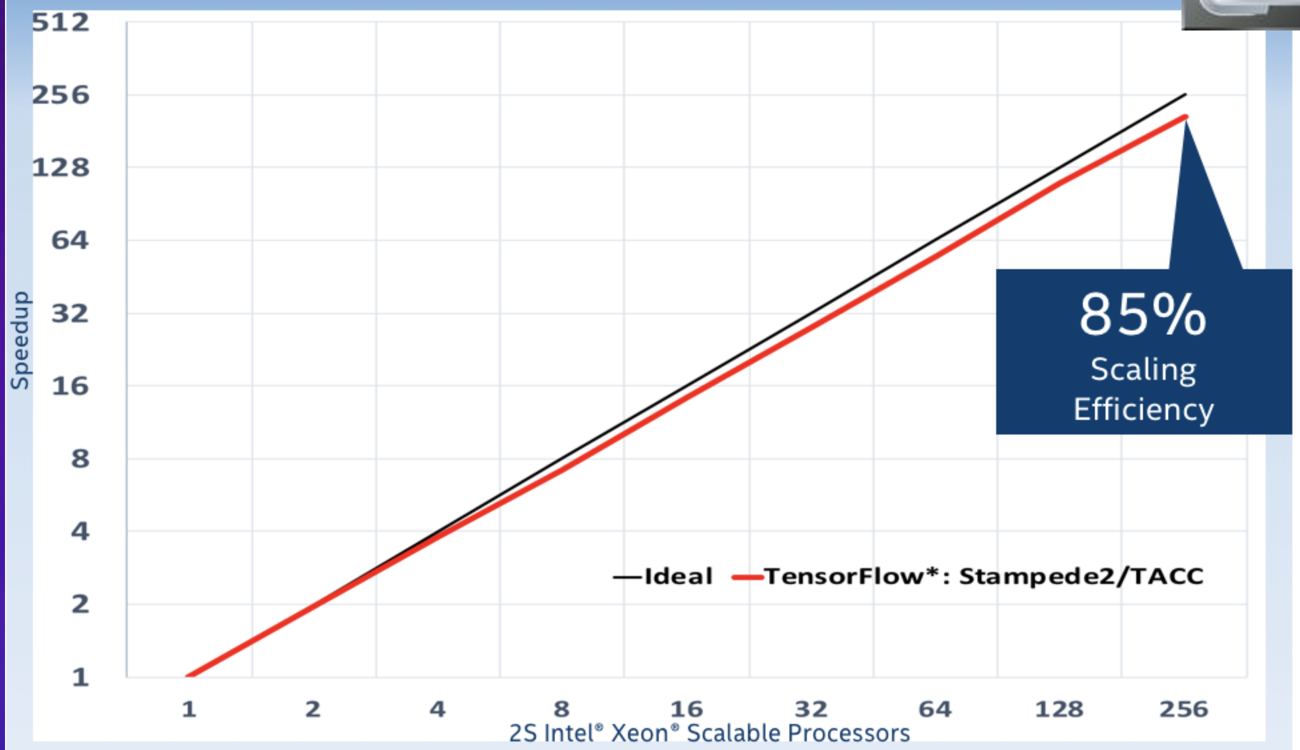
Where hostfile is the file containing the hostnames, one on each new line

EFFICIENT DL SCALING ON EXISTING INFRASTRUCTURE

High scaling efficiencies with 74% Top-1 accuracy on multi node 2S Intel® Xeon® Platinum 8160 Processor Cluster

