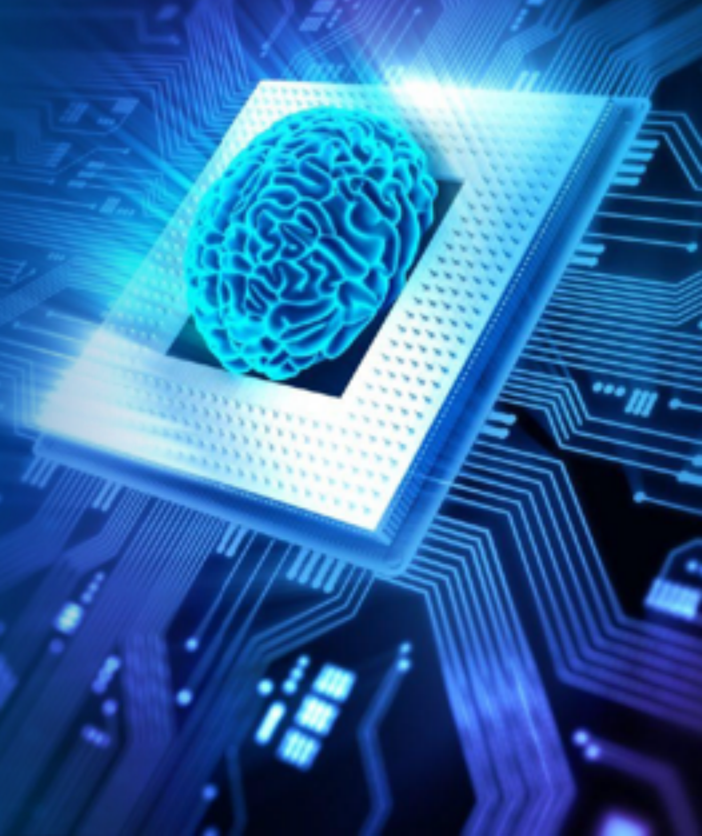
HPC FOR AI TRAINING & INFERENCE



October 9, 2020

G Anthony Reina, M.D.
Chief AI Architect for Health & Life Sciences, Intel

Ravi Panchumarthy, Ph.D
Machine Learning Engineer, Intel.

NOTICES AND DISCLAIMERS

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration.

No product or component 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. For more complete information about performance and benchmark results, visit http://www.intel.com/benchmarks .

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/benchmarks .

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'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.

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.

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.

WORKSHOP 1: MACHINE LEARNING MODULE

9:00 - 10:30

- Deep Learning 101 – Introduction to Convolutional Neural Networks with TensorFlow
- Intel's Hardware and Software directions for Artificial Intelligence (AI),
  Machine Learning (ML), and Deep Learning (DL)
- Hardware Accelerated Deep Learning instructions and implementations
  DL Boost, VNNI instructions

10:30 - 11:00 Coffee break

11:00 - 12:30 Hands On Session

- Performance optimized Python
  - Hands-on Labs with Python focus on Classical Machine Learning examples and algorithms
  - Distributed Machine Learning with Daal4py

ANALYTICS & AI EVERYWHERE

Part of every top 10 strategic technology trend for 2020

MANY APPROACHES TO ANALYTICS & AI

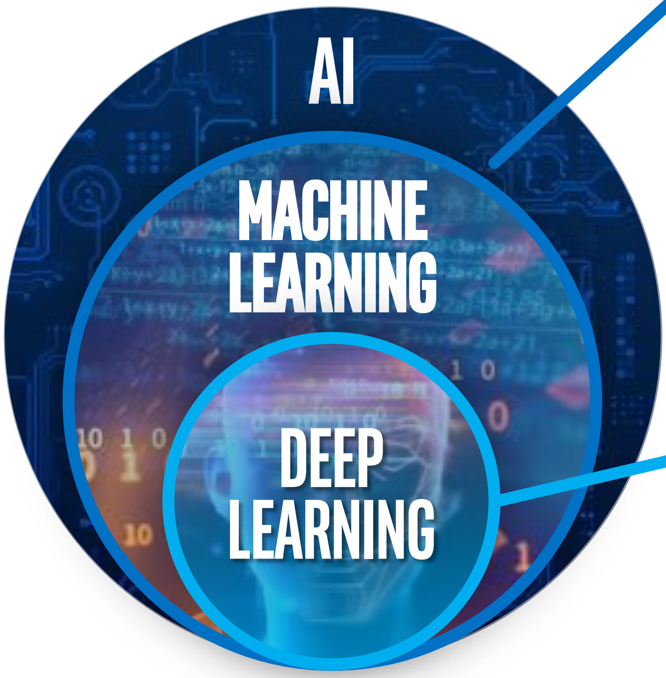
NO ONE SIZE FITS ALL

SUPERVISED LEARNING

UNSUPERVISED LEARNING

SEMI-SUPERVISED LEARNING

REINFORCEMENT LEARNING



Regression (Linear/Logistic)

Classification (Support Vector Machines/SVM, Naïve Bayes)

Clustering (Hierarchical, Bayesian, K-Means, DBSCAN)

Decision Trees (RandomForest)

Extrapolation (Hidden Markov Models/HMM)

More…

Image Recognition (Convolutional Neural Networks/CNN, Single-Shot Detector/SSD)

Speech Recognition (Recurrent Neural Network/RNN)

Natural Language Processing (Long-Short Term Memory/LSTM)

Data Generation (Generative Adversarial Networks/GAN)

Recommender System (Multi-Layer Perceptron/MLP)

Time-Series Analysis (LSTM, RNN)

Reinforcement Learning (CNN, RNN)

More…

MACHINE VS. DEEP LEARNING

MACHINE LEARNING

How do you engineer the best features?

$N \times N$

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

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

CLASSIFIER ALGORITHM

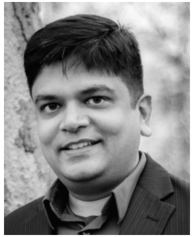
SVM
Random Forest
Naïve Bayes
Decision Trees
Logistic Regression
Ensemble methods

Prashant

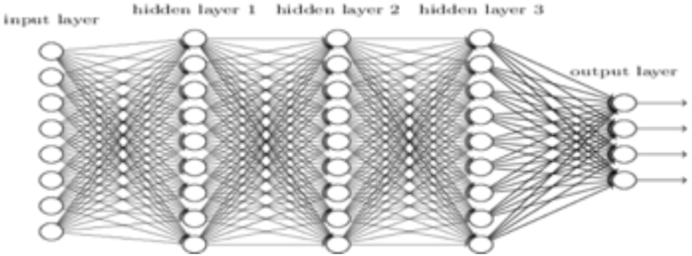
DEEP LEARNING

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

$N \times N$



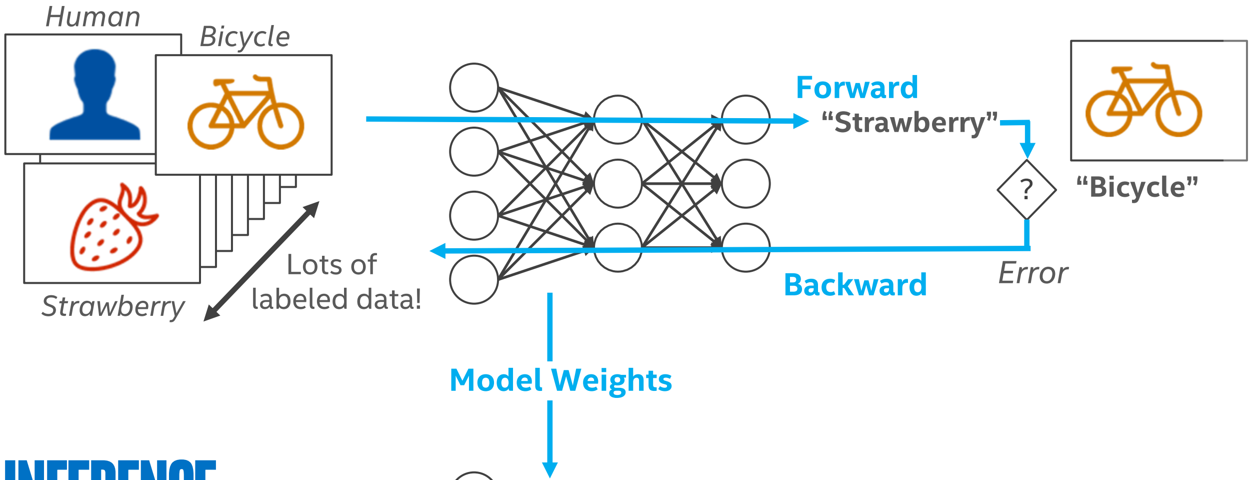
NEURAL NETWORK



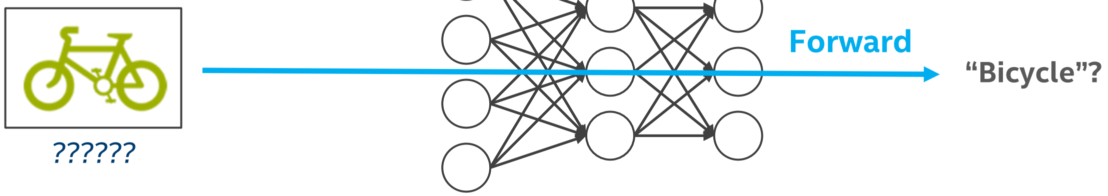
Prashant

DEEP LEARNING BASICS

TRAINING

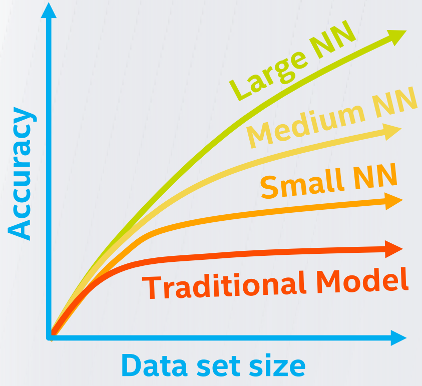


INFERENCE



DID YOU KNOW?

Training with a large data set AND deep (many layered) neural network often leads to the highest accuracy inference



DEEP LEARNING GLOSSARY

LIBRARY

Intel MKL-DNN
Intel DAAL
Spark MlLib
Scikit-Learn
Intel Distribution for Python
Mahout
NumPy
Pandas

Hardware-optimized mathematical and other primitive functions that are commonly used in machine and deep learning algorithms, topologies and frameworks

FRAMEWORK

TensorFlow
BigDL
Spark
mxnet
PaddlePaddle
PyTorch
Caffe

Open-source software environments that facilitate deep learning model development and deployment through built-in components and the ability to customize code

TOPOLOGY

Wide variety of algorithms modeled loosely after the human brain that use neural networks to recognize complex patterns in data that are otherwise difficult to reverse engineer

TRANSLATING COMMON DEEP LEARNING TERMINOLOGY

WHAT IS DEEP LEARNING?

Deep Learning: A subset of Machine Learning focused on Deep Neural Networks using non-linearity

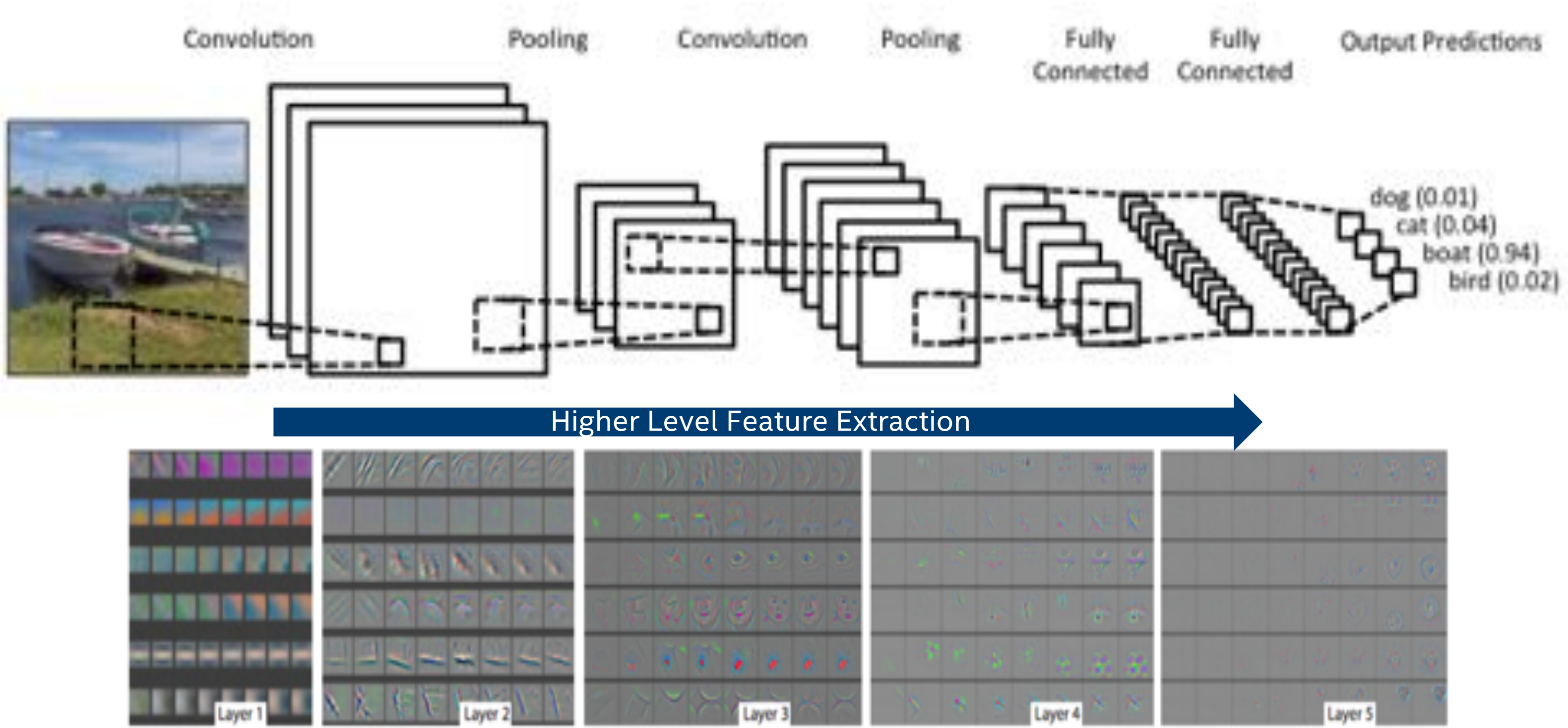
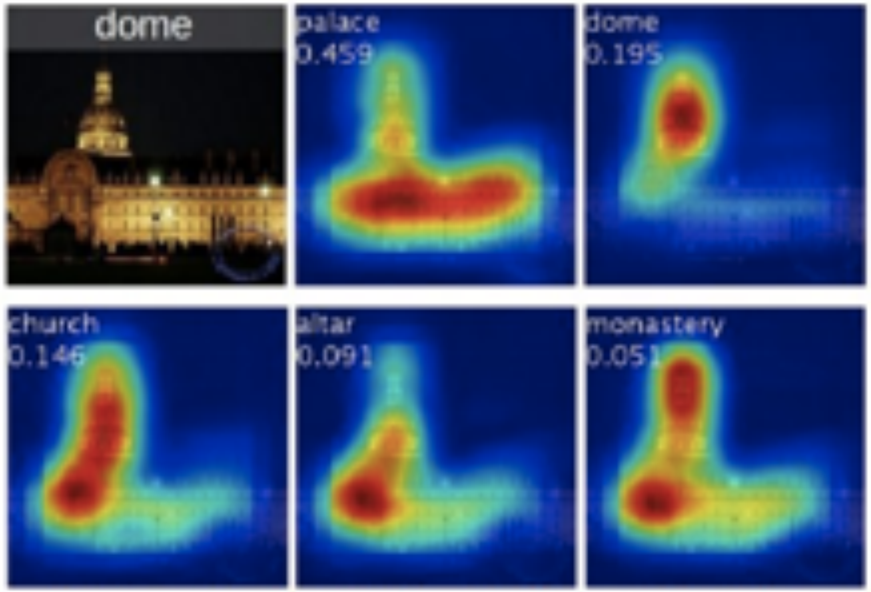
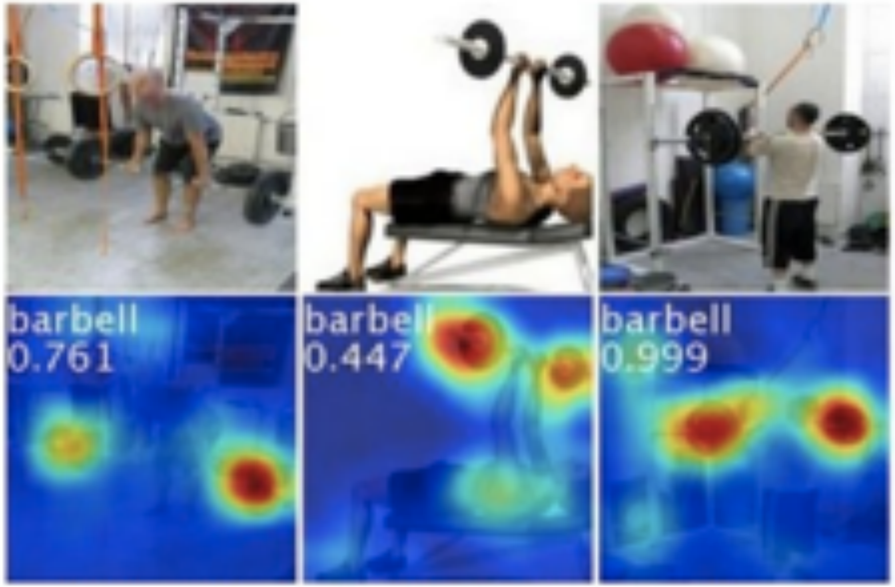


Image Credit: [Visualizing and Understanding Convolutional Networks](#) - Zeiler and Fergus

ACTIVATION MAPS OF CNNS



Class activation maps of top 5 predictions

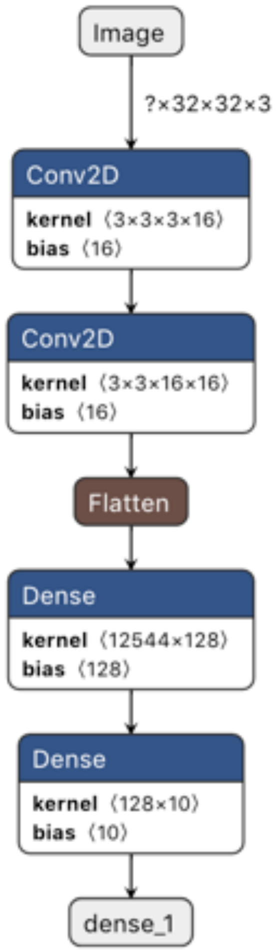


Class activation maps for one object class

CONVOLUTIONAL NEURAL NETWORKS (CNN)

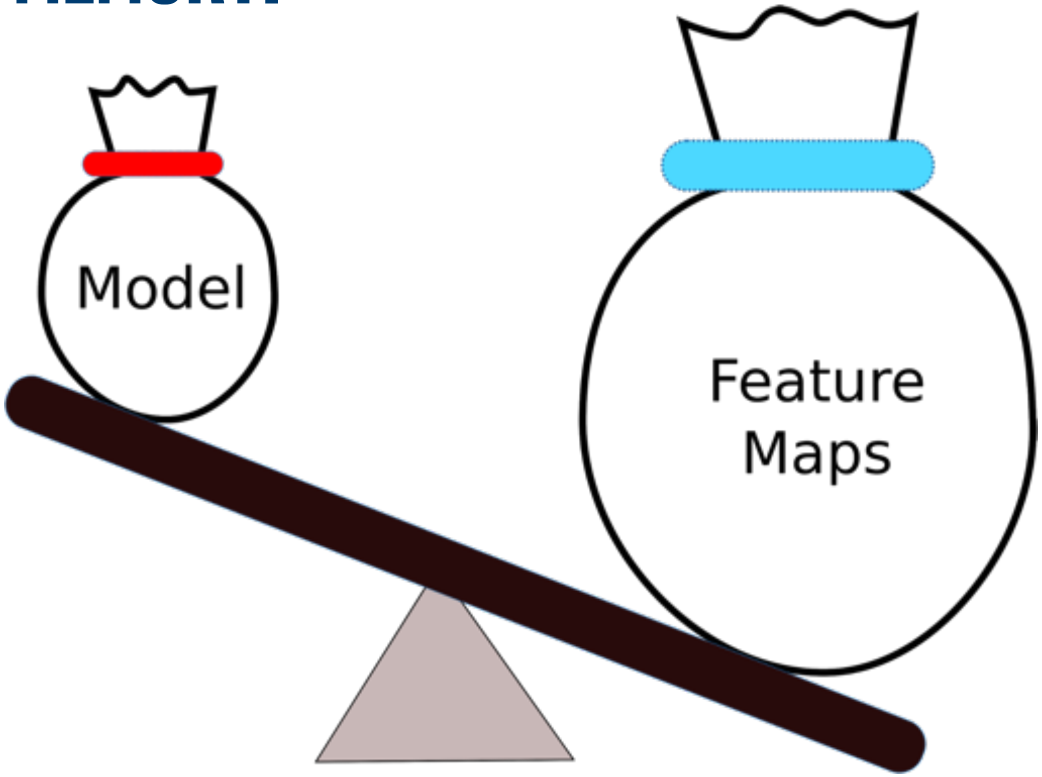
```python
from tensorflow import keras as K

inputs = K.layers.Input((32, 32, 3), name="Image")

cnn_layer1 = K.layers.Conv2D(filters=16,
                             kernel_size=(3,3),
                             activation="relu")(inputs)

cnn_layer2 = K.layers.Conv2D(filters=16,
                             kernel_size=(3,3),
                             activation="relu")(cnn_layer1)

flatten = K.layers.Flatten()(cnn_layer2)

dense1 = K.layers.Dense(units=128, activation="relu")(flatten)

prediction = K.layers.Dense(units=10, activation="softmax")(dense1)

model = K.models.Model(inputs=[inputs], outputs=[prediction])

model.compile(optimizer="adam", loss="binary_crossentropy")
```

Trainable parameters
1,609,818



WHY CPUS? ONE WORD: MEMORY.