TENSORFLOW

RESNET-50



ImageNet-1K Global BS=64K

Convergence: 74.5% Top-1/92% Top-5

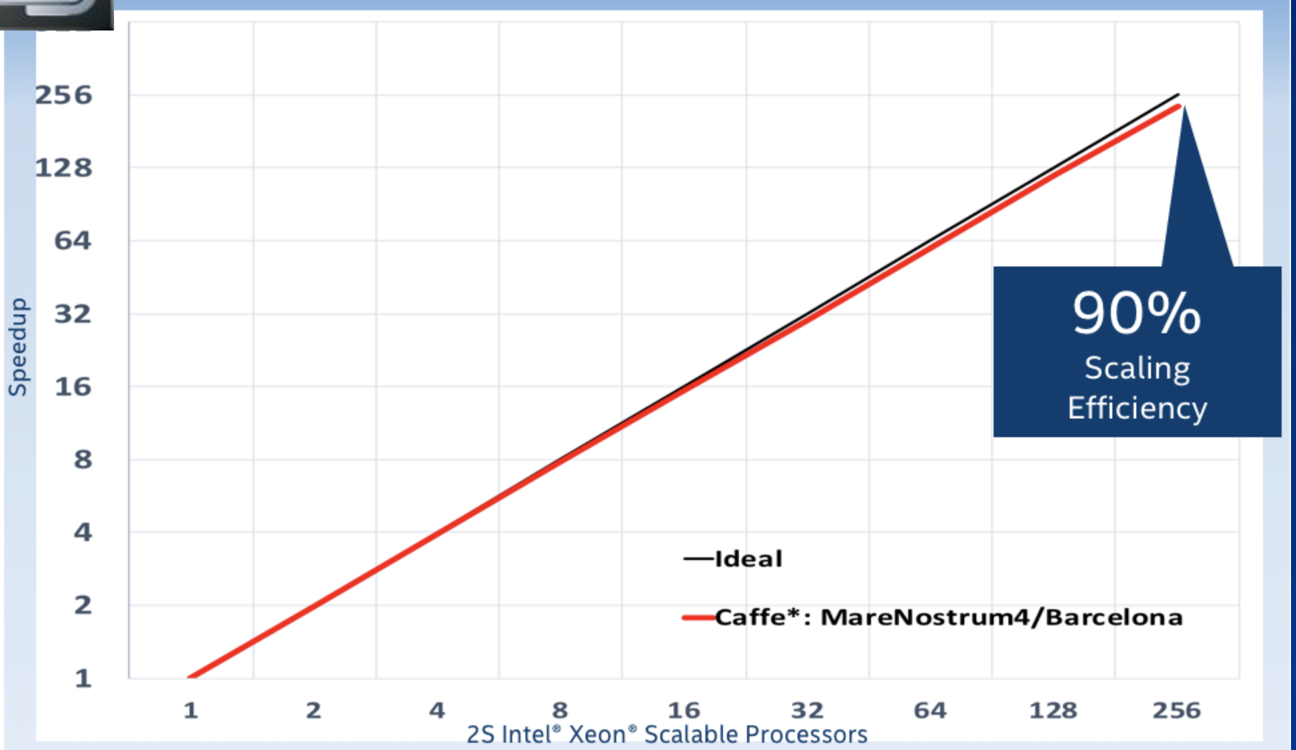
Improved single-node perf with multi-workers/socket

Best Practices From SURFsara B.V.: https://surfdrive.surf.nl/files/index.php/s/xrEFLPvo7IDRARs

Boosting AI Performance: https://software.intel.com/en-us/articles/boosting-deep-learning-training-inference-performance-on-xeon-and-xeon-phi

INTEL CAFFE

RESNET-50



ImageNet-1K Global BS=8192

Convergence: Top-1/Top-5 > 74%/92%

Throughput: 15170 Images/sec

Best Practices From SURFsara B.V: https://surfdrive.surf.nl/files/index.php/s/xrEFLPvo7IDRARs

INNOVATE HARDWARE SOLUTIONS

TRAINING ENHANCEMENTS

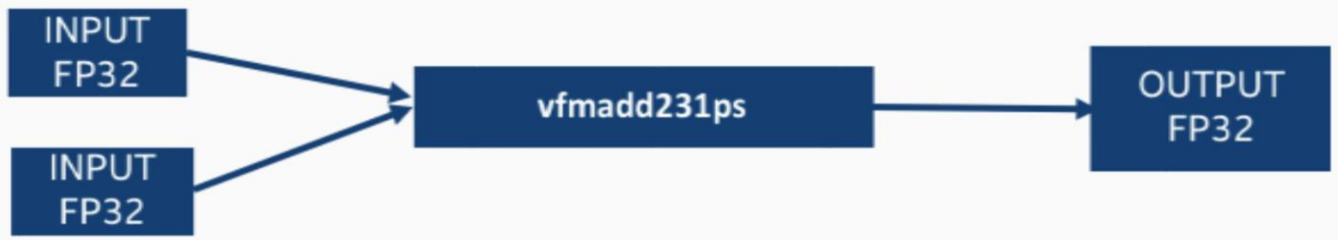
EMBEDDED ACCELERATION WITH AVX-512

AVX-512 Instructions bring embedded acceleration for AI on Intel® Xeon® Scalable processors

FP32

Sign	Exponent								Mantissa																						
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00

Typical AVX-512 instruction to perform FP32 convolutions: **vfmadd231ps**



VECTOR NEURAL NETWORK INSTRUCTIONS

Low Precision Instructions bring embedded acceleration for AI on Intel® Xeon® Scalable

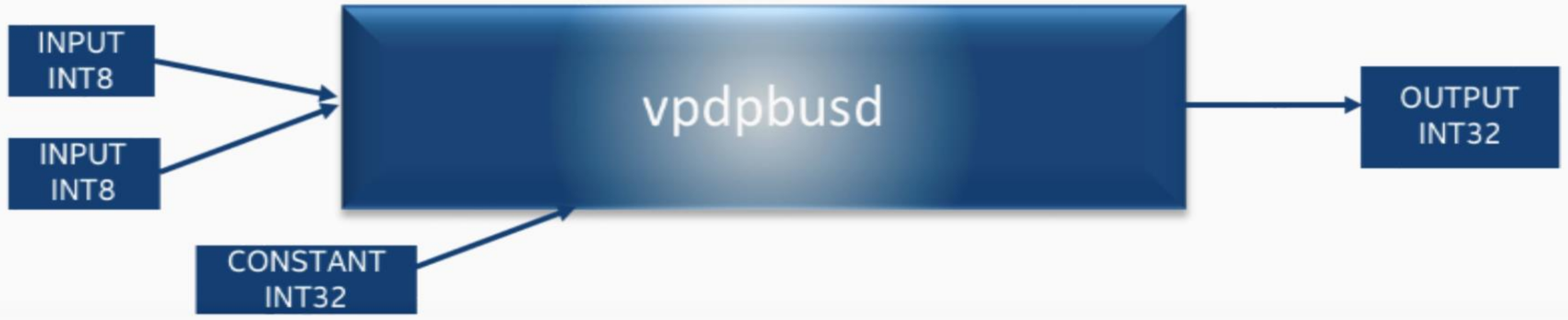
INT8

Sign	Mantissa						
07	06	05	04	03	02	01	00

Current AVX-512 instructions to perform INT8 convolutions: **vpmaddubsw, vpmaddwd, vpaddd**

Future AVX-512 (VNNI) instruction to accelerate INT8 convolutions: **vpdpbusd****

FUTURE VNNI



** ~~vpmaddubsw, vpmaddwd, vpaddd~~ → vpdpbusd

ACCELERATE DEPLOYMENTS

DEPLOYING TENSORFLOW WITH SINGULARITY

- **Install Singularity on the Infrastructure Nodes**
  - https://singularity.lbl.gov/install-linux - as root/sudo
- **Build a TensorFlow Singularity Image: *tf_singularity.simg***
  - Build an image comprising of
    - Linux OS
    - Optimized TensorFlow*
    - Horovod Communication MPI library
    - TensorFlow Application
- **Running Application**
  - *singularity exec tf_singularity.img* \

    python /<path-to-benchmarks>/tf_cnn_benchmarks/tf_cnn_benchmarks.py --model resnet50 \

    --batch_size 64 --data_format NCHW --num_batches 1000 --distortions=True --mkl=True --device cpu \

    --num_intra_threads $OMP_NUM_THREADS --num_inter_threads 2 --kmp_blocktime=0

ECOSYSTEM ENABLEMENT

AI ECOSYSTEM & RESEARCH

ESTABLISH THOUGHT LEADERSHIP, INNOVATE ON INTEL ARCHITECTURE, SYNERGIZE INTEL PRODUCT & RESEARCH

Computational Intelligence

Heterogenous architectures for adaptive and always learning devices with NLP and conversational understanding capabilities and visual applications

Experiential Computing

3D scene understanding using DL based analysis of large video databases, Computer Vision in the cloud – enable effective data mining of large collections of Video

Approximate Computing

Always-on audio-visual multi-modal interaction , Self configuring audio-visual hierarchical sensing through approximate computing data path

Intel Invests **$1+ Billion** in the AI Ecosystem to Fuel Adoption and Product Innovation(1)

100+ University engagements

Deep Learning Architecture

Deep Learning hardware and software advancements, scaling to very large clusters and new applications

Visual Cloud Systems

Large-scale systems research for scaling out visual applications and data, large-scale video analysis

Security in Deep Learning

Built-in safety mechanisms for wide spread mission critical use, ascertaining the confidence and removing anomalous and out of distribution samples in autonomous driving, medicine, security