For a 384 x 384 x 128 image the combined size of the activation maps is **over 800 times larger** than the size of the 3D U-Net model.



DEEP LEARNING USAGES AND KEY TOPOLOGIES

Image Recognition

Resnet-50
Inception V3
MobileNet
SqueezeNet

Object Detection

R-FCN
Faster-RCNN
Yolo V2
SSD-VGG16, SSD-MobileNet

Image Segmentation

Mask R-CNN

Language Translation

GNMT

Text to Speech

Wavenet

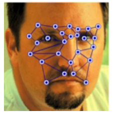
Understand Legalese

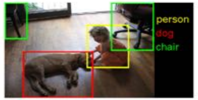
Recommendation System

Wide and Deep, NCF

THERE ARE MANY DEEP LEARNING USAGES AND TOPOLOGIES FOR EACH

DEEP LEARNING | DISRUPTING AT SUPER-HUMAN LEVELS





Allowing Computers to See

Image Classification, Object Detection, Semantic Segment, etc..

Providing Accurate Recommendations

Recommendation Engines, Collaborative Filtering, Missing Interactions

Detecting Threats and Fraud in Systems

Clustering, Outlier detection..

Interacting Naturally with Humans

Forecasting/prediction based on Sequences

Event Prediction

Temporal Data Mining

DRIVERLESS CARS

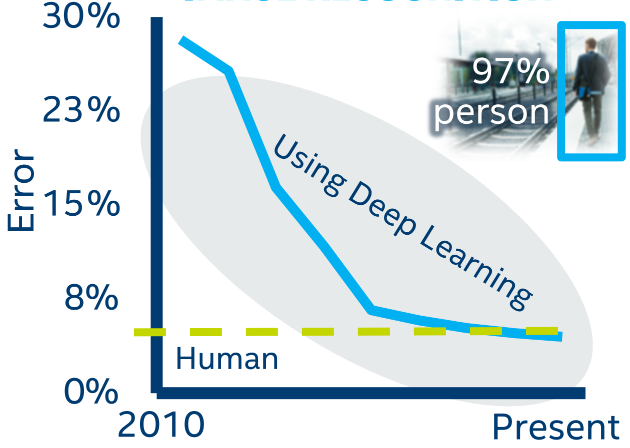
Making Decisions

Agents acting within Environments

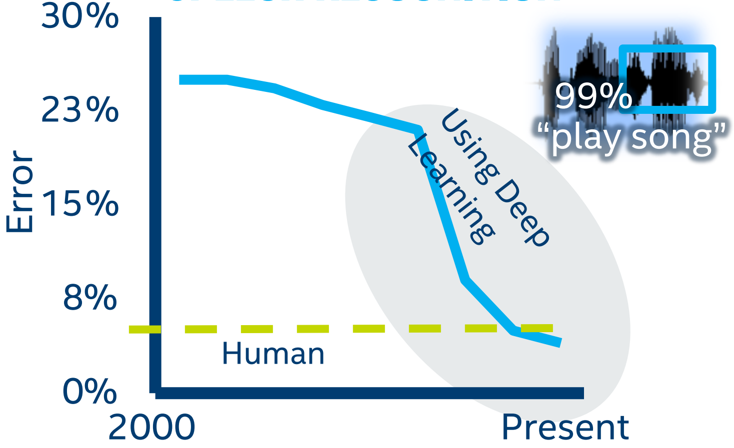
DEEP LEARNING BREAKTHROUGHS

MACHINES ABLE TO MEET OR EXCEED HUMAN IMAGE AND SPEECH RECOGNITION

IMAGE RECOGNITION



SPEECH RECOGNITION



e.g.,

Source: ILSVRC ImageNet winning entry classification error rate each year 2010-2016 (Left), https://www.microsoft.com/en-us/research/blog/microsoft-researchers-achieve-new-conversational-speech-recognition-milestone/ (Right)

AI IS INTERDISCIPLINARY

DATA

Create

Transmit

Ingest

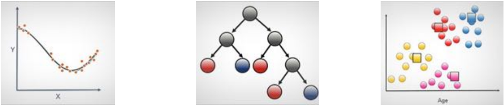
Integrate

Stage

Clean

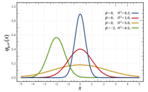
Normalize

MACHINE LEARNING



Regression Classification Clustering

DATA ANALYTICS



Search/Query, Statistics, etc.

ENSEMBLE METHODS

MORE AI

Reinforcement Learning

Symbolic Reasoning

Analogical Reasoning

Evolutionary Computing

Bayes Methods

And More…

DEEP LEARNING

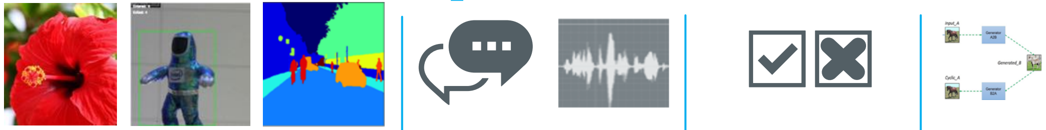


Image: Image Recognition · Object Detection · Image Segmentation

Language: Natural Language Processing (NLP) · Speech ⇄ Text

Recommender: Recommender Systems

More: Data Generation

INSIGHTS

Business

Operational

Security

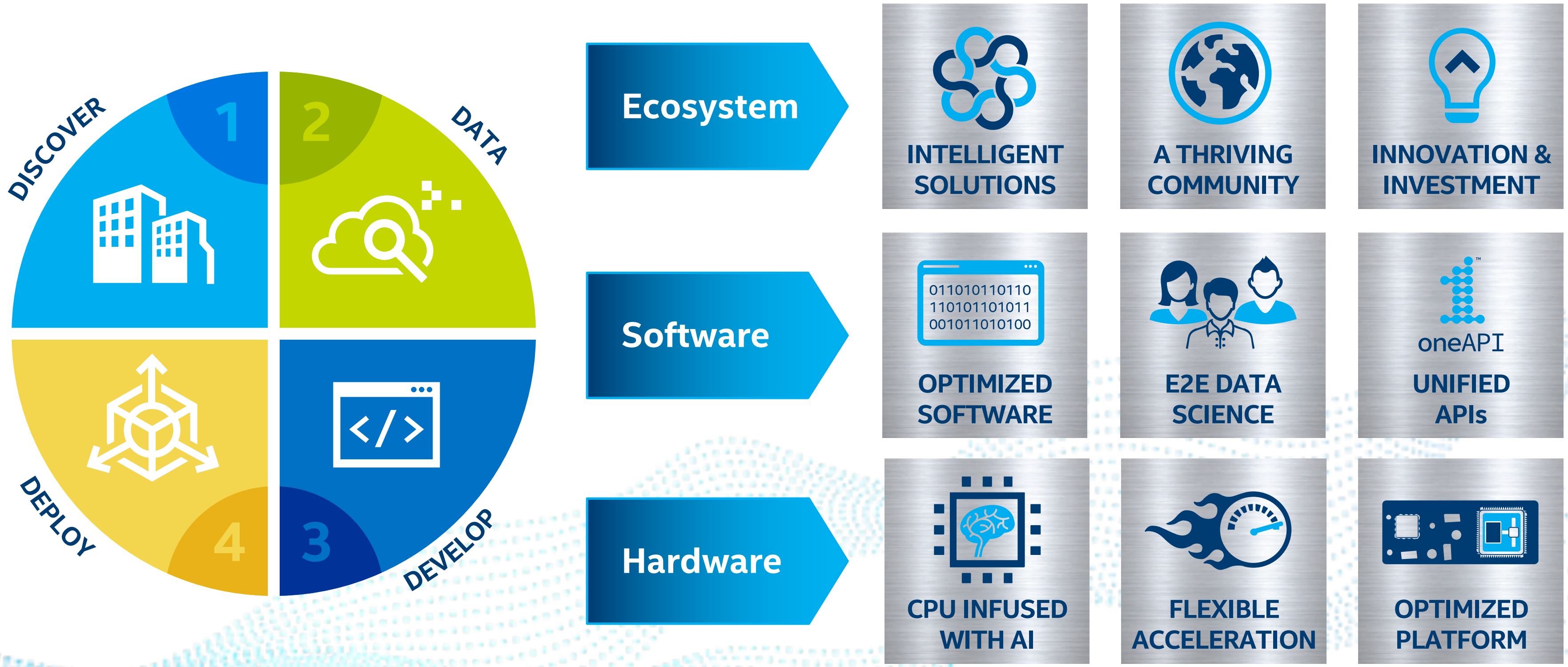
WHICH APPROACH IS BEST?

CHOOSE THE RIGHT TOOL FOR THE JOB

SMART FACTORY

How many parts should we manufacture?	**Analytics** to understand historical supply & demand
What will our production yield be?	**Machine Learning** to identify variables related to yield
Which parts have visual defects?	**Deep Learning** to identify defects in images
Can my robotic arm learn to get better?	**Deep Learning** to learn & adapt to feedback

ACCELERATE YOUR AI JOURNEY WITH INTEL



TOMORROW'S AI

INNOVATION & INVESTMENT

Intel Labs

Innovating Beyond Today's AI

Cognitive
- Knowledgeable AI
- Knowledge Mgmt, (VDMS)
- Robots that Learn

Autonomous
- Drone Acrobatics
- Robotic Surgery
- Path Planner Chip

Efficient
- Neuromorphic (Loihi/Pohoiki)
- Brain-Inspired Compute

Intuitive
- Kids Space / Immersive
- Probabilistic – Human to Robot Interaction
- Healthcare Robotics

Trustworthy
- Autonomous Vehicle Safety (RSS)
- Federated Learning
- Attack mitigation

Intel Capital

Investing in Disruptive AI Innovation

Acquisitions

Movidius
Mobileye
habana
Altera

Investments

Mighty Ai
AEYE
Matroid
CognitiveScale
Element AI
helpshift
DataRobot
& More

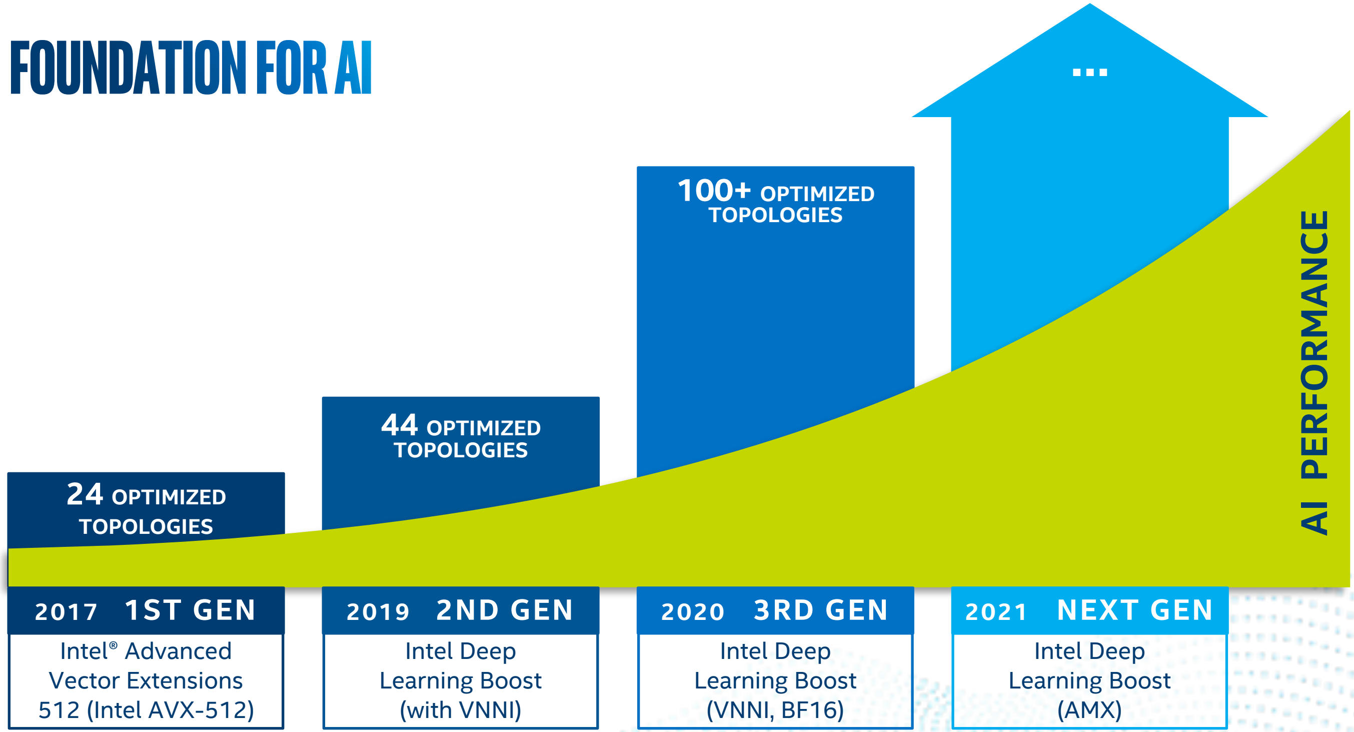
INTEL-OPTIMIZED SOFTWARE for AI Acceleration

OPTIMIZED SOFTWARE

	For Data Scientist	For Developer	Machine Learning	Deep Learning	
Intel® Distribution for python	✓		✓		Accelerate data analytics and machine learning using NumPy, SciPy, scikit-learn & more **software.intel.com/distribution-for-python**
ANALYTICS ZOO	✓		✓	✓	Seamlessly scale AI models on Spark/Hadoop big data clusters for distributed training and inference **software.intel.com/ai/analytics-zoo**
TensorFlow PYTORCH	✓		✓	✓	Develop machine and deep learning models using Intel-optimized popular open-source frameworks **software.intel.com/oneapi/ai-kit**
Model Zoo	✓			✓	Access a repository of deep learning models, scripts, tutorials & more for Intel® Xeon® Scalable processors **github.com/IntelAI/models**
OpenVINO™	✓	✓		✓	Deploy optimized deep learning inference on the Intel hardware that meets your application's unique needs **software.intel.com/openvino-toolkit**

FOUNDATION FOR AI

CPU INFUSED WITH AI



More built-in
AI acceleration &
optimized topologies
with each new gen

oneAPI | ONNX RUNTIME | OpenVINO™ | TensorFlow | PyTorch

OPTIMIZED LIBRARIES AND FRAMEWORKS

INTEL® DEEP LEARNING BOOST (DL BOOST)

featuring Vector Neural Network Instructions (VNNI)

	Sign	Mantissa						
INT8	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****

1. Fused multiply-add instruction
2. MKLDNN is optimized for VNNI

Speeds-up image classification, speech recognition, language translation, object detection and more

DEEP LEARNING AT SCALE


Applied Machine Learning at Facebook:
A Datacenter Infrastructure Perspective

Services	Ranking Algorithm	Photo Tagging	Photo Text Generation	Search	Language Translation	Spam Flagging	Speech
Model(s)	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

LARGE CLOUD USERS EMPLOY CPU EXTENSIVELY FOR DEEP LEARNING

"Inference is one thing we do, but we do lots more. That's why flexibility is really essential."

Kim Hazelwood
Head of AI Infrastructure Foundation
Facebook

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

FOUNDATION FOR ANALYTICS AND AI

THE ONLY DATACENTER CPU WITH INTEGRATED AI ACCELERATION

INTEL® ADVANCED VECTOR EXTENSIONS 512

INTEL® DEEP LEARNING BOOST (INTEL® DL BOOST)

SOFTWARE OPTIMIZATIONS FOR DL FRAMEWORKS

INTEL® OPTANE™ TECHNOLOGY

AVAILABLE TODAY

2ND GENERATION*

PURLEY PLATFORM

DL BOOST (VNNI)

→ *NOW WITH ENHANCED REFRESH SKU'S (-R)!*

3RD GENERATION**

CEDAR ISLAND PLATFORM (4/8S)

NEW EXTENDED DL BOOST (VNNI, BFLOAT16)

FUTURE

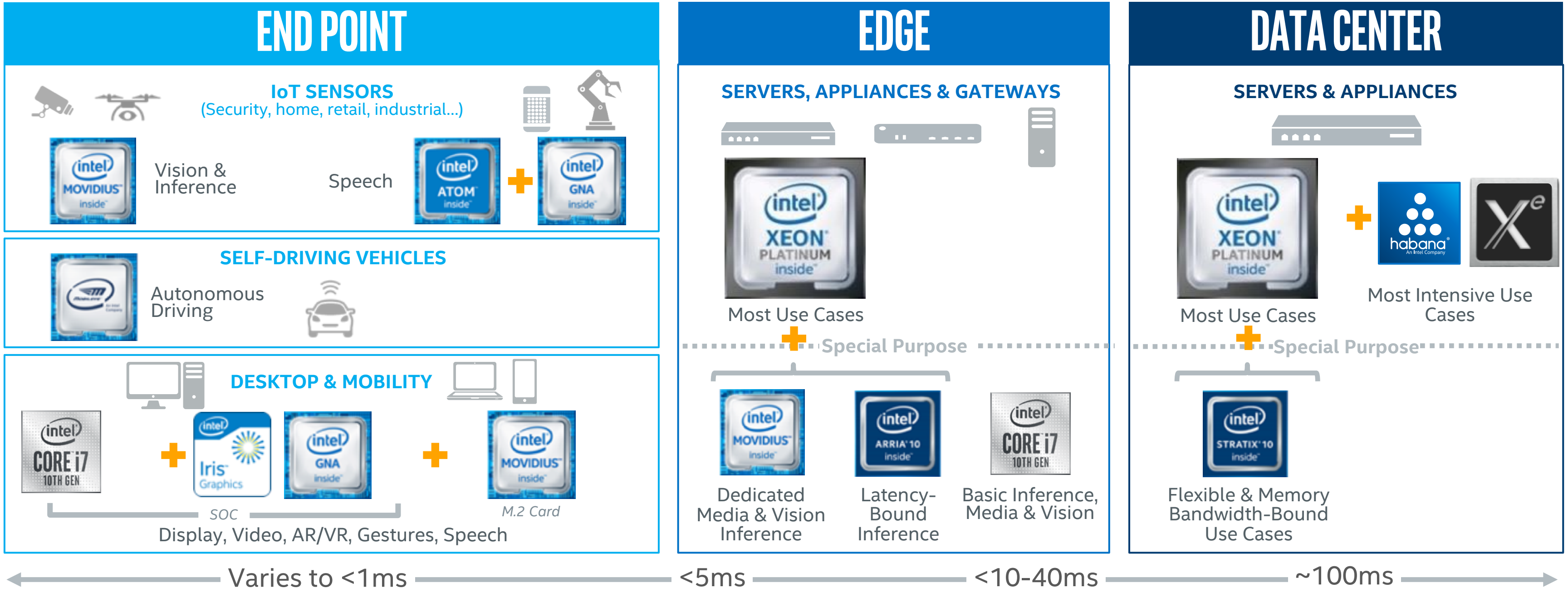
ICE LAKE, SAPPHIRE RAPIDS

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.
*Formerly codenamed "Cascade Lake" ** Formerly codenamed "Cooper Lake"

intel

INTEL HARDWARE

MULTI-PURPOSE TO PURPOSE-BUILT AI COMPUTE FROM DEVICE TO CLOUD



ONE SIZE DOES NOT FIT ALL

GNA=Gaussian Neural Accelerator

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice. Images are examples of intended applications but not an exhaustive list.