Brain Science

Create an instrument to connect human behavior to brain function, toolkits for the analysis of brain function , Real-time cloud services for neuroimaging analysis Applying neuroscientific insights to AI

(1) https://newsroom.intel.com/editorials/intel-invests-1-billion-ai-ecosystem-fuel-adoption-product-innovation/

ADVANCING AI PERFORMANCE WITH INTEL® XEON® SCALABLE

TIME TO SOLUTION
FOR PRODUCTION AI

JOURNEY TO
PRODUCTION AI

WORKLOAD FLEXIBILITY &
SCALABILITY

DEEP LEARNING IN
DATA CENTERS

MAXIMIZE PERFORMANCE
USE OPTIMIZED SW

INTEL AI FOCUS
PILLARS

intel® AI

AIDC
INTEL AI DEVCON 2018

ADVANCING AI PERFORMANCE WITH INTEL® XEON® SCALABLE

TIME TO SOLUTION IS KEY FOR PRODUCTION AI

builders.intel.com/ai/solutionslibrary

MAXIMIZE PERFORMANCE WITH OPTIMIZED SW

software.intel.com/en-us/ai-academy

ENGAGE, CONTRIBUTE & DEVELOP AI ON IA

builders.intel.com/ai/devcloud

NOTICES & DISCLAIMERS

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer. No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.

The products described 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.

Performance estimates were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.

AIDC

INTEL AI DEVCON 2018

BACKUP

OPTIMIZED LIBRARIES

Library	Commercial Product Version	Opens source version
Intel® Math Kernel Library for Deep Neural Networks (Intel® MKL-DNN)		MKL-DNN Github
Intel® Math Kernel Library (Intel® MKL)	Intel MKL Product site	
Intel® Data Analytics Acceleration Library (Intel® DAAL)	Intel DAAL Product Site	Intel DAAL Github
Intel® Integrated Performance Primitives (Intel® IPP)	Intel IPP Product Site	

OPTIMIZED FRAMEWORKS

Framework	How to Access Optimized Framework
TensorFlow	Install Intel optimized wheel, see tensorflow.org page for CPU optimization instructions
MXNet	Intel optimizations in main branch via experimental path, available here
Caffe2	Will upstream to master branch in Q2
PaddlePaddle	Paddle Paddle master branch
PyTorch	Intel optimizations available in this branch
Caffe	Intel optimized version of Caffe
CNTK	CNTK master branch

Configuration Details 1

Benchmark Segment	AI/ML
Benchmark type	Training/Inference
Benchmark Metric	Images/Sec or Time to train in seconds
Framework	Caffe
Topology	
# of Nodes	1
Platform	Purley
Sockets	2S
Processor	Xeon Platinum, 205W, 28 core, 2.5 GHz
BIOS	SE5C620.86B.01.00.0470.040720170855
Enabled Cores	28
Platform	
Slots	12
Total Memory	192GB (96 GB per socket)
Memory Configuration	6 slots/16 GB/2666 Mt/s DDR4 RDIMMs
Memory Comments	Micron (18ASF2G72PDZ-2G6D1)
SSD	INTEL SSDSC2KW48 480 GB
OS	RHEL Server Release 7.2 (Maipo), Linux kernel 3.10.0-327.el7.x86_64
OS\Kernel Comments	
Other Configurations	
HT	ON
Turbo	ON
Computer Type	Server
Framework Version	http://github.com/intel/caffe/ c7ed32772affaf1d9951e2a93d986d22a8d14b88 (release_1.0.6)
Topology Version, BATCHSIZE	best (resnet -- 50, gnet_v3 -- 224, ssd -- 224)
Dataset, version	ImageNet / DummyData layer
Performance command	Inference measured with “caffe time --forward_only -phase TEST” command, training measured with “caffe train” command.
Data setup	DummyData layer (generating dummy images on the fly)
Compiler	Intel C++ compiler ver. 17.0.5 20171101
MKL Library version	Intel MKL small libraries version 2018.0.1.20171007
MKL DNN Library Version	-
Performance Measurement Knobs	Environment variables: KMP_AFFINITY='granularity=fine, compact,1,0‘, OMP_NUM_THREADS=28, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance
Memory knobs	Caffe run with “numactl -l“.

Performance estimates were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

Skylake AI Configuration Details as of July 11th, 2017

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.