AI WILL INFUSE EVERYTHING… …SO WE PUT IT EVERYWHERE

DELIVERING AI FROM CLOUD-TO-DEVICE

FLEXIBLE ACCELERATION

CPU only

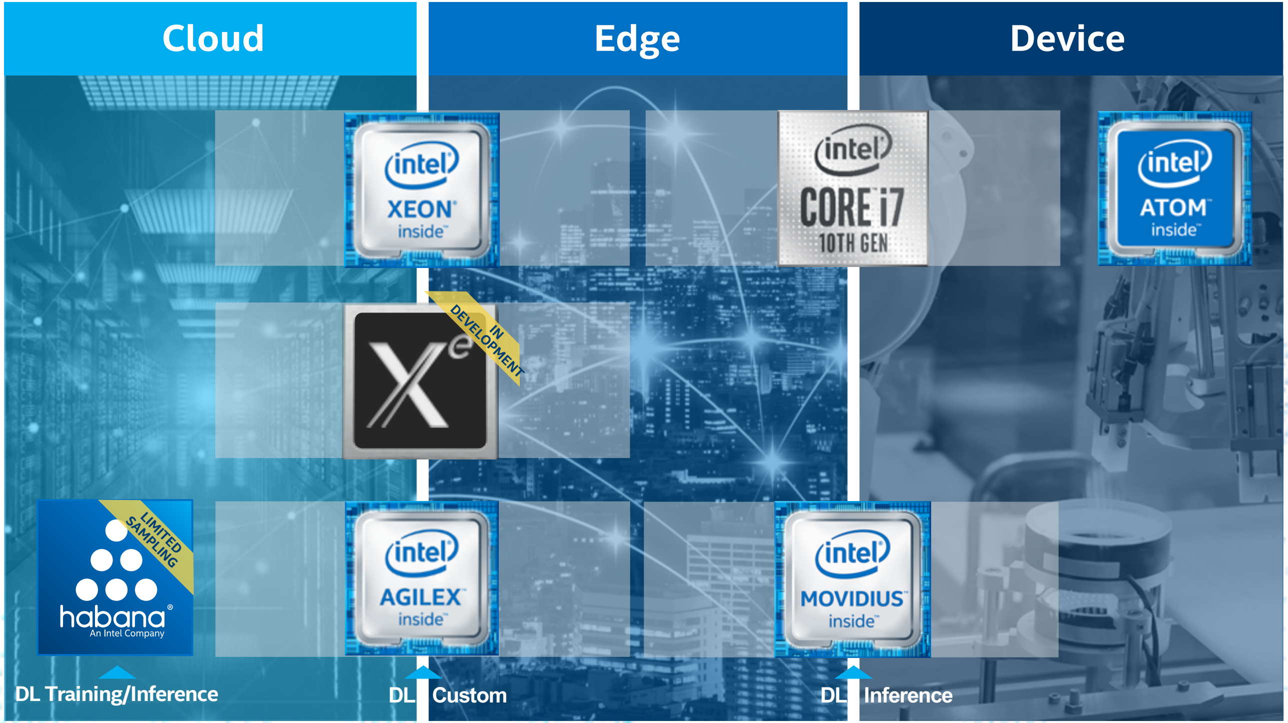
For broad market when AI is a portion of 1,000+ workloads

CPU + GPU

When compute is dominated by AI, HPC, graphics, and real-time media

CPU + XPU

When compute is dominated by deep learning (DL)



PROGRAMMING CHALLENGES
FOR MULTIPLE ARCHITECTURES

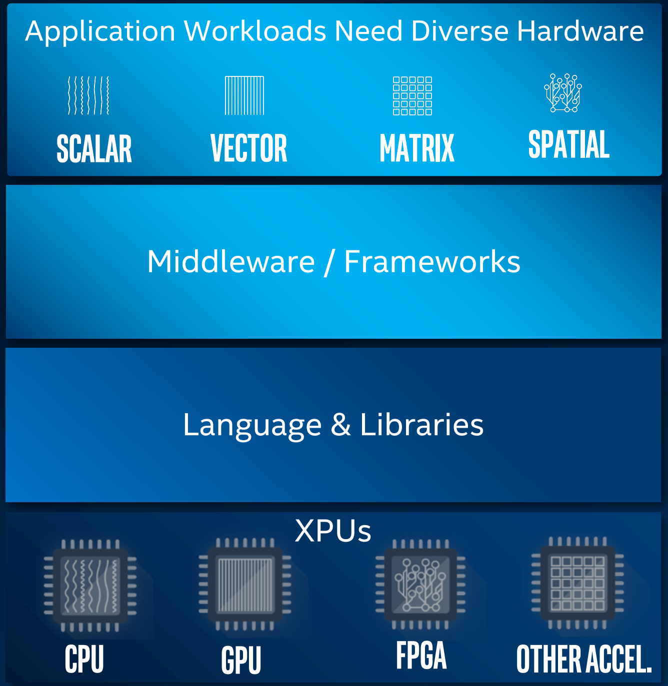
Growth in specialized workloads

Variety of data-centric hardware required

No common programming language or APIs

Inconsistent tool support across platforms

Each platform requires unique software investment



INTRODUCING ONEAPI

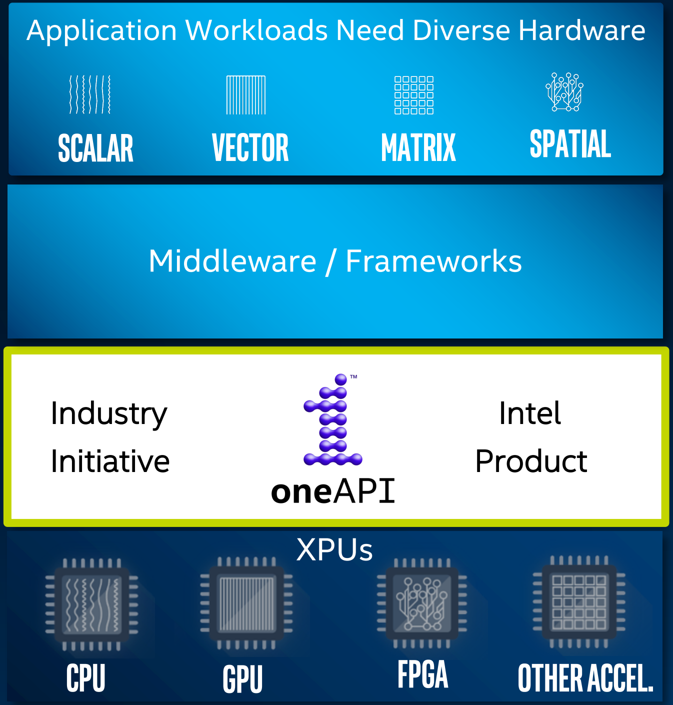
Unified programming model to simplify development across diverse architectures

Unified and simplified language and libraries for expressing parallelism

Uncompromised native high-level language performance

Based on industry standards and open specifications

Interoperable with existing HPC programming models



Refer to software.intel.com/articles/optimization-notice for more information regarding performance & optimization choices in Intel software products.

INTEL® ONEAPI TOOLKITS (BETA)

TOOLKITS TAILORED TO YOUR NEEDS: NATIVE CODE | DATA SCIENTISTS & AI | SYSTEMS

Native Code Developers, start with the Intel® oneAPI Base Toolkit.

Intel® oneAPI Base Toolkit

A core set of high-performance tools for building Data Parallel C++ applications and oneAPI library based applications

Learn More

Add-on Domain-specific Toolkits for Specialized Workloads

Intel® oneAPI HPC Toolkit

Deliver fast C++, Fortran, & OpenMP* applications that scale

Learn More

Intel® oneAPI IoT Toolkit

Building high-performing, efficient, reliable solutions that run at the network's edge

Learn More

Intel® oneAPI DL Framework Developer Toolkit

Build deep learning frameworks or customize existing ones so applications run faster

Learn More

Intel® oneAPI Rendering Toolkit

Create high-performance, high-fidelity visualization applications

Learn More

Toolkits Powered by oneAPI:

Data Scientists & AI Toolkits

Intel® AI Analytics Toolkit

Accelerate E2E machine learning & data science pipelines with optimized DL frameworks & high-performing Python libraries.

Learn More

Intel® Distribution of OpenVINO™ Toolkit

Deploy high performance inference & applications from edge to cloud (production-level tool)

Learn More

Systems Toolkit

Intel® System Bring-Up Toolkit

Debug & tune systems for power & performance

Learn More

Refer to software.intel.com/articles/optimization-notice for more information regarding performance & optimization choices in Intel software products.

oneAPI UNIFIED APIs

INTEL® ONEAPI TOOLKITS

A **single programming model** to deliver cross-architecture performance

Intel Distribution of OpenVINO™ toolkit

Deploy high-performance inference applications from device to cloud

- ✓ OpenCV
- ✓ Intel Deep Learning Deployment Toolkit
- ✓ Inference Support
- ✓ Deep Learning Workbench

Intel AI Analytics Toolkit

Develop machine and deep learning models to generate insights

- ✓ Intel optimization for TensorFlow
- ✓ PyTorch optimized for Intel technology
- ✓ Intel Distribution for Python

Intel oneAPI DL Framework Developer Toolkit

Build deep learning frameworks or customize existing ones

- ✓ Intel oneAPI Collective Library
- ✓ Intel oneAPI Deep Neural Network Library (oneDNN)

CPU | GPU | FPGA | Other accelerators

*Hardware support varies by individual Intel oneAPI tool. FPGA architecture is supported only by the Intel oneAPI base toolkit. Architecture support will be expanded over time.
https://software.intel.com/oneAPI
https://software.intel.com/content/www/us/en/develop/tools/oneapi/dl-framework-developer-toolkit.html

ONEAPI AVAILABLE NOW ON INTEL® DEVCLOUD

A development sandbox to develop, test and run your workloads across a range of Intel CPUs, GPUs, and FPGAs using Intel's oneAPI beta software

software.intel.com/devcloud/oneapi

- Use Intel oneAPI Toolkits
- Learn Data Parallel C++
- Evaluate Workloads
- Build Heterogenous Applications
- Prototype your project

NO DOWNLOADS | NO HARDWARE ACQUISITION | NO INSTALLATION | NO SET-UP & CONFIGURATION

GET UP & RUNNING IN SECONDS!

UNMATCHED SILICON & SOFTWARE FOUNDATION

for AI & analytics

Software & solutions

oneAPI

OpenVINO

intel select solution

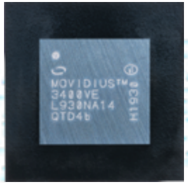
Process

3rd Gen Intel Xeon Scalable processor

GPU

Intel Stratix 10 NX

Gen 3 Intel Movidius VPU



Habana Gaudi & Goya

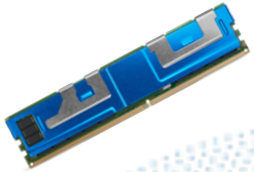
habana®
An Intel Company

CPU | GPU | FPGA | SPECIALIZED ACCELERATORS

WORKLOAD BREADTH — **AI SPECIFIC**

Store

Intel Optane persistent memory 200 series



Intel SSD D7-P5500
Intel SSD D7-P5600

UNLEASHING THE POTENTIAL OF DATA

OPTIMIZED PLATFORM

Move faster

BAREFOOT NETWORKS | an Intel company

Intel® Ethernet

Intel Silicon Photonics

Store more

intel OPTANE™ PERSISTENT MEMORY

intel OPTANE™ SSD

intel 3D NAND SSD

Process everything

intel XEON® SCALABLE inside™

intel ATOM® inside™

intel AGILEX™ inside™

intel eASIC® inside™

intel MOVIDIUS™ inside™

habana® An Intel Company — LIMITED SAMPLING

X — IN DEVELOPMENT

Software- and system-level optimized

OPTIMIZED DEEP LEARNING FRAMEWORKS AND TOOLKITS

GEN ON GEN PERFORMANCE GAINS FOR RESNET-50 WITH INTEL DL BOOST

2S Intel Xeon Platinum 8280 Processor vs 2S Intel Xeon Platinum 8180 Processor

Intel Xeon Scalable Processor	2nd Gen Intel Xeon Scalable Processor	mxnet	PyTorch	TensorFlow	Caffe	OpenVINO
FP32	INT8 w/ Intel DL Boost	3.0x	3.7x	3.9x	4.0x	3.9x
INT8	INT8 w/ Intel DL Boost	1.8x	2.1x	1.8x	2.3x	1.9x

See Configuration Details 5 in backup.

Performance results are based on testing as of dates shown in configuration and may not reflect all publicly available security updates. No product can be absolutely secure. See configuration disclosure for details. 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

ANALYTICS & AI SOFTWARE OPTIMIZATIONS MATTER

IBM Db2
IN-MEMORY DATABASE
4.43X
THROUGHPUT FP32 TO INT8[1]

Microsoft
SQL DATA WAREHOUSING
24.8X
8280 VS 4-YEAR-OLD SYSTEM[2]

sas
BUSINESS ANALYTICS
2.38X
8268 VS E5-2699 V4[3]

ORACLE
TIMESTEN IMDB
6.49X
8260 + INTEL OPTANE DCPMM VS DRAM[4]

H2O.ai
DRIVERLESS AI PLATFORM
4.5X
WITH OPTIMIZED XGBOOST + 8260[5]

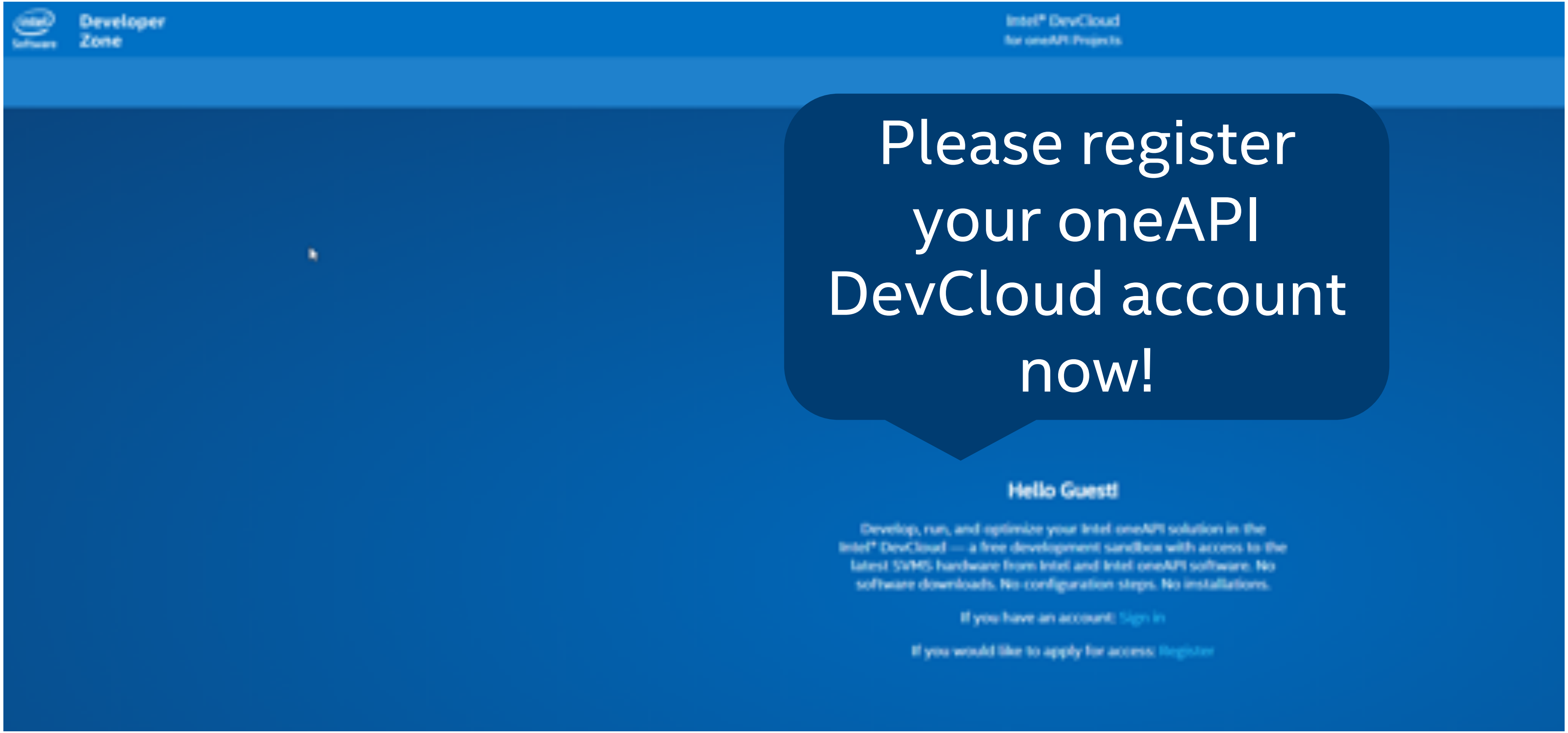
OpenVINO AND TensorFlow
AI INFERENCING SOLUTION
3.75X
WITH OPENVINO OR TENSORFLOW USING INTEL DL BOOST[6]

BigDL for Spark
BIGDL ON APACHE SPARK
5.4X
WITH INTEL OPTIMIZATION OF CAFFE RESNET-50 + 8180[7]

hazelcast
HAZELCAST RESTART TIME
2.5X
WITH INTEL OPTANE DCPMM VS SSDS[8]

For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. See configurations in backup for details.

OPTMIZED ML ON INTEL



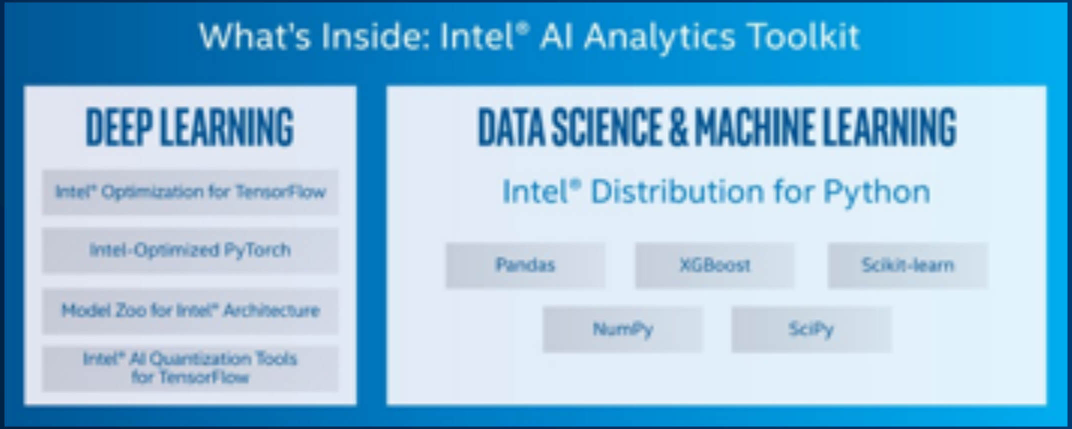
Please register your oneAPI DevCloud account now!

https://devcloud.intel.com/oneapi/get-started/

INTEL® AI ANALYTICS TOOLKIT (BETA)

POWERED BY ONEAPI

A toolkit that helps accelerate end-to-end **machine learning & data science pipelines with optimized DL frameworks & high-performing Python libraries**



Who Uses It?

AI researchers & application developers, data scientists

Key Usages

AI Research & applications across Finance, Retail, E-commerce, Robotics, Transportation & more

Top Features/Benefits

Accelerate end-to-end AI and Data Science pipelines with optimized Python tools built using Intel® oneAPI Libraries

Provides high performance for deep learning training and inference with Intel-optimized TensorFlow and PyTorch

Drop-in acceleration for data science workflows from preprocessing through machine learning

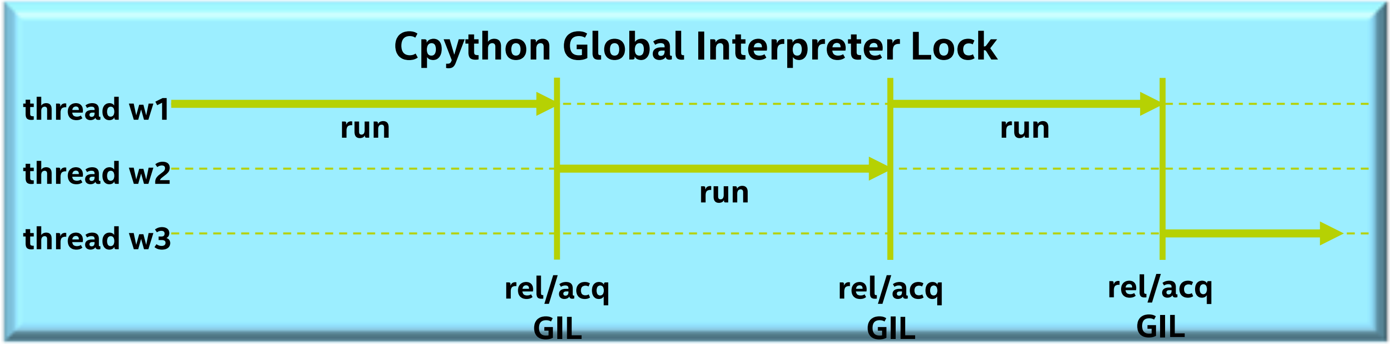
Scale-out efficiently using the high-performing Python packages, such as NumPy, Scikit-learn, XGBoost and more

Supports cross-architecture development and compute (Intel CPUs & future Xe/GPU architecture)

INTRODUCTION TO PYTHON* PERFORMANCE

General Python behavior (Cpython)

- Cpython provides an interpreter to run commands from Python Bytecode (.pyc)
- Compiling doesn’t go down to x86 instructions, but instead
- Python interpreter → Compiled Bytecode → Python Virtual Machine
- Allows for very flexible bytecode, and the Python interpreter is the main ingredient
- Cpython and PyPy have a Global Interpreter Lock (GIL)

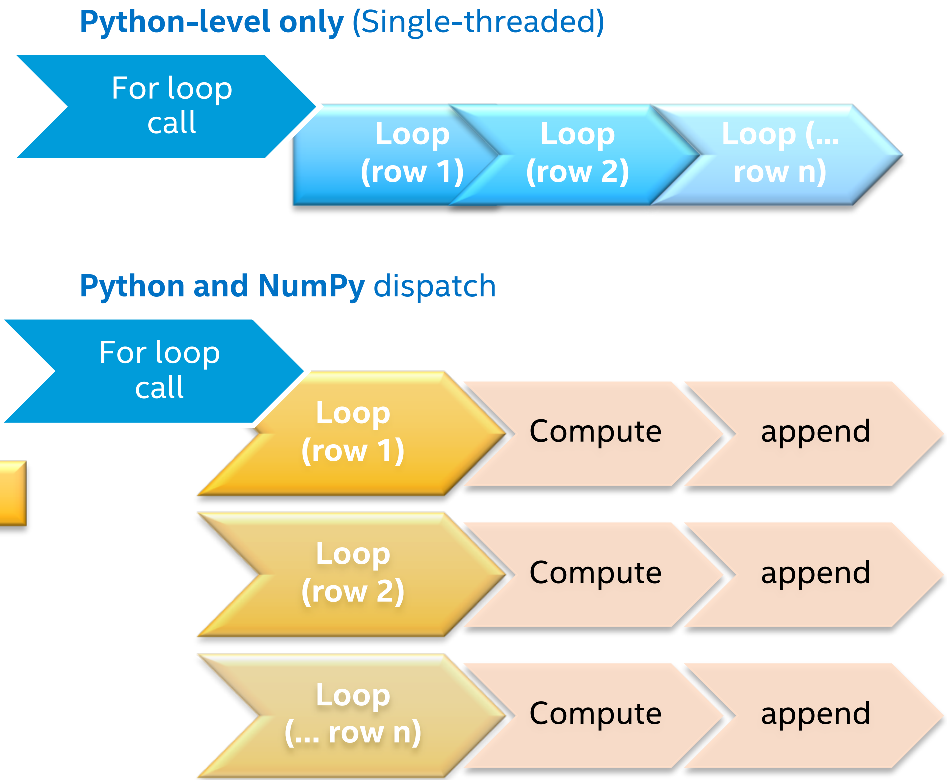


INTRODUCTION TO PYTHON* PERFORMANCE, CONT.

Why does this matter? (Python layers)

- Example with array loops
- GIL will force loops to run in a single threaded fashion
- NumPy* dispatch helps get around single-threaded by using C functions
- C functions can then call processor vectorization

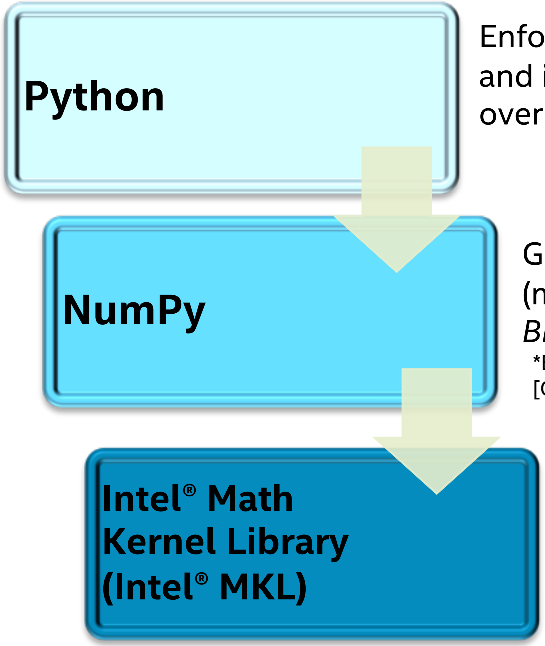
Getting out of Python layer is key for performance



INTRODUCTION TO PYTHON* PERFORMANCE, CONT.

The layers of quantitative Python

- The Python language is interpreted and has many type checks to make it flexible
- Each level has various tradeoffs; *NumPy** value proposition is immediately seen
- For best performance, escaping the Python layer early is best method



Enforces *Global Interpreter Lock* (GIL) and is single-threaded, abstraction overhead. No advanced types.

Gets around the GIL (multi-thread and multi-core)
BLAS API can be the bottleneck
*Basic Linear Algebra Subprograms (BLAS) [CBLAS]

Gets around BLAS API bottleneck
Much stricter typing
Fastest performance level
Dispatches to hardware vectorization

Intel® MKL included with Anaconda* standard bundle; is Free for Python

PERFORMANCE OF PYTHON

Python + Numba*

http://numba.pydata.org/

LLVM-based compiler
Multiple threading runtimes

Small %% performance gap

C

Optimizing compiler
OpenMP*/TBB/pthreads

Operations that can be accelerated using numba

- Basic math and comparison operators
- NumPy ufuncs (that are supported in nopython mode)
- User-defined ufuncs created with numba.vectorize
- Reduction functions: sum, prod
- Array creation: np.ones and np.zeros
- Dot products: vector-vector and matrix-vector

```
@numba.jit(nopython=True, parallel=True)
def logistic_regression(Y, X, w0, step, iterations):
    """SGD solver for binary logistic regression."""
    w = w0.copy()
    for i in range(iterations):
        w += step * np.dot((1.0/(1.0 + np.exp(Y * np.dot(X, w)))) * Y, X)
    return w
```

https://www.anaconda.com/blog/developer-blog/parallel-python-with-numba-and-parallelaccelerator/

INTEL® MKL: PYTHON* INTEGRATION

Python usage

Intel® MKL included in Intel® Distribution of Python*

Numpy accelerated out of the box

No code changes

What MKL brings to Python

Single-Core: vectorization, prefetching, cache utilization

→ SIMD support for AVX-512 ISA

Multi-Many Core (processor/socket) level parallelization

→ OpenMP and TBB support

Multi-Socket (node) level parallelization & Clusters scaling

Requires No Python Code Changes

```
# Calculate with Numpy
import numpy as np
result = np.cov(fullArray, rowvar=False, bias=True)

# Calculate with Scikit-learn
from sklearn.decomposition import PCA
pca = PCA()
pca.fit(npa)
result = pca.get_covariance()
```

ACCELERATE LIBRARIES WITH INTEL® DISTRIBUTION FOR PYTHON*

HIGH PERFORMANCE PYTHON* FOR SCIENTIFIC COMPUTING, DATA ANALYTICS, MACHINE LEARNING

FASTER PERFORMANCE

Performance Libraries, Parallelism, Multithreading, Language Extensions

Accelerated NumPy/SciPy/scikit-learn with Intel® MKL[1] & Intel® DAAL[2]

Data analytics, machine learning with scikit-learn, daal4py

Optimized run-times Intel MPI®, Intel® TBB

Scale with Numba* & Cython*

Includes optimized mpi4py, works with Dask* & PySpark*

Optimized for latest Intel® architecture

GREATER PRODUCTIVITY

Prebuilt & Accelerated Packages

Prebuilt & optimized packages for numerical computing, machine/deep learning, HPC & data analytics

Drop-in replacement for existing Python

Usually No code changes required!

Conda build recipes included in packages

Free download & free for all uses including commercial deployment

ECOSYSTEM COMPATIBILITY

Supports Python 2.7 & 3.6, conda, pip

Compatible & powered by Anaconda*, supports conda & pip

Distribution & individual optimized packages also available at conda &

Intel MKL accelerated Numpy, and scipy now in Anaconda!

Optimizations upstreamed to main Python trunk

Commercial support through Intel® Parallel Studio XE 2018

Intel® Architecture Platforms

Operating System: Windows*, Linux*, MacOS[1]*

[1]Intel® Math Kernel Library
[2]Intel® Data Analytics Acceleration Library

INTEL® DISTRIBUTION PYTHON* DISTRIBUTION CHANNELS

Standalone Installer
Free Download

https://software.intel.com/en-us/distribution-for-python

Open-source Channels

ANACONDA CLOUD

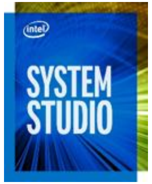
apt-get

yum

python Package Index

docker

Intel Software Tools suite



Windows*

Linux*

macOS*

2.7 & 3.6

SPEED-UP MACHINE LEARNING AND ANALYTICS WITH INTEL® DATA ANALYTICS ACCELERATION LIBRARY (INTEL® DAAL)

Boost Machine Learning & Data Analytics Performance

- Helps applications deliver better predictions faster
- Optimizes data ingestion & algorithmic compute together for highest performance
- Supports offline, streaming & distributed usage models to meet a range of application needs
- Split analytics workloads between edge devices and cloud to optimize overall application throughput

Learn More: software.intel.com/daal

What's New in the 2020 Release

New Algorithms

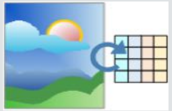
- **High performance Multiclass Adaboost**, widely-used classification algorithm
- **Extended Gradient Boosting Functionality** provides probabilistic classification and variable importance
- **Extended Decision Tree Functionality** provides probabilistic classification and weighted data

Pre-processing → Transformation → Analysis → Modeling → Validation → Decision Making

Pre-processing

Decompression, Filtering, Normalization

Transformation



Aggregation, Dimension Reduction

Analysis

Summary Statistics Clustering, etc.

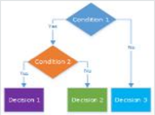
Modeling

Machine Learning (Training) Parameter Estimation Simulation

Validation

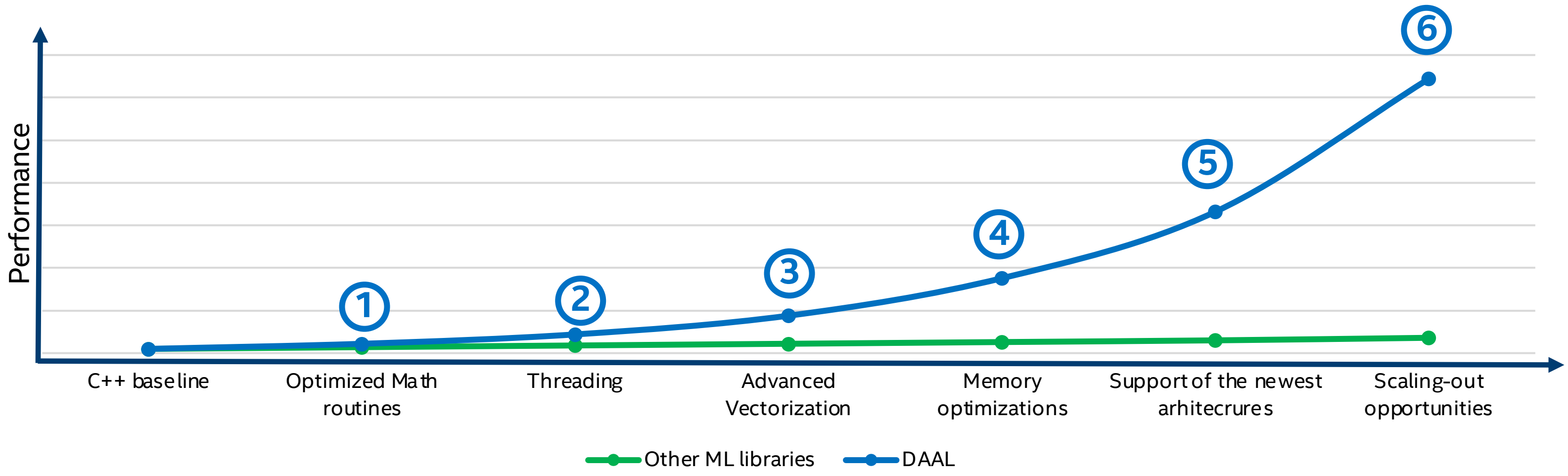
Hypothesis Testing Model Errors

Decision Making



Forecasting Decision Trees, etc.

WHAT MAKES INTEL® DAAL FASTER?



1 The best performance on Intel Architectures with Intel® MKL vs. less performance OS BLAS/LAPACK libs

2 Intel® DAAL targets to many-core systems to achieve the best scalability on Intel® Xeon, other libs mostly target to client versions with small amount of cores

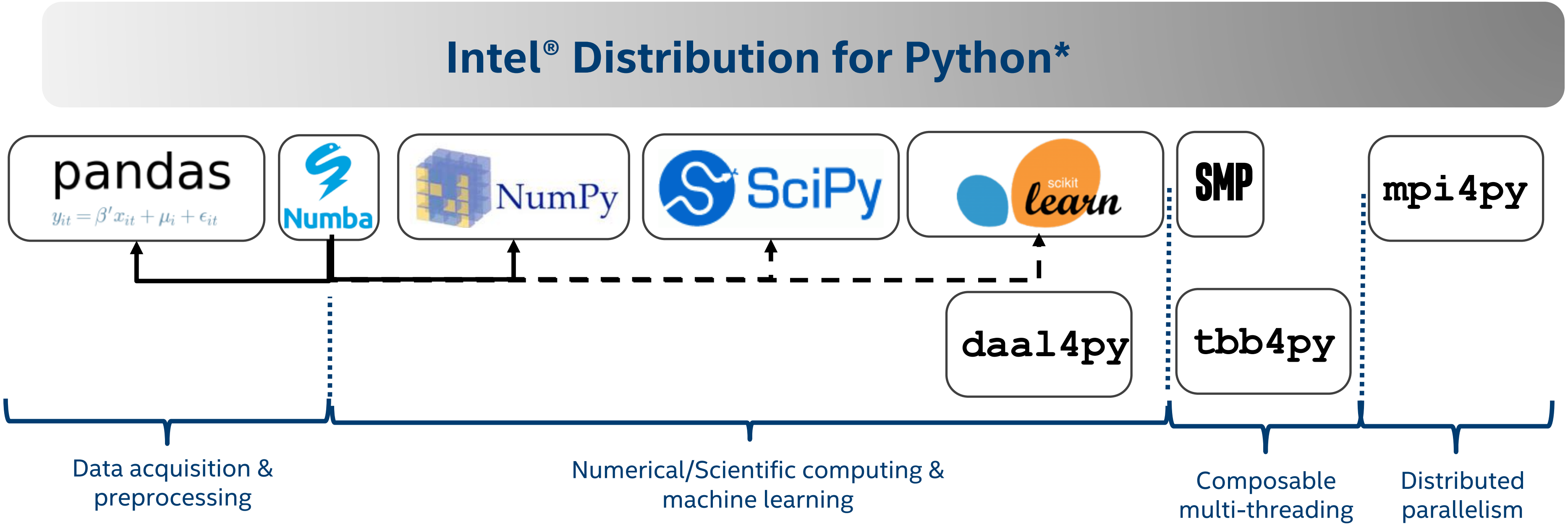
3 Intel® DAAL uses the latest available vector-instructions on each architecture, enables them by compiler options, intrinsics. Usually other ML libs build application without vector-instructions support or sse4.2 only.

4 Intel® DAAL's uses the most efficient memory optimization practices: minimally access memory, cache access optimizations, SW memory prefetching. Usually Other ML libs don't make low-level optimizations.

5 Intel® DAAL enables new instruction sets and other HW features even before official HW lunch. Usually other ML libs do this with long delay.