Broadwell AI Configuration Details as of July 11th, 2017

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 May8th 2018

Benchmark Segment	AI/ML
Benchmark type	Training/Inference
Benchmark Metric	Images/Sec or Time to train in seconds
Framework	Caffe
Topology	
# of Nodes	1
Platform	Purley
Sockets	2S
Processor	Xeon Platinum, 205W, 28 core, 2.5 GHz
BIOS	SE5C620.86B.01.00.0470.040720170855
Enabled Cores	28
Platform	
Slots	12
Total Memory	192GB (96 GB per socket)
Memory Configuration	6 slots/16 GB/2666 Mt/s DDR4 RDIMMs
Memory Comments	Micron (18ASF2G72PDZ-2G6D1)
SSD	INTEL SSDSC2KW48 480 GB
OS	RHEL Server Release 7.2 (Maipo), Linux kernel 3.10.0-327.el7.x86_64
OS\Kernel Comments	
Other Configurations	
HT	ON
Turbo	ON
Computer Type	Server
Topology Version, BATCHSIZE	best (resnet -- 50, gnet_v3 -- 224, ssd -- 224)
Dataset, version	ImageNet / DummyData layer
Performance command	Inference measured with “caffe time --forward_only -phase TEST” command, training measured with “caffe train” command.
Data setup	DummyData layer (generating dummy images on the fly)
Compiler	Intel C++ compiler ver. 17.0.5 20171101
Performance Measurement Knobs	Environment variables: KMP_AFFINITY='granularity=fine, compact,1,0‘, OMP_NUM_THREADS=28, CPU Freq set with cpupower frequency-set -d 2.5G -u 3.8G -g performance
Memory knobs	Caffe run with “numactl -l“.

caffe branch: origin/master version: a3d5b022fe026e9092fc7abc7654b1162ab9940d	**MKLDNN** version: 464c268e544bae26f9b85a2acb9122c766a4c396

Performance estimates were obtained prior to implementation of recent software patches and firmware updates intended to address exploits referred to as "Spectre" and "Meltdown." Implementation of these updates may make these results inapplicable to your device or system.

Config 41

CONFIGURATION DETAILS OF AMAZON EC2 C5.18XLARGE 1 NODE SYSTEMS

Benchmark Segment	AI/ML
Benchmark type	Inference
Benchmark Metric	Sentence/Sec
Framework	Official mxnet
Topology	GNMT(sockeye)
# of Nodes	1
Platform	Amazon EC2 C5.18xlarge instance
Sockets	2S
Processor	Intel® Xeon® Platinum 8124M CPU @ 3.00GHz (Skylake)
BIOS	N/A
Enabled Cores	18 cores / socket
Platform	N/A
Slots	N/A
Total Memory	144GB
Memory Configuration	N/A
SSD	EBS Optimized 200GB, Provisioned IOPS SSD
OS	Red Hat 7.2 (HVM) Amazon Elastic Network Adapter (ENA) Up to 10 Gbps of aggregate network bandwidth
Network Configurations	Installed Enhanced Networking with ENA on Centos Placed the all instances in the same placement
HT	ON
Turbo	ON
Computer Type	Server

Config 34

Configuration details of Amazon EC2 C5.18xlarge 1 node systems

Framework Version	mxnet mkldnn : https://github.com/apache/incubator-mxnet/ 4950f6649e329b23a1efdc40aaa25260d47b4195
Topology Version	GNMT: https://github.com/awslabs/sockeye/tree/master/tutorials/wmt
Batch size	GNMT:1 2 8 16 32 64 128
Dataset, version	GNMT: WMT 2017 (http://data.statmt.org/wmt17/translation-task/preprocessed/)
MKLDNN	F5218ff4fd2d16d13aada2e632afd18f2514fee3
MKL	Version: parallel_studio_xe_2018_update1 http://registrationcenterdownload.intel.com/akdlm/irc_nas/tec/12374/parallel_studio_xe_2018_update1_cluster_edition_online.tgz
Compiler	g++: 4.8.5 gcc: 7.2.1

Dawnbench Configurations

EC2 Instance type	Machine Type	vCPU (#s)	Memory (GiB)	Disk(GB)	Storage (mbps, EBS bandwidth)	Ethernet (Gigabit)	Price ($ per Hour)
C5.2xlarge	SKX 8124M 4 cores 3 GHz base frequency	8	16	128	Up to 2,250	Up to 10	0.34
C5.4xlarge	SKX 8124M 8 cores 3 GHz base frequency	16	32	128	2,250	Up to 10	0.68
C5.18xlarge	SKX 8124M 2S x 18 cores 3 GHz base frequency	72	144	128	9,000	25	3.06