6 Intel® DAAL provides distributed algorithms which scale on many nodes

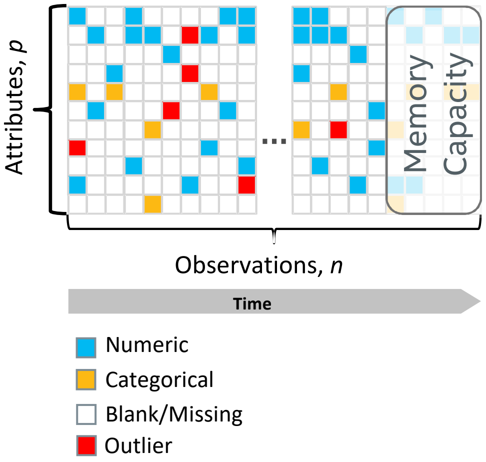
PRODUCTIVITY WITH PERFORMANCE VIA INTEL® PYTHON*



Learn More: software.intel.com/distribution-for-python

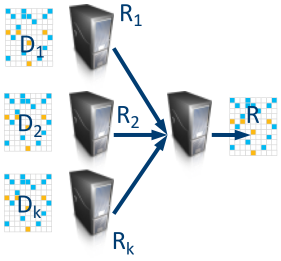
https://www.anaconda.com/blog/developer-blog/parallel-python-with-numba-and-parallelaccelerator/

COMPUTATIONAL ASPECTS OF BIG DATA ADDRESSED BY DAAL



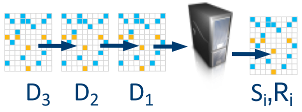
Big Data Attributes	Computational Solution
Distributed across different nodes/devices	•Distributed computing, e.g. comm-avoiding algorithms
Huge data size not fitting into node/device memory	•Distributed computing •Streaming algorithms
Data coming in time	•Data buffering •Streaming algorithms
Non-homogeneous data	•Categorical→Numeric (counters, histograms, etc) •Homogeneous numeric data kernels • Conversions, Indexing, Repacking
Sparse/Missing/Noisy data	•Sparse data algorithms •Recovery methods (bootstrapping, outlier correction)

Distributed Computing



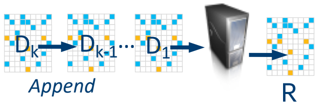
$R = F(R_1,\ldots,R_k)$

Streaming Computing



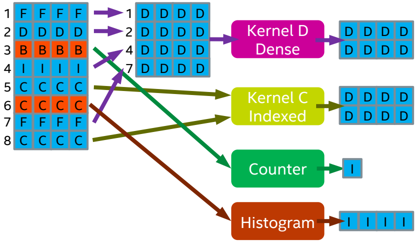
$S_{i+1} = T(S_i,D_i)$
$R_{i+1} = F(S_{i+1})$

Offline Computing

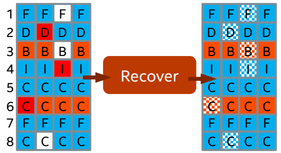


$R = F(D_1,\ldots,D_k)$

Converts, Indexing, Repacking

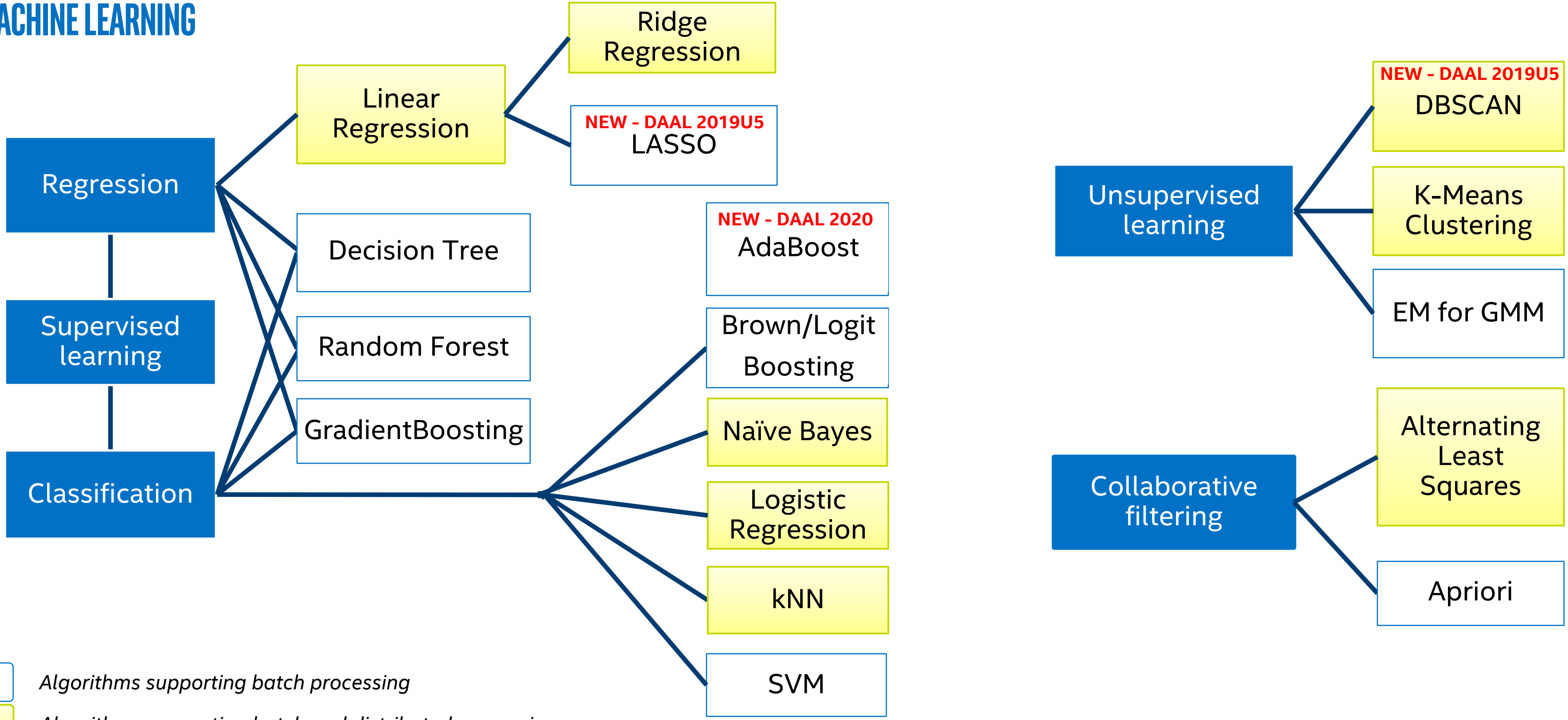


Data Recovery



INTEL® DAAL ALGORITHMS

MACHINE LEARNING



Algorithms supporting batch processing

Algorithms supporting batch and distributed processing

INTEL® DAAL ALGORITHMS
DATA TRANSFORMATION AND ANALYSIS

- Basic statistics for datasets
  - Low order moments
  - Quantiles
  - Order statistics
- Correlation and dependence
  - Cosine distance
  - Correlation distance
  - Variance-Covariance matrix
- Matrix factorizations
  - SVD
  - QR
  - Cholesky
  - tSVD
- Dimensionality reduction
  - PCA
- Outlier detection
  - Univariate
  - Multivariate

Association rule mining (Apriori)

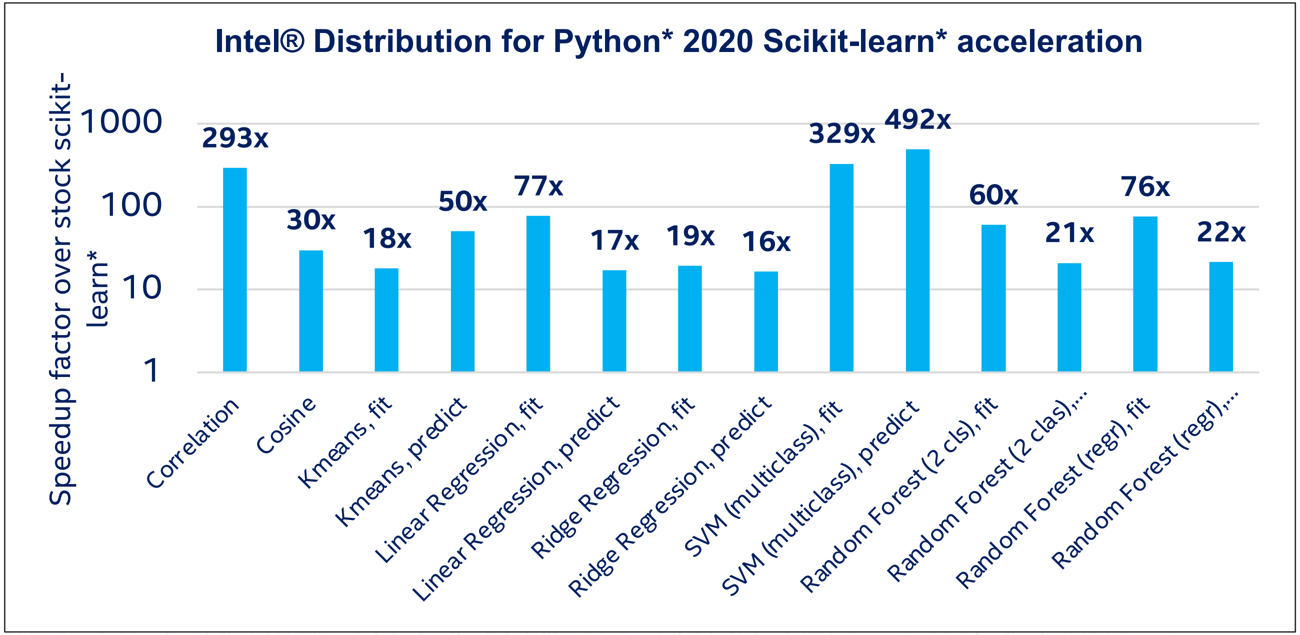
Optimization solvers (SGD, AdaGrad, lBFGS, CD)

Math functions (exp, log,...)

Algorithms supporting batch processing

Algorithms supporting batch, online and/or distributed processing

INTEL SCIKIT-LEARN

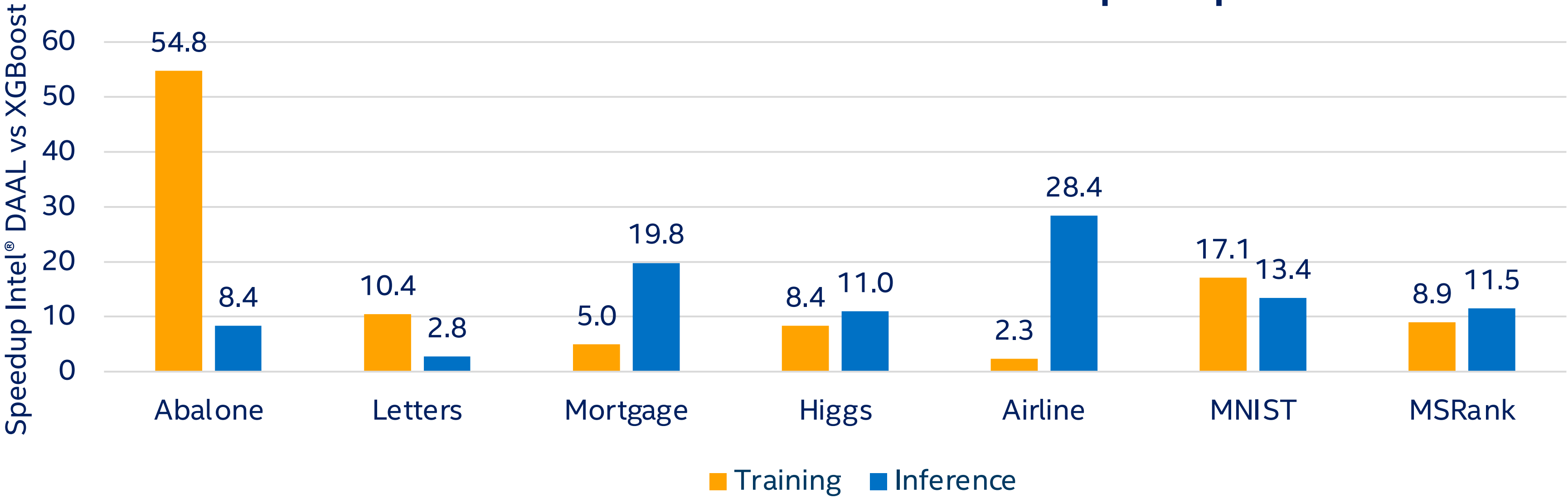


Same Code, Same Behavior

PASSED

- Scikit-learn, <u>not</u> scikit-learn-*like*
- Scikit-learn conformance (mathematical equivalence) defined by Scikit-learn Consortium, continuously vetted by public CI

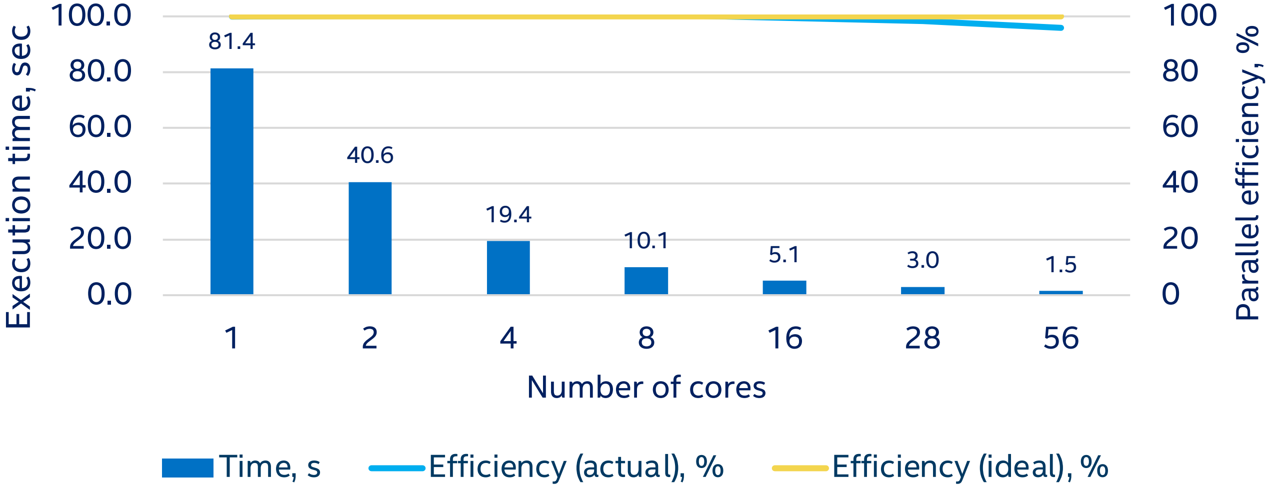
Gradient Boosting performance (Higher is better)
Intel® DAAL 2020 vs DMLC XGBoost* 0.9 speedup



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.
Performance results are based on testing as of **11/11/2019** and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.
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 www.intel.com/benchmarks.
Configuration: Testing by Intel as of **11/11/2019**. CPU configuration: c5.metal AWS Instance (2nd Generation Intel® Xeon® Scalable Processors 2 sockets, HT:on, Turbo:on, OS: Ubuntu 18.04.2 LTS, Total Memory 193 GB (12 slots/16GB/2933 MHz), BIOS: 1.0 Amazone EC2 (ucode: 0x5000017), OMP Environment: OMP_NUM_THREADS=48 OMP_PLACES={0}:96:1). SW: XGBoost 0.9 download from PIP, Intel DAAL: 2020 version. Python env: Python 3.6, Numpy 1.16.4, Pandas 0.25, Scikit-lean 0.21.2. Parameters for XGBoost: on CPU = { 'alpha': 0.9, 'max_bin': 256, 'scale_pos_weight': 2, 'learning_rate': 0.1, 'subsample': 1, 'reg_lambda': 1, "min_child_weight": 0, 'max_depth': 8, 'max_leaves': 2**8, 'tree_method': 'hist', 'predictor': 'cpu_predictor' }. Parameters for XGB on GPU { 'alpha': 0.9, 'max_bin': 256, 'scale_pos_weight': 2, 'learning_rate': 0.1, 'subsample': 1, 'reg_lambda': 1, "min_child_weight": 0, 'max_depth': 8, 'max_leaves': 2**8, 'tree_method': 'gpu_hist', 'predictor': 'gpu_predictor' }. Input data format: Numpy array . objective functions: 'binary:logistic' for binary classification, 'multi:softmax' for multiclass, 'reg:squarederror' for regressions. Number of iterations: 200 for MNIST, 100 for Mortgage, 1000 for others
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

Intel® DAAL 2020 K-means fit, cores scaling

(10M samples, 10 features, 100 clusters, 100 iterations, float32)



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. Performance results are based on testing as of **11/11/2019** and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.

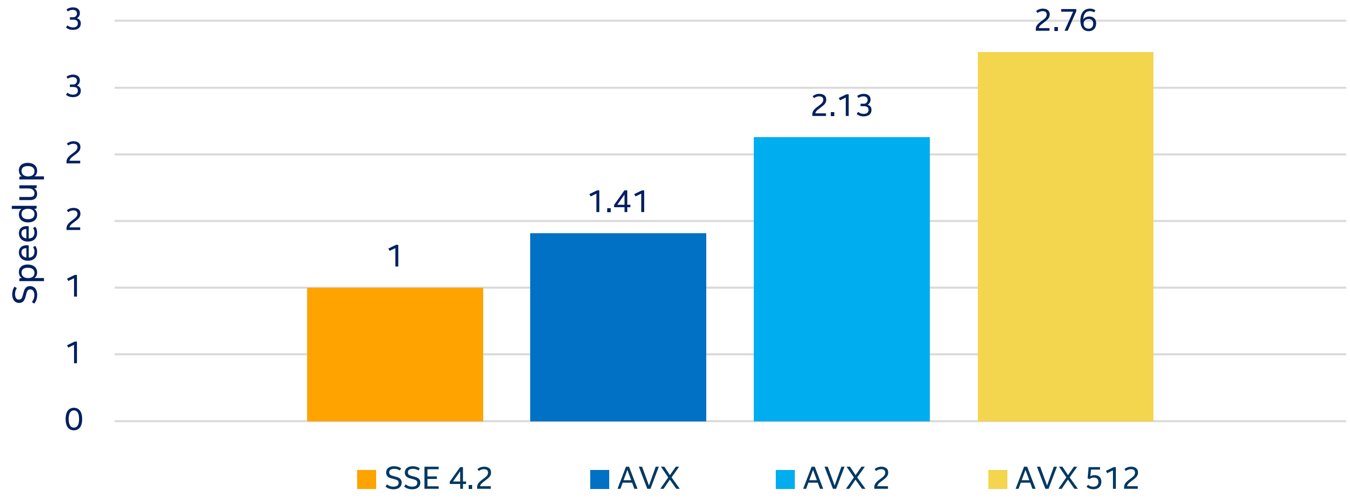
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may 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 www.intel.com/benchmarks.

Configuration: Testing by Intel as of **11/11/2019**. Intel® Data Analytics Acceleration Library 2019.3 (Intel® DAAL); Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz, 2 sockets, 28 cores per socket, 10M samples, 10 features, 100 clusters, 100 iterations, float32

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

Intel® DAAL 2020 K-means fit, vectorization gain

(10M samples, 10 features, 100 clusters, 100 iterations, float32)



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. Performance results are based on testing as of **11/11/2019** and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.

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 www.intel.com/benchmarks.