Config 38

Dawnbench IntelCaffe Inference topology

Inference: INT8 Resnet50 with 15% pruning, 53% performance gain over FP32

- Based on 93.3% accuracy FP32 Resnet50 topology
- Pure INT8 except for first convolution layer: <0.2% accuracy drop, 43% performance gain over FP32
- 15% filter pruned according to KL distance: <0.1% accuracy drop, 15% performance gain over FP32

Config 39

Intel® and SURFsara* Research Collaboration MareNostrum4/BSC* Configuration Details

***MareNostrum4/Barcelona Supercomputing Center**: https://www.bsc.es/

Compute Nodes: 2 sockets Intel® Xeon® Platinum 8160 CPU with 24 cores each @ 2.10GHz for a total of 48 cores per node, 2 Threads per core, L1d 32K; L1i cache 32K; L2 cache 1024K; L3 cache 33792K, 96 GB of DDR4, Intel® Omni-Path Host Fabric Interface, dual-rail. Software: Intel® MPI Library 2017 Update 4Intel® MPI Library 2019 Technical Preview OFI 1.5.0PSM2 w/ Multi-EP, 10 Gbit Ethernet, 200 GB local SSD, Red Hat* Enterprise Linux 6.7.

Intel® Caffe: Intel® version of Caffe; http://github.com/intel/caffe/, revision 8012927bf2bf70231cbc7ff55de0b1bc11de4a69.
Intel® MKL version: mklml_lnx_2018.0.20170425; Intel® MLSL version: l_mlsl_2017.1.016

Model: Topology specs from https://github.com/intel/caffe/tree/master/models/intel_optimized_models (ResNet-50) and modified for wide-RedNet-50. Batch size as stated in the performance chart

Time-To-Train: measured using "train" command. Data copied to memory on all nodes in the cluster before training. No input image data transferred over the fabric while training;

Performance measured with:
export OMP_NUM_THREADS=44 (the remaining 4 cores are used for driving communication), export I_MPI_FABRICS=tmi, export I_MPI_TMI_PROVIDER=psm2

OMP_NUM_THREADS=44 KMP_AFFINITY="proclist=[0-87],granularity=thread,explicit" KMP_HW_SUBSET=1t MLSL_NUM_SERVERS=4 mpiexec.hydra -PSM2 -l -n $SLURM_JOB_NUM_NODES -ppn 1 -f hosts2 -genv OMP_NUM_THREADS 44 -env KMP_AFFINITY "proclist=[0-87],granularity=thread,explicit" -env KMP_HW_SUBSET 1t -genv I_MPI_FABRICS tmi -genv I_MPI_HYDRA_BRANCH_COUNT $SLURM_JOB_NUM_NODES -genv I_MPI_HYDRA_PMI_CONNECT alltoall sh -c 'cat /ilsvrc12_train_lmdb_striped_64/data.mdb > /dev/null ; cat /ilsvrc12_val_lmdb_striped_64/data.mdb > /dev/null ; ulimit -u 8192 ; ulimit -a ; numactl -H ; /caffe/build/tools/caffe train --solver=/caffe/models/intel_optimized_models/multinode/resnet_50_256_nodes_8k_batch/solver_poly_quick_large.prototxt -engine "MKL2017"

SURFsara blog: https://blog.surf.nl/en/imagenet-1k-training-on-intel-xeon-phi-in-less-than-40-minutes/ ; Researchers: Valeriu Codreanu, Ph.D. (PI).; Damian Podareanu, MSc; SURFsara* & Vikram Saletore, Ph.D. (co-PI): Intel Corp.

*SURFsara B.V. is the Dutch national high-performance computing and e-Science support center. Amsterdam Science Park, Amsterdam, The Netherlands.

Stampede2*/TACC* Configuration Details

***Stampede2/TACC**: https://portal.tacc.utexas.edu/user-guides/stampede2

Compute Nodes: 2 sockets Intel® Xeon® Platinum 8160 CPU with 24 cores each @ 2.10GHz for a total of 48 cores per node, 2 Threads per core, L1d 32K; L1i cache 32K; L2 cache 1024K; L3 cache 33792K, 96 GB of DDR4, Intel® Omni-Path Host Fabric Interface, dual-rail. Software: Intel® MPI Library 2017 Update 4Intel® MPI Library 2019 Technical Preview OFI 1.5.0PSM2 w/ Multi-EP, 10 Gbit Ethernet, 200 GB local SSD, Red Hat* Enterprise Linux 6.7.