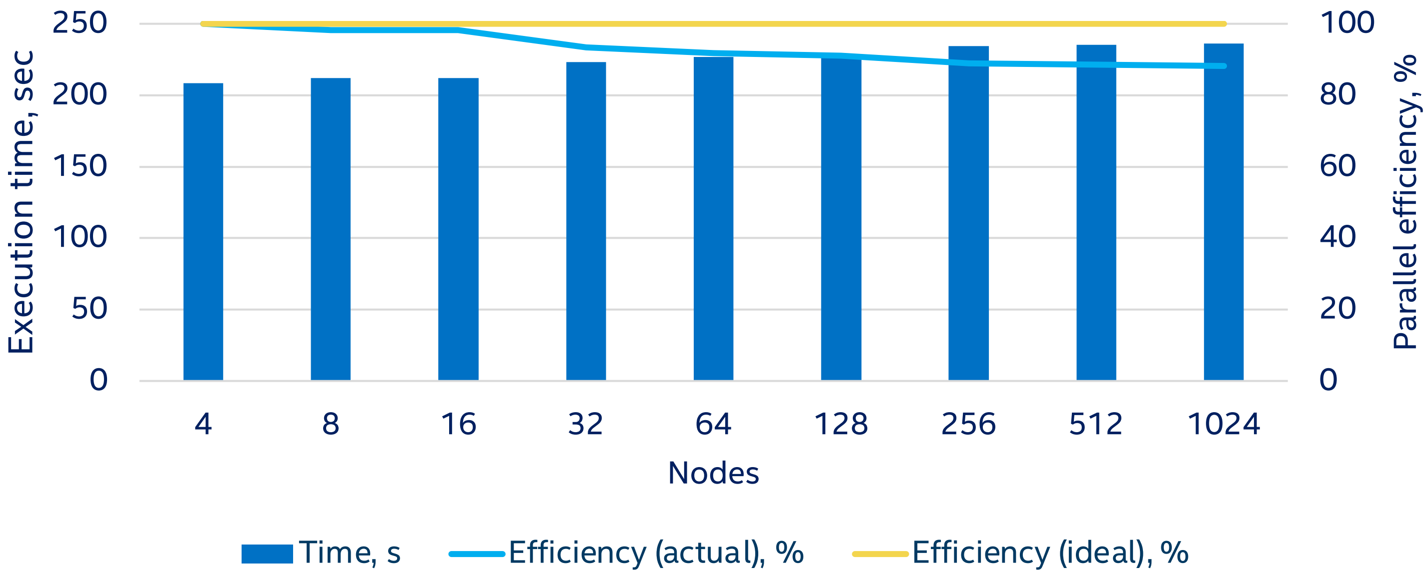
Configuration: Testing by Intel as of **11/11/2019**. Intel® Data Analytics Acceleration Library 2019.3 (Intel® DAAL); Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz, 2 sockets, 28 cores per socket, 10M samples, 10 features, 100 clusters, 100 iterations, float32

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

Optimization Notice

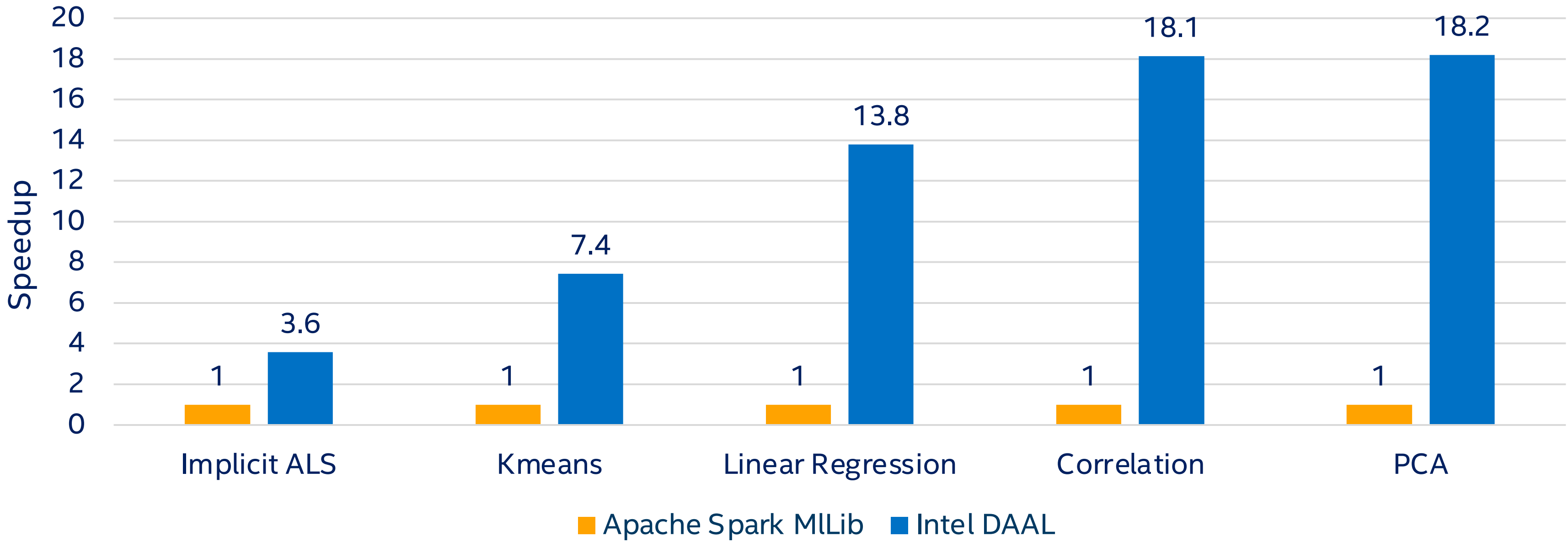
Intel® DAAL K-means fit, week scaling results

(87.44GB/node, 84 features, 8 clusters, 100 iterations, float32)



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.
Performance results are based on testing as of **09/25/2019** and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.
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 www.intel.com/benchmarks.
Configuration: Testing by Intel as of **09/25/2019**. 7 x m5.2xlarge AWS instances, Intel® Data Analytics Acceleration Library 2019.3 (Intel® DAAL); Intel Xeon Processor E5-2698 v3 @ 2.3GHz, 2 sockets, 16 cores per socket, MPI4Py (3.0.0), Intel® Distribution Of Python (IDP) 3.6.8, float, Source: https://arxiv.org/abs/1909.11822
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

Intel® DAAL 2020 vs Apache Spark* MlLib performance (Higher is better)



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.
Performance results are based on testing as of **11/11/2019** and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.
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 www.intel.com/benchmarks.
Configuration: Testing by Intel as of **11/11/2019**. 7 x m5.2xlarge AWS instances, Intel® Data Analytics Acceleration Library 2020 (Intel® DAAL); Correlation (# samples = 10M, # features = 1000, (Intel® DAAL=35.2s, MLLib=638.2s)), PCA (# samples = 10M, # features = 1000 (Intel® DAAL=35.2s, MLLib=639.8s)), implicit ALS (# users = 1M, # items = 1M, # factors = 100, # Iterations = 1 (Intel® DAAL=37.6s, MLLib=134.9s)), Linear Regression (# samples = 100M, # features = 50 (Intel® DAAL=16.3s, MLLib=224.5s)), k-means (# samples = 100M, # features = 50, # clusters = 10, # Iterations = 100 (Intel® DAAL=211s, MLLib=1567.3s))
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

SCIKIT-LEARN

Top Open Source ML Library (Python)

- Large # of ML algorithms, user-friendly
- Self-reported 500K users (Intel estimated 2M): 60% academia, 40% industry
- Backed by INRIA (French national research institute)

scikit learn

consortium

new

Vendor Consortium announced in September 2018

- Broadest enabling path for optimizations
- Intel, NVidia, Microsoft has joined it.

SIMPLIFIED HL PYTHON API FOR EASE OF USE (DAAL4PY)

- Code for distributed algorithms is up to 100x smaller

- Code for batch algorithms is up to 10x smaller

python™ **De-Facto #1 language for Data Science**

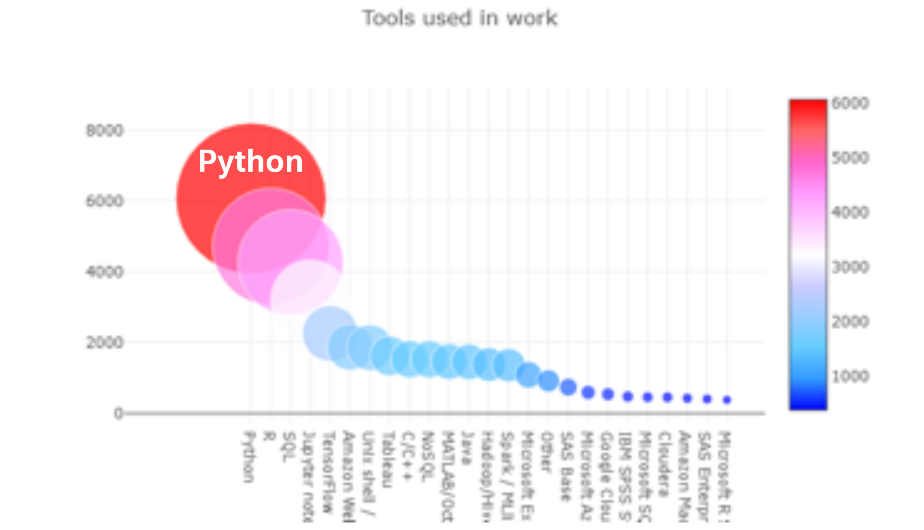
PCA algorithm

pyDAAL (previous)

daal4py

1-line per classical algorithm

Kaggle ML and Data Science Survey, 2017



https://www.kaggle.com/sudalairajkumar/an-interactive-deep-dive-into-survey-results/data

ACCELERATING SCIKIT-LEARN THROUGH DAAL4PY

```
> python -m daal4py <your-scikit-learn-script>
```

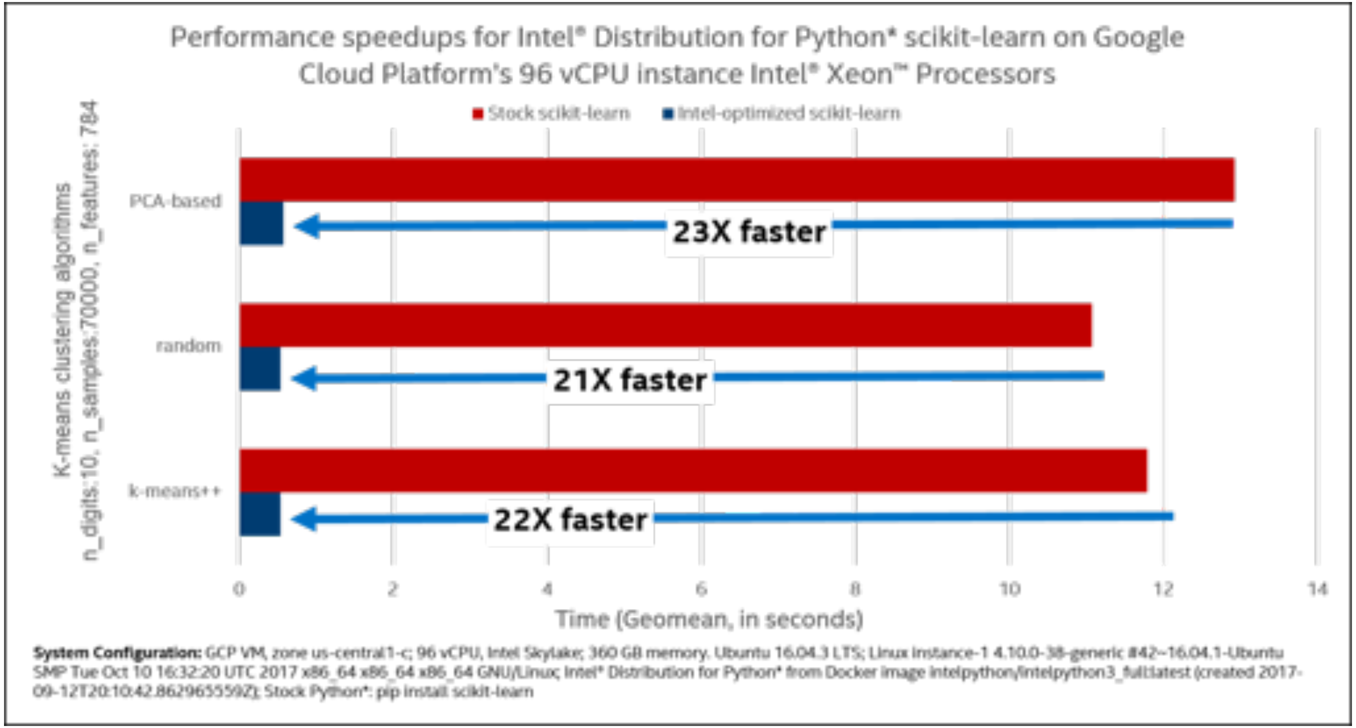
Monkey-patch any scikit-learn on the command-line

```
import daal4py.sklearn
daal4py.sklearn.patch_sklearn('kmeans')
```

Monkey-patch any scikit-learn programmatically

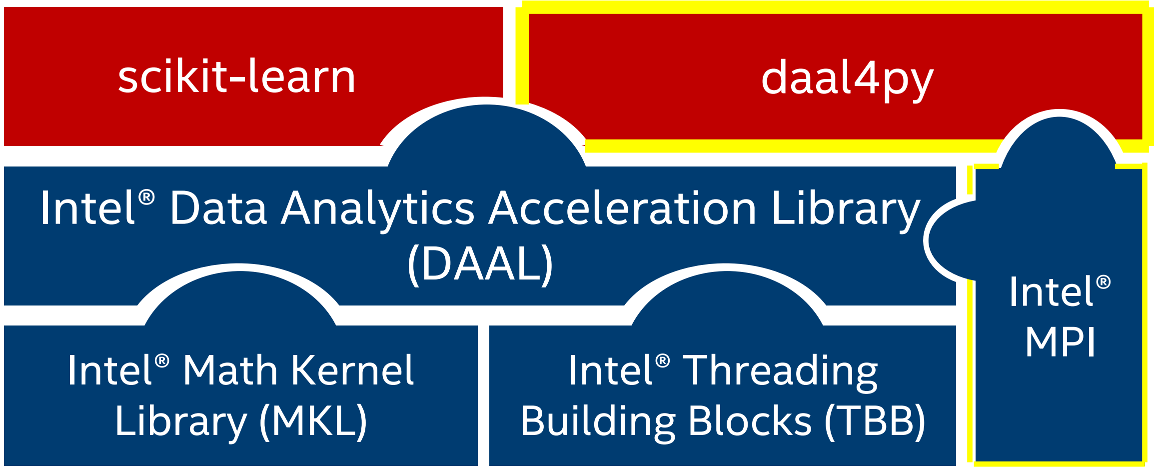
Scikit-learn with daal4py patches applied passes scikit-learn test-suite

ACCELERATING K-MEANS



https://cloudplatform.googleblog.com/2017/11/Intel-performance-libraries-and-python-distribution-enhance-performance-and-scaling-of-Intel-Xeon-Scalable-processors-on-GCP.html

SCALING MACHINE LEARNING BEYOND A SINGLE NODE



Simple Python API
Powers scikit-learn

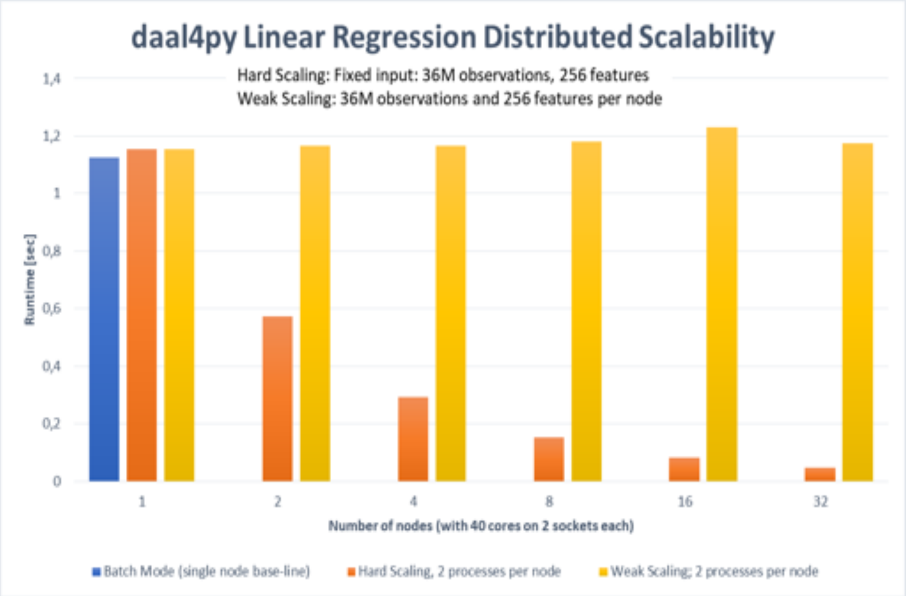
Powered by DAAL

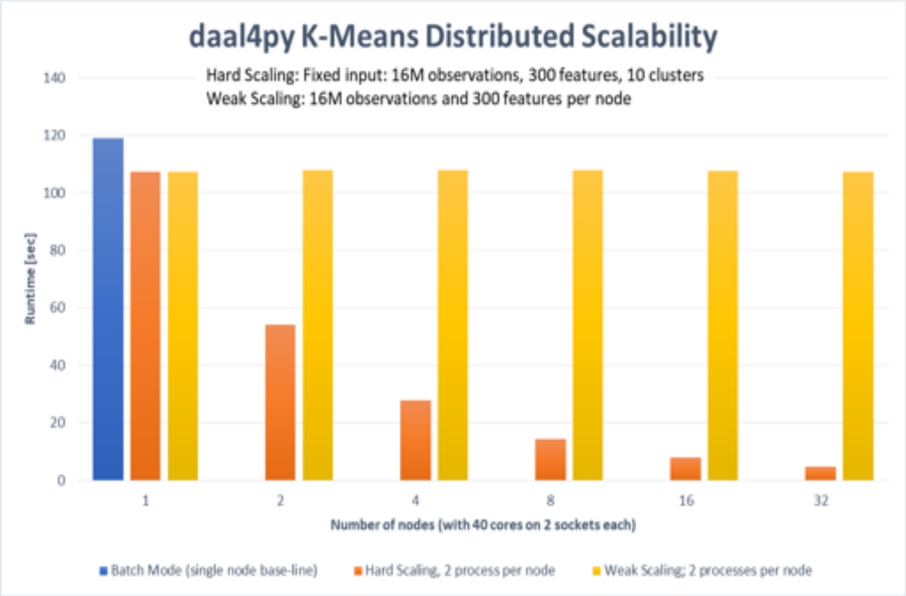
Scalable to multiple nodes

Try it out! `conda install -c intel daal4py`

WORKING IN DISTRIBUTED ENVIRONMENT

Hardware	Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz, EIST/Turbo on 2 sockets, 20 Cores per socket 192 GB RAM 16 nodes connected with Infiniband
Operating System	Oracle Linux Server release 7.4
Data Type	double





On a 32-node cluster (1280 cores) daal4py computed linear regression of 2.15 TB of data in 1.18 seconds and 68.66 GB of data in less than 48 milliseconds.

On a 32-node cluster (1280 cores) daal4py computed K-Means (10 clusters) of 1.12 TB of data in 107.4 seconds and 35.76 GB of data in 4.8 seconds.