TensorFlow*: http://github.com/intel/caffe/, revision 8012927bf2bf70231cbc7ff55de0b1bc11de4a69.
Intel® MKL version: mklml_lnx_2018.0.20170425; Intel® MLSL version: l_mlsl_2017.1.016

Model: Topology specs from https://github.com/intel/caffe/tree/master/models/intel_optimized_models (ResNet-50) and modified for wide-RedNet-50.; Batch size as stated in the performance chart

Performance measured with:
export OMP_NUM_THREADS=10 Per Worker (the remaining 2 cores are used for driving communication), export I_MPI_FABRICS=tmi, export I_MPI_TMI_PROVIDER=psm2

OMP_NUM_THREADS=10 KMP_AFFINITY="proclist=[0-63],granularity=thread,explicit" KMP_HW_SUBSET=1t MLSL_NUM_SERVERS=4 mpiexec.hydra -PSM2 -l -n $SLURM_JOB_NUM_NODES -ppn 1 -f hosts2 -genv OMP_NUM_THREADS 64 -env KMP_AFFINITY "proclist=[0-63],granularity=thread,explicit" -env KMP_HW_SUBSET 1t -genv I_MPI_FABRICS tmi -genv I_MPI_HYDRA_BRANCH_COUNT $SLURM_JOB_NUM_NODES -genv I_MPI_HYDRA_PMI_CONNECT alltoall sh -c 'cat /ilsvrc12_train_lmdb_striped_64/data.mdb > /dev/null ; cat /ilsvrc12_val_lmdb_striped_64/data.mdb > /dev/null ; ulimit -u 8192 ; ulimit -a ; numactl -H ; /caffe/build/tools/caffe train --solver=/caffe/models/intel_optimized_models/multinode/resnet_50_256_nodes_8k_batch/solver_poly_quick_large.prototxt -engine "MKL2017"

BACKUP

Case Study: Time-series Pattern Detection
Leading U.S. Market Exchange

RESULT

10X REDUCTION

In data storage and search complexity costs, with more accurate matches than non-deep learning approach

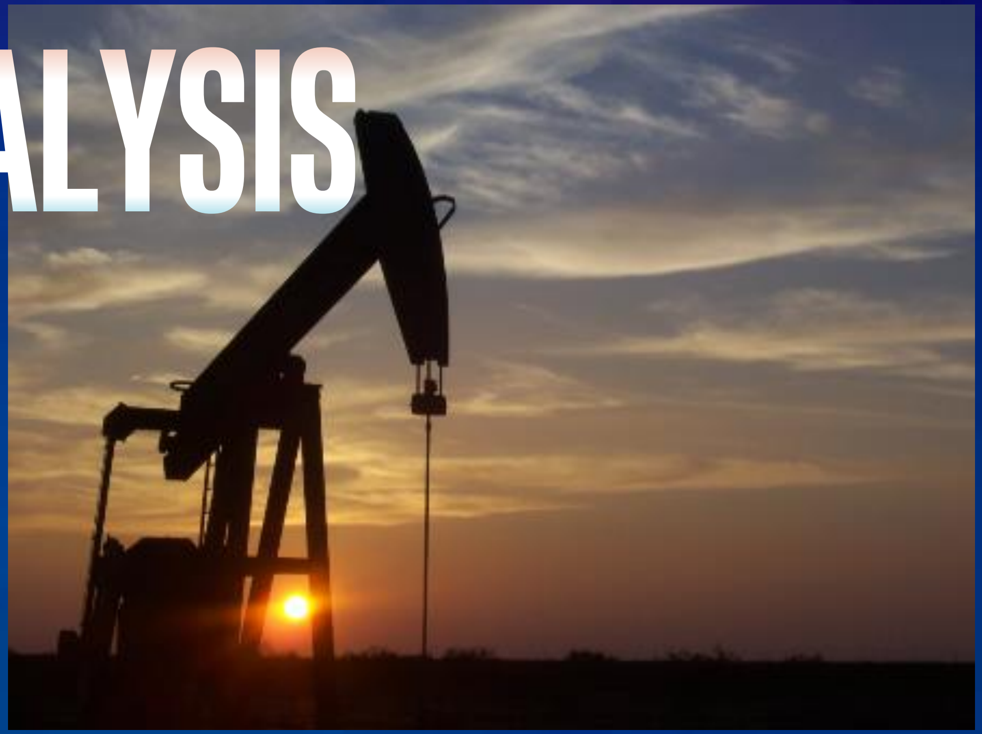
Client: Leading U.S. market exchange

Challenge: Identify known patterns or anomalies in market trading data in order to predict investment activity, such as fraudulent activity or spikes in trading volume and whether any action is required.