K-MEANS USING DAAL4PY

```
import daal4py as d4p

# daal4py accepts data as CSV files, numpy arrays or pandas dataframes
# here we let daal4py load process-local data from csv files
data = "kmeans_dense.csv"

# Create algob object to compute initial centers
init = d4p.kmeans_init(10, method="plusPlusDense")
# compute initial centers
ires = init.compute(data)
# results can have multiple attributes, we need centroids
Centroids = ires.centroids
# compute initial centroids & kmeans clustering
result = d4p.kmeans(10).compute(data, centroids)
```

DISTRIBUTED K-MEANS USING DAAL4PY

```
import daal4py as d4p

# initialize distributed execution environment
d4p.daalinit()

# daal4py accepts data as CSV files, numpy arrays or pandas dataframes
# here we let daal4py load process-local data from csv files
data = "kmeans_dense_{}.csv".format(d4p.my_procid())

# compute initial centroids & kmeans clustering
init = d4p.kmeans_init(10, method="plusPlusDense", distributed=True)
centroids = init.compute(data).centroids
result = d4p.kmeans(10, distributed=True).compute(data, centroids)
```

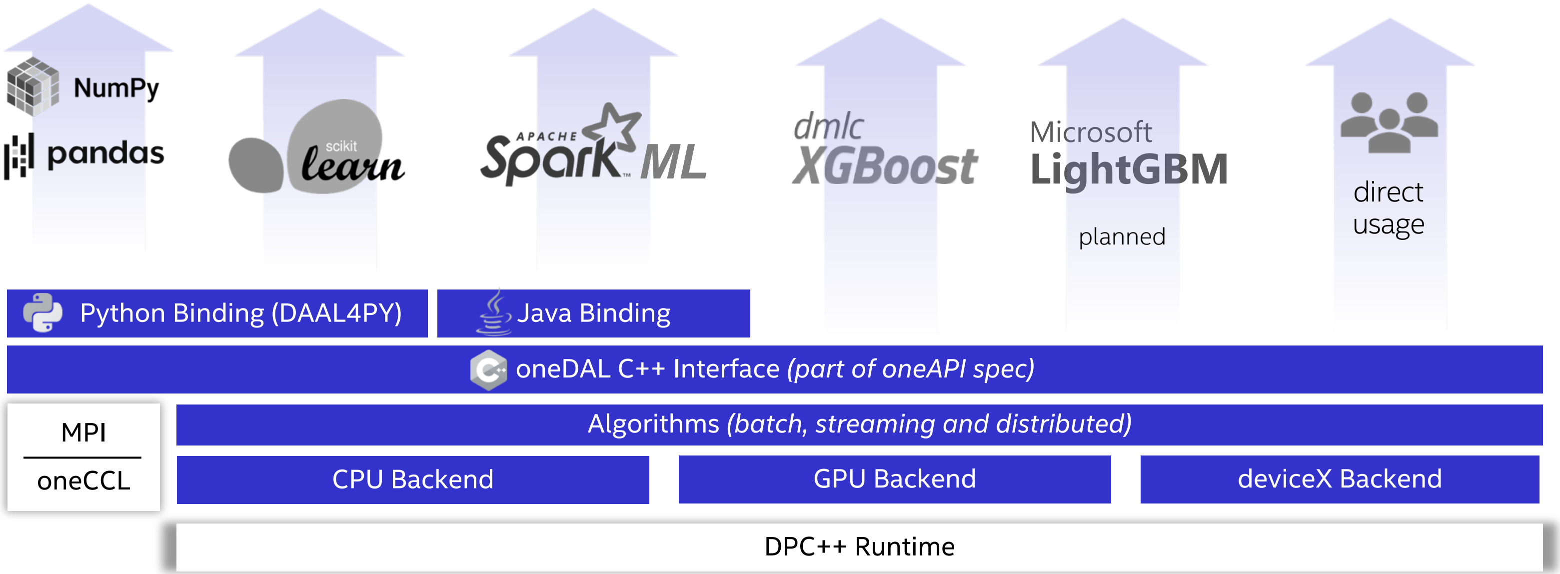
```
mpirun -n 4 python ./kmeans.py
```

STREAMING DATA (LINEAR REGRESSION) USING DAAL4PY

```
import daal4py as d4p

# Configure a Linear regression training object for streaming
train_algo = d4p.linear_regression_training(interceptFlag=True, streaming=True)

# assume we have a generator returning blocks of (X,y)...
rn = read_next(infile)

# on which we iterate
for chunk in rn:
    algo.compute(chunk.X. chunk.y)

# finalize computation
result = algo.finalize()
```

ONEAPI DATA ANALYTICS LIBRARY (ONEDAL)

Open Source Implementation



github.com/oneapi-src/oneDAL

ACCELERATED SCIKIT-LEARN USING ONEAPI

Common Scikit-learn

```
from sklearn.svm import SVC

X, Y = get_dataset()

clf = SVC().fit(X, y)
res = clf.predict(X)
```

Scikit-learn mainline

Scikit-learn with Intel CPU opts

```
import daal4py as d4p
d4p.patch_sklearn()
from sklearn.svm import SVC

X, Y = get_dataset()

clf = SVC().fit(X, y)
res = clf.predict(X)
```

Available through Intel conda
(conda install daal4py –c intel)

Run Scikit-learn on Intel GPU

```
import daal4py as d4p
d4p.patch_sklearn()
from sklearn.svm import SVC

X, Y = get_dataset()

with d4p.sycl_context(“gpu”):
    clf = SVC().fit(X, y)
    res = clf.predict(X)
```

In progress

PROFILING

TUNE PYTHON* + NATIVE CODE FOR BETTER PERFORMANCE

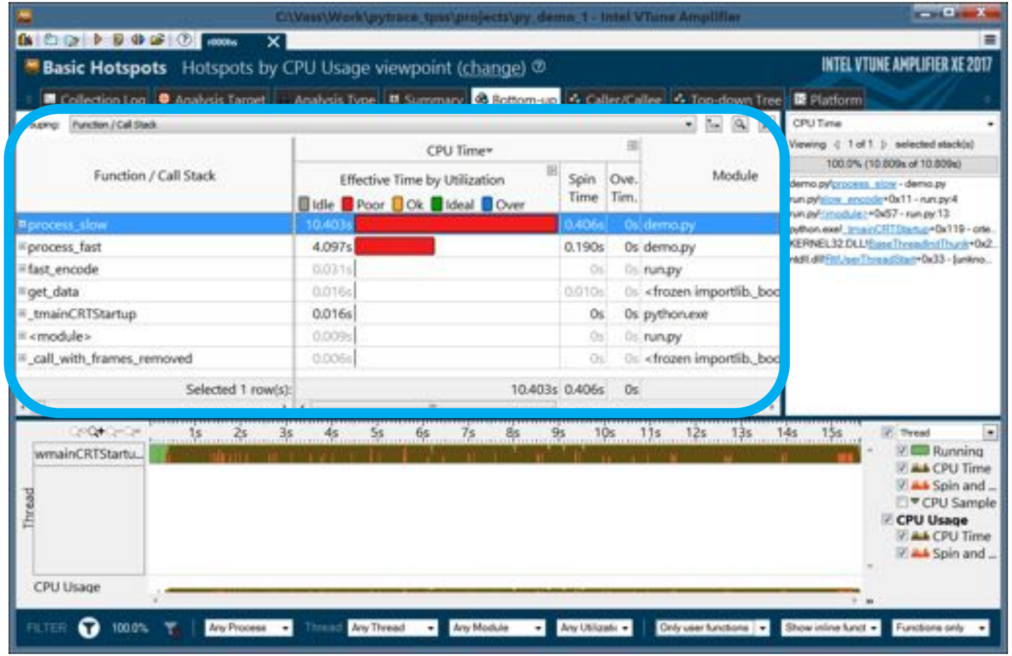
ANALYZE PERFORMANCE WITH INTEL® VTUNE™ AMPLIFIER (AVAILABLE IN INTEL® PARALLEL STUDIO XE)

Challenge

- Single tool that profiles Python + native mixed code applications
- Detection of inefficient runtime execution

Solution

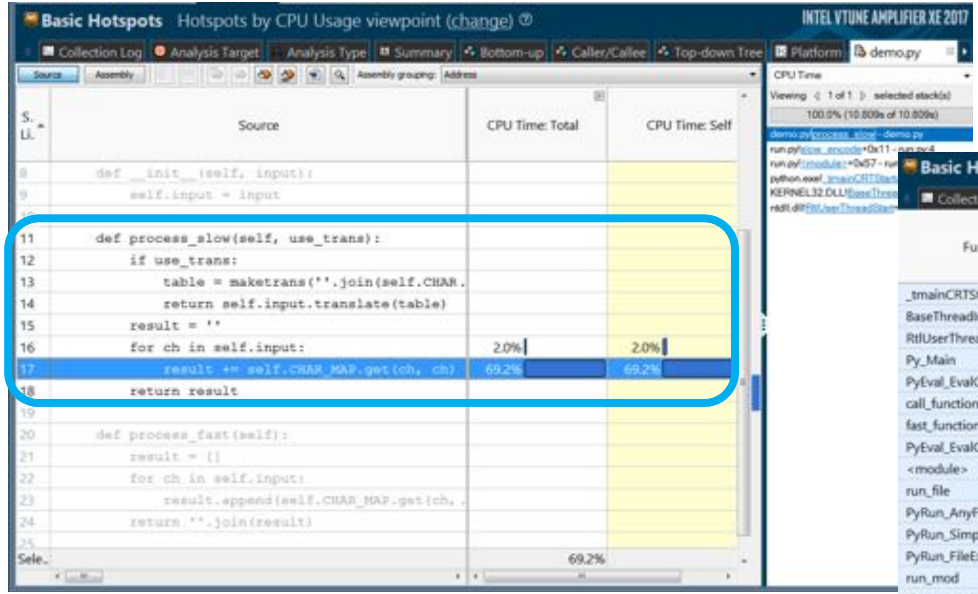
- Auto-detect mixed Python/C/C++ code & extensions
- Accurately identify performance hotspots at line-level
- Low overhead, attach/detach to running application
- Focus your tuning efforts for most impact on performance



Auto detection & performance analysis of Python & native functions

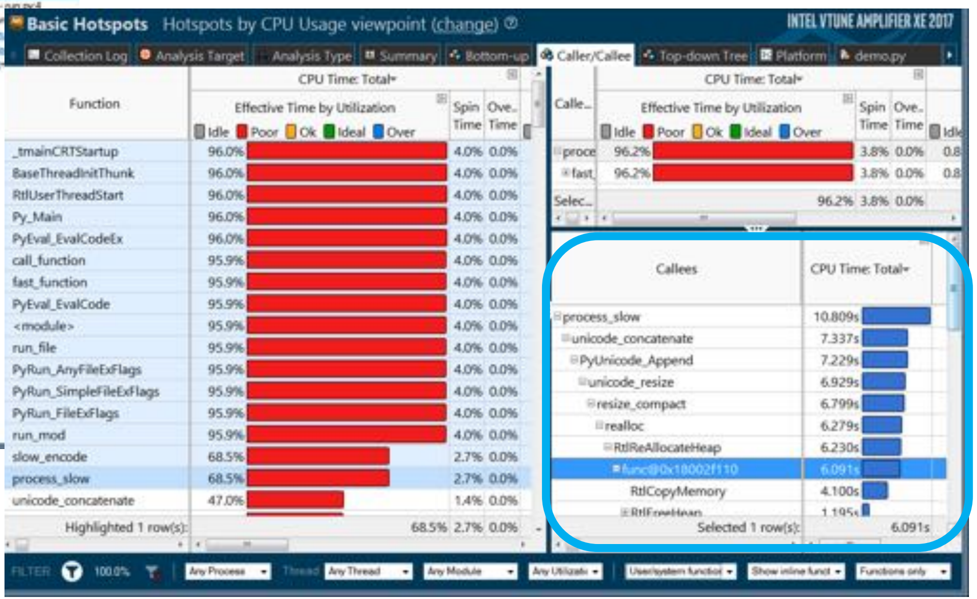
Available in Intel® VTune™ Amplifier & Intel® Parallel Studio XE

DIAGNOSE PROBLEM CODE QUICKLY & ACCURATELY

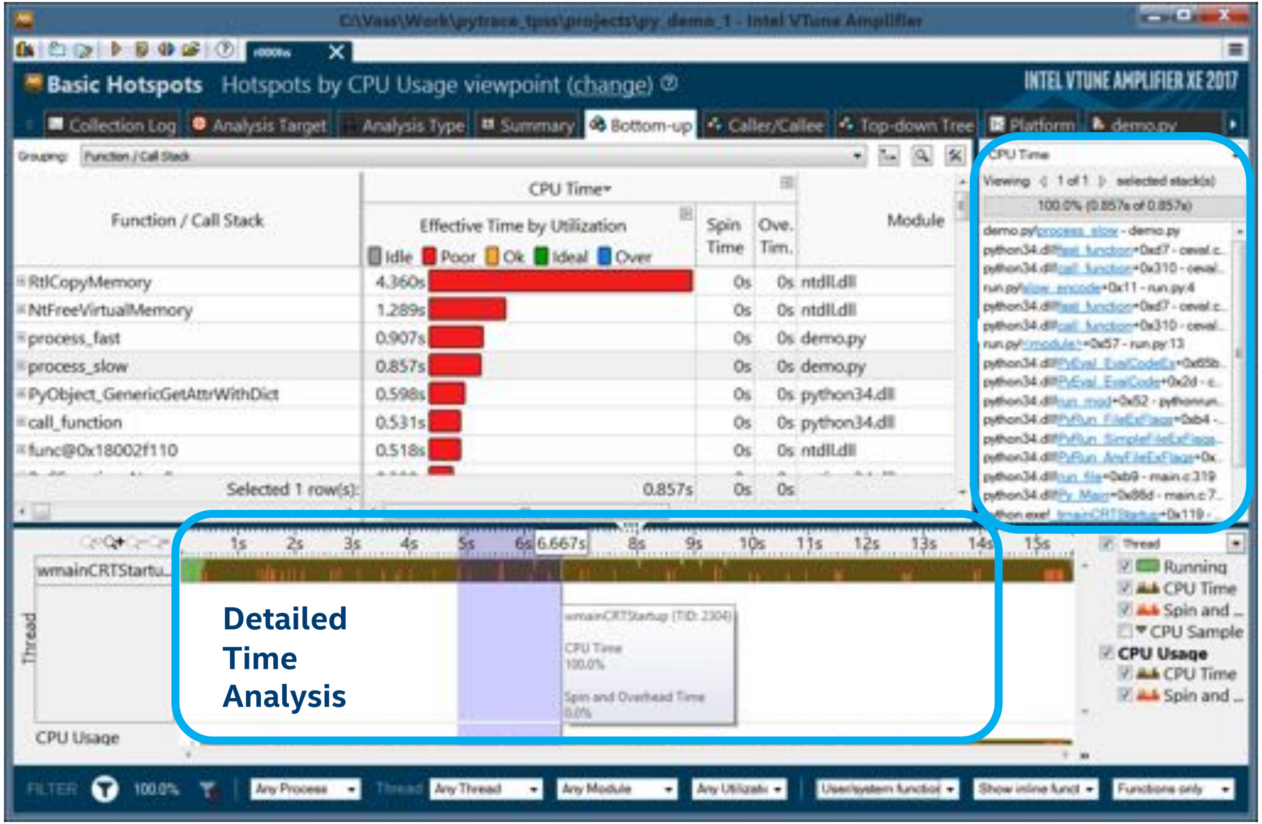


Identifies exact line of code that is a bottleneck

Details Python* calling into native functions



DEEPER ANALYSIS WITH CALL STACK LISTING & TIME ANALYSIS



Call Stack Listing for Python* & Native Code

A 2-PRONG APPROACH FOR FASTER PYTHON* PERFORMANCE

HIGH PERFORMANCE PYTHON DISTRIBUTION + PERFORMANCE PROFILING

Step 1: Use Intel® Distribution for Python

- Leverage optimized native libraries for performance
- Drop-in replacement for your current Python - no code changes required
- Optimized for multi-core and latest Intel processors

Step 2: Use **Intel® VTune™ Amplifier** for profiling

- Get detailed summary of entire application execution profile
- Auto-detects & profiles Python/C/C++ mixed code & extensions with low overhead
- Accurately detect hotspots - line level analysis helps you make smart optimization decisions fast!
- Available in Intel® Parallel Studio XE Professional & Cluster Edition

MORE RESOURCES

Intel® Distribution for Python

- Product page – overview, features, FAQs…
- Training materials – movies, tech briefs, documentation, evaluation guides…
- Support – forums, secure support…

Intel® VTune Amplifier

- Product page – overview, features, FAQs…
- Training materials – movies, tech briefs, documentation, evaluation guides…
- Reviews
- Support – forums, secure support…

Intel® DAAL Product Information

- http://software.intel.com/en-us/intel-daal

Intel® DAAL Getting Started Guides

- https://software.intel.com/en-us/intel-daal-support/training
- **DAAL4PY Examples:** https://github.com/IntelPython/daal4py/tree/master/examples
- DAAL4PY docs: https://intelpython.github.io/daal4py/
- **OneAPI-Samples:** https://github.com/oneapi-src/oneAPI-samples/
- **Workshop example:** https://github.com/IntelAI/unet/tree/master/single-node

ONEAPI RESOURCES

Use *Slideshow mode* to click links

oneAPI

oneAPI Industry Initiative

- oneAPI Initiative site | Overview video [3.40]
- oneAPI Industry Specification
- Ecosystem Support

Data Parallel C++ (DPC++)

- Videos
  - DPC++ Overview [3.41]
  - DPC++: Open Alternative for Cross-Architecture Development Q&A - Intel Senior Fellow Geoff Lowney [12.05]
- DPC++ open source project on GitHub
- oneAPI Programming Guide
- DPC++ book 4 preview chapters

Intel® oneAPI Products

Includes domain-specific toolkits

- Intel® oneAPI Toolkits
  - Product Brief
  - Documentation
  - Training
  - Code Samples to get started (see domain-specific toolkits for their samples)
- Intel® DevCloud – Test workloads, code & oneAPI tools on a variety of Intel® architecture - free-of-charge

Free oneAPI, DPC++ & Intel oneAPI Products webinars & quick how-to's

Refer to software.intel.com/articles/optimization-notice for more information regarding performance & optimization choices in Intel software products.

ACCELERATE YOUR ANALYTICS & AI JOURNEY

[WWW.INTEL.AI](http://www.intel.ai)

THANK YOU

BACKUP

INTEL® XEON® SCALABLE PROCESSORS

THE ONLY DATA CENTER CPU OPTIMIZED FOR AI

INTEL ADVANCED VECTOR EXTENSIONS 512
INTEL DEEP LEARNING BOOST (INTEL DL BOOST)
INTEL OPTANE DC PERSISTENT MEMORY

2019	2020	2021
CASCADE LAKE 14NM NEW AI ACCELERATION (VNNI) NEW MEMORY STORAGE HIERARCHY	**COOPER LAKE** 14NM NEXT GEN INTEL DL BOOST (BFLOAT16) **ICE LAKE** 10NM SHIPPING 1H'20, SAMPLES SHIPPING NOW	**SAPPHIRE RAPIDS** NEXT-GENERATION TECHNOLOGIES

LEADERSHIP PERFORMANCE

INTEL FPGA FOR AI

intel ARRIA 10 inside
intel STRATIX inside
intel Falcon Mesa inside

FIRST TO MARKET TO ACCELERATE EVOLVING AI WORKLOADS

- Precision
- Latency
- Sparsity
- Adversarial Networks
- Reinforcement Learning
- Neuromorphic Computing
- ...

DEPLOYING AI+ FOR FLEXIBLE SYSTEM-LEVEL FUNCTIONALITY

- AI+ I/O Ingest
- AI+ Networking
- AI+ Security
- AI+ Pre/Post Processing
- ...