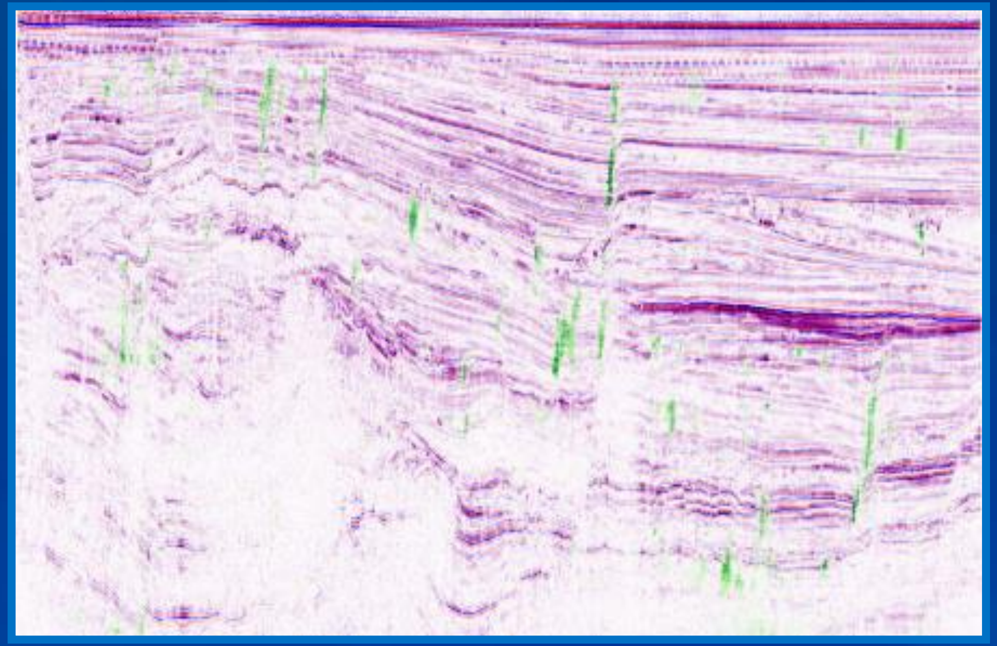
Solution: Using Intel® Nervana™ Cloud and Neon framework. Built a recurrent neural network (RNN)-based model, utilizing encoders and decoders, in order to ingest public order book data and automatically learn informative patterns of activity in the historical data. Time series analysis enables new use cases for fraud detection, anomaly detection, and other future applications

PROOF OF CONCEPT: IMAGE RECOGNITION

SEISMIC REFLECTION ANALYSIS



- Client:
  - A leading developer of software solutions to the global oil and gas industry.
- Challenge:
  - Automate identification of fault lines within seismic reflection data.
- Solution:
  - Built a proof of concept that is trained using seismic reflection data and can predict the probability of finding fault lines on previously unseen images.
  - Performs pixel-wise semantic segmentation of SEG-Y formatted data
  - Model trained using supervised learning
- Advantages:
  - Automation enables analysis of vast amounts of data faster
  - Could identify potentially rewarding locations from subtle clues in the data



CASE STUDY: ENTERPRISE ANALYTICS

SERPRO*

RESULT

$1 BILLION

Streamlined collection of US $1 billion in revenue by designing new APIs for car and license plate image recognition.

Client: SERPRO, Brazil's largest government-owned IT services corporation, providing technology services to multiple public sector agencies.

Challenge: Across Brazil, 35,000 traffic enforcement cameras document 45 million violations every year, generating US $1 billion in revenue. Fully automating the complex, labor-intensive process for issuing tickets by integrating image recognition via AI could reduce costs and processing time.

Solution: Used deep learning techniques to optimize SERPRO's code. With Brazilian student-partners, developed new algorithms, training and inference tests using Google TensorFlow* on Dell EMC PowerEdge R740*, running on Intel® Xeon® Scalable processor-based platforms.

AIDC
INTEL AI DEVCON 2018