REAL-TIME WORKLOADS

- Recurrent Neural Networks (RNN)
- Long-short Term Memory (LSTM)
- Speech Workload

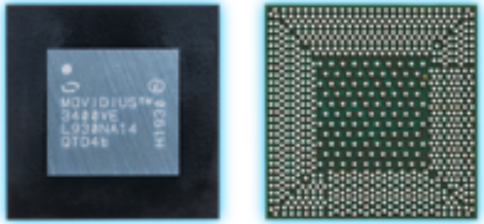
ENABLING REAL-TIME AI IN A WIDE RANGE OF EMBEDDED, EDGE, AND CLOUD APPS

NEXT-GEN MOVIDIUS VPU (KEEM BAY)

LAUNCHING 1H'20

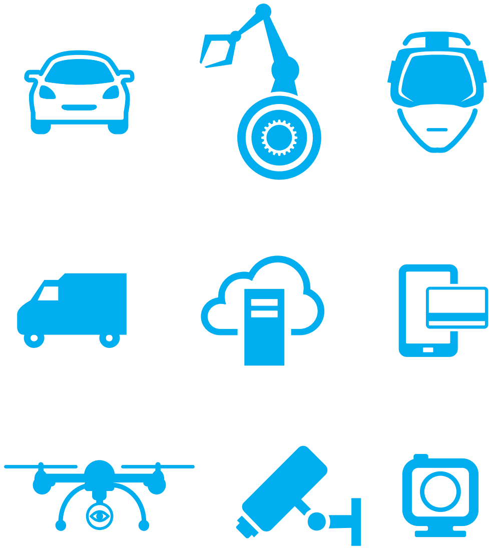
BUILT FOR EDGE AI

- Deep learning inference + computer vision + media
- Faster memory bandwidth
- Groundbreaking high-efficiency architecture
- Accelerated with [redacted]



FLEXIBLE FORM FACTORS

EDGE EXPERIENCES



KEEM BAY IS BUILT FOR EDGE AI…

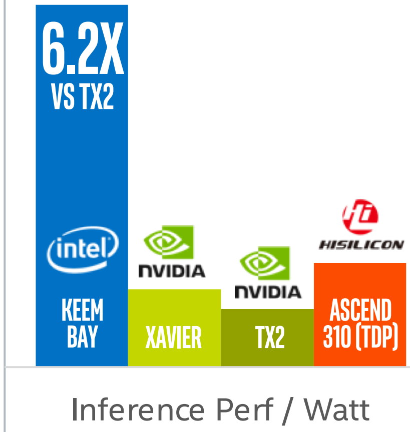
FAST +

4X NVIDIA TX2

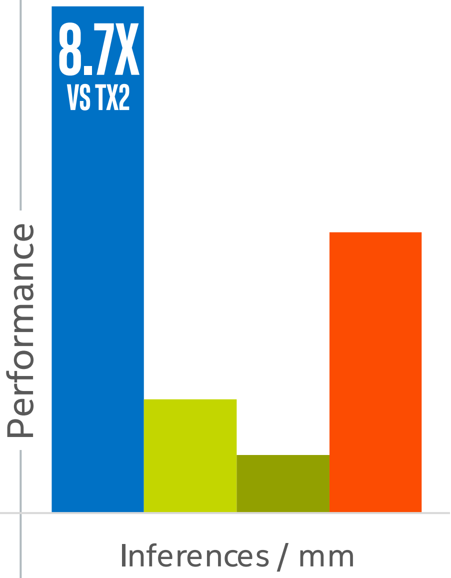
1.25X ASCEND 310

VS. NVIDIA XAVIER **ON PAR**[1] @**1/5** TH POWER

GREEN



SMALL



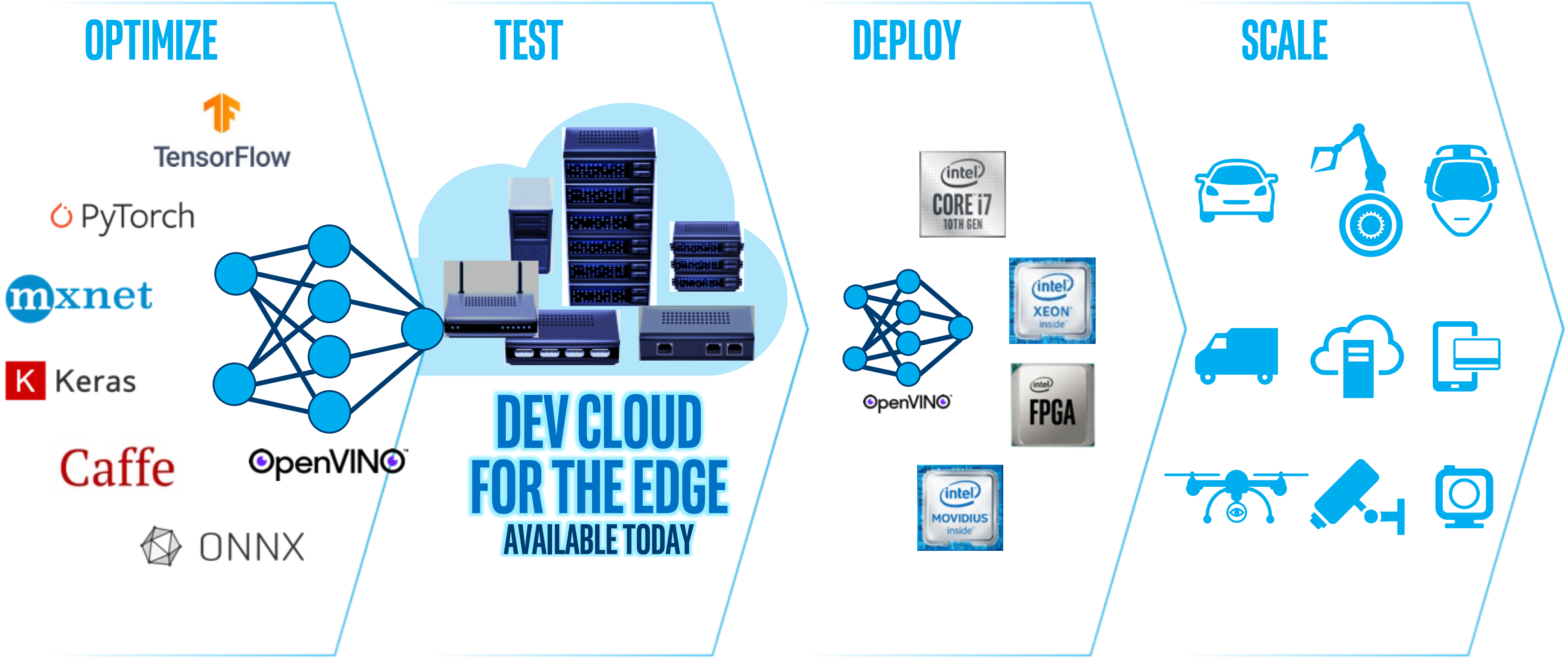
EFFICIENT

4X

INFERENCES / SEC / TOPS VS NVIDIA XAVIER

The above is preliminary performance data based on pre-production components. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. See backup for configuration details. Comparison of Frames Per Second utilizing Resnet-50, Batch 1.
1. Keem Bay throughput within 10% vs Xavier throughput.

OpenVINO™ AI INFERENCE SOFTWARE WORKFLOW



DEEP LEARNING FRAMEWORK (OPTIMIZATIONS BY INTEL)

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

TensorFlow
mxnet
Caffe2
PyTorch
BigDL for Spark

More framework optimizations underway (e.g., PaddlePaddle*, CNTK* and more)

See installation guides at
ai.intel.com/framework-optimizations/

SEE ALSO: Machine Learning Libraries for Python (Scikit-learn, Pandas, NumPy), R (Cart, randomForest, e1071), Distributed (MlLib on Spark, Mahout)
*Limited availability today
Optimization Notice

INTEL DISTRIBUTION FOR PYTHON

python™ software.intel.com/intel-distribution-for-python

FOR DEVELOPERS USING THE MOST POPULAR AND FASTEST-GROWING PROGRAMMING LANGUAGE FOR AI

EASY, OUT-OF-THE-BOX ACCESS TO HIGH-PERFORMANCE PYTHON

- Prebuilt, optimized for numerical computing, data analytics, HPC
- Drop-in replacement for your existing Python (no code changes required)

DRIVE PERFORMANCE WITH MULTIPLE OPTIMIZATION TECHNIQUES

- Accelerated NumPy/SciPy/Scikit-Learn with Intel Math Kernel Library (Intel MKL)
- Data analytics with pyDAAL, enhanced thread scheduling with TBB, Jupyter Notebook interface, Numba, Cython
- Scale easily with optimized MPI4Py and Jupyter notebooks

FASTER ACCESS TO LATEST OPTIMIZATIONS FOR INTEL ARCHITECTURE

- Distribution and individual optimized packages available through conda and Anaconda Cloud
- Optimizations upstreamed back to main Python trunk

ADVANCING PYTHON PERFORMANCE CLOSER TO NATIVE SPEEDS

INTEL DATA ANALYTICS ACCELERATION LIBRARY (INTEL DAAL)

BUILDING BLOCKS FOR ALL DATA ANALYTICS STAGES, INCLUDING DATA PREPARATION, DATA MINING & MACHINE LEARNING

Pre-processing → Transformation → Analysis → Modeling → Validation → Decision Making

Open Source | Apache 2.0 License

Common Python, Java and C++ APIs across all Intel hardware

Optimized for large data sets including streaming and distributed processing

Flexible interfaces to leading big data platforms including Spark and range of data formats (CSV, SQL, etc.)

HIGH-PERFORMANCE MACHINE LEARNING AND DATA ANALYTICS LIBRARY

INTEL DISTRIBUTION OF OPENVINO TOOLKIT

OpenVINO™

DEEP LEARNING

Caffe · TensorFlow · ONNX · mxnet · KALDI

- Model optimizer
- Inference engine
- Supports 300+ public models, incl. 40+ pretrained models

COMPUTER VISION

OpenCV · OpenVX · OpenCL™

- Computer vision library (kernel & graphic APIs)
- Optimized media encode/decode functions

SUPPORTS MAJOR AI FRAMEWORKS

Rapid adoption by developers

CROSS-PLATFORM FLEXIBILITY

Multiple products launched based on this toolkit

HIGH PERFORMANCE, HIGH EFFICIENCY

intel ATOM x7 inside · intel CORE i7 inside · intel XEON inside · intel MOVIDIUS inside · intel ARRIA inside

Breadth of product portfolio

STRONG ADOPTION + RAPIDLY EXPANDING CAPABILITY

SOFTWARE.INTEL.COM/OPENVINO-TOOLKIT

Optimization Notice

Obtain open source version at 01.org/openvinotoolkit

CSP IAAS OFFERINGS – OVERVIEW

	AWS		Azure		GCP
Name	**DL AMI**		**Data Science VMs**	**Cycle* Cloud**	**Google* Compute Engine**
Instance	C5	C5	Fv2 or HC Series	HC Series	Platform based on Skylake
Description	Pre-installed pip packages	Customer-built DL engine – clean slate	Azure VM images, pre-installed, configured and tested with several popular AI/DL tools	Easy-to-set-up clusters with Singularity containers	Scalable, high-performance virtual machines
HW SKUs	Intel Xeon Platinum 8000 series (code-named Skylake)		Various HW Platforms	Any HW platforms (validated on Skylake)	Intel Xeon Platinum family (Skylake)
Optimized FW	**TensorFlow, MxNet, and PyTorch**		**TensorFlow and VM templates on MarketPlace**	**TensorFlow**	**TensorFlow**
Instance Size	2vCPU to 72vCPU		Fsv2-Series 2 to 72 vCPU	Any Instance size	Up to 160 vCPU
Memory	144 GiB		Up to 144 GiB		Up to 3.75 TB
Use Case	**Advanced compute intensive workloads: high performance web servers, HPC, batch processing, ad serving, gaming, distributed analytics and ML/DL inference**		**Batch processing, web servers, analytics and gaming**	**HPC workloads but can run deep learning**	**Improve and manage patient data, create intuitive customer experience**
CSP Value Prop	Best price performance		Lower per-hour list price is best value in price-performance in Azure portfolio Easily transition from on-prem to cloud, compliance and global reach	Dynamically provision HPC Azure clusters and orchestrate data and jobs for hybrid and cloud workflows	Industry-leading price and performance

CSP PAAS OFFERINGS – OVERVIEW

	AWS	Azure	GCP
Name	**SageMaker**	Azure Machine Learning with Brainwave	Google App Engine
Type	PaaS	PaaS	PaaS
Instance	C5 Instance	Fv2 or HC Series	Flexible Environment
Description	A fully managed platform to easily build, train and deploy machine learning models at any scale	A fullymanaged cloud service to easily build, deploy, and share predictive analytics solutions.	A fully managed serverless platform to build highly scalable applications
OS	N/A	N/A	N/A
HW SKUs	C5 Instance (Skylake)	Intel Arria® 10 FPGA	
FW	Pre-configured DAAL4Py (marketplace)	Marketplace approach for optimized FW WIP	
Use Case	Ad targeting, prediction & forecasting, industrial IoT & Machine Learning		Modern web applications and scalable mobile backends
CSP Value Prop	Ease of use. Pre-configured environment		

Configuration Details for 2nd Gen Intel® Xeon® Processor Slide

2x Average Generational Gains: On 2-socket servers with 2nd Gen Intel® Xeon® Platinum 9200 processor. Geomean of est SPECrate2017_int_base, est SPECrate2017_fp_base, STREAM-Triad, Intel® Distribution of LINPACK, server-side Java*. Platinum 92xx vs. Platinum 8180. Baseline: 1-node, 2x Intel® Xeon® Platinum 8180 processor on Wolf Pass with 384 GB (12 X 32GB 2666) total memory, ucode 0x200004D on RHEL7.6, 3.10.0-957.el7.x86_64, IC19u1, AVX512, HT on all (off Stream, LINPACK), Turbo on all (off Stream, LINPACK), result: est int throughput=307, est fp throughput=251, STREAM-Triad=204, LINPACK=3238, server-side Java=165724, test by Intel on 1/29/2019. New configuration: 1-node, 2x Intel® Xeon® Platinum 9282 processor on Walker Pass with 768 GB (24x 32GB 2933) total memory, ucode 0x400000A on RHEL7.6, 3.10.0-957.el7.x86_64, IC19u1, AVX512, HT on all (off Stream, LINPACK), Turbo on all (off Stream, LINPACK), result: est int throughput=635, est fp throughput=526, STREAM-Triad=407, LINPACK=6411, server-side Java=332913, test by Intel on 2/16/2019.

LINPACK: AMD EPYC 7601: Supermicro AS-2023US-TR4 with 2 AMD EPYC 7601 (2.2GHz, 32 core) processors, SMT OFF, Turbo ON, BIOS ver 1.1a, 4/26/2018, microcode: 0x8001227, 16x32GB DDR4-2666, 1 SSD, Ubuntu 18.04.1 LTS (4.17.0-041700-generic Retpoline), High Performance Linpack v2.2, compiled with Intel(R) Parallel Studio XE 2018 for Linux, Intel MPI version 18.0.0.128, AMD BLIS ver 0.4.0, Benchmark Config: Nb=232, N=168960, P=4, Q=4, Score =1095GFs, tested by Intel as of July 31, 2018. vs. 1-node, 2x Intel® Xeon® Platinum 9282 cpu on Walker Pass with 768 GB (24x 32GB 2933) total memory, ucode 0x400000A on RHEL7.6, 3.10.0-957.el7.x86_65, IC19u1, AVX512, HT off, Turbo on, score=6411, test by Intel on 2/16/2019. 1-node, 2x Intel® Xeon® Platinum 8280M cpu on Wolf Pass with 384 GB (12 X 32GB 2933) total memory, ucode 0x400000A on RHEL7.6, 3.10.0-957.el7.x86_65, IC19u1, AVX512, HT off Linpack, Turbo on, score=3462, test by Intel on 1/30/2019.

Config for – Accelerator Like Performance on Intel Xeon Processors with Intel DL Boost

Nvidia data source: https://Modeler.nvidia.com/deep-learning-performance-training-inference

Max Inference throughput at <7ms

Intel® Xeon® Platinum 8180 processor: Tested by Intel as of 2/26/2019. 2S Intel® Xeon® Platinum 8280(28 cores per socket), HT ON, turbo ON, Total Memory 384 GB (12 slots/ 32 GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0348.011820191451, Centos 7 Kernel 3.10.0-957.5.1.el7.x86_64, Deep Learning Framework: Intel® Optimization for Caffe version: https://github.com/intel/caffe Commit id: 362a3b3, ICC 2019.2.187 for build, MKL DNN version: v0.17 (commit hash: 830a10059a018cd2634d94195140cf2d8790a75a), model: https://github.com/intel/caffe/blob/master/models/intel_optimized_models/int8/resnet50_int8_full_conv.prototxt, BS=10, synthetic Data:3x224x224, 2 instance/2 socket, Datatype: INT8; latency: 6.16 ms

Intel® Xeon® Platinum 9242 Processor: Tested by Intel as of 2/26/2019 2S Intel® Xeon® Platinum 9242(48 cores per socket), HT ON, turbo ON, Total Memory 768 GB (24 slots/ 32 GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0403.022020190327, Centos 7 Kernel 3.10.0-957.5.1.el7.x86_64, Deep Learning Framework: Intel® Optimization for Caffe version: https://github.com/intel/caffe Commit id: 362a3b3, ICC 2019.2.187 for build, MKL DNN version: v0.17 (commit hash: 830a10059a018cd2634d94195140cf2d8790a75a), model: https://github.com/intel/caffe/blob/master/models/intel_optimized_models/int8/resnet50_int8_full_conv.prototxt, BS= 2, synthetic Data:3x224x224, 16 instance/2 socket, Datatype: INT8; latency: 6.90 ms

Intel® Xeon® Platinum 9282 Processor: Tested by Intel as of 2/26/2019. DL Inference: Platform: Dragon rock 2S Intel® Xeon® Platinum 9282(56 cores per socket), HT ON, turbo ON, Total Memory 768 GB (24 slots/ 32 GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0241.112020180249, Centos 7 Kernel 3.10.0-957.5.1.el7.x86_64, Deep Learning Framework: Intel® Optimization for Caffe version: https://github.com/intel/caffe Commit id: 362a3b3, ICC 2019.2.187 for build, MKL DNN version: v0.17 (commit hash: 830a10059a018cd2634d94195140cf2d8790a75a), model: https://github.com/intel/caffe/blob/master/models/intel_optimized_models/int8/resnet50_int8_full_conv.prototxt, BS=10, synthetic Data:3x224x224, 4 instance/2 socket, Datatype: INT8; latency: 6.91 ms

Max Inference throughput

Intel® Xeon® Platinum 8180 processor: Tested by Intel as of 2/26/2019. 2S Intel® Xeon® Platinum 8280(28 cores per socket), HT ON, turbo ON, Total Memory 384 GB (12 slots/ 32 GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0348.011820191451, Centos 7 Kernel 3.10.0-957.5.1.el7.x86_64, Deep Learning Framework: Intel® Optimization for Caffe version: https://github.com/intel/caffe Commit id: 362a3b3, ICC 2019.2.187 for build, MKL DNN version: v0.17 (commit hash: 830a10059a018cd2634d94195140cf2d8790a75a), model: https://github.com/intel/caffe/blob/master/models/intel_optimized_models/int8/resnet50_int8_full_conv.prototxt, BS=8, syntheticData:3x224x224, 14 instance/2 socket, Datatype: INT8

Intel® Xeon® Platinum 9242 Processor: Tested by Intel as of 2/26/2019 2S Intel® Xeon® Platinum 9242(48 cores per socket), HT ON, turbo ON, Total Memory 768 GB (24 slots/ 32 GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0403.022020190327, Centos 7 Kernel 3.10.0-957.5.1.el7.x86_64, Deep Learning Framework: Intel® Optimization for Caffe version: https://github.com/intel/caffe Commit id: 362a3b3, ICC 2019.2.187 for build, MKL DNN version: v0.17 (commit hash: 830a10059a018cd2634d94195140cf2d8790a75a), model: https://github.com/intel/caffe/blob/master/models/intel_optimized_models/int8/resnet50_int8_full_conv.prototxt, BS=128, synthetic Data:3x224x224, 4 instance/2 socket, Datatype: INT8

Intel® Xeon® Platinum 9282 Processor: Tested by Intel as of 2/26/2019. DL Inference: Platform: Dragon rock 2S Intel® Xeon® Platinum 9282(56 cores per socket), HT ON, turbo ON, Total Memory 768 GB (24 slots/ 32 GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0241.112020180249, Centos 7 Kernel 3.10.0-957.5.1.el7.x86_64, Deep Learning Framework: Intel® Optimization for Caffe version: https://github.com/intel/caffe Commit id: 362a3b3, ICC 2019.2.187 for build, MKL DNN version: v0.17 (commit hash: 830a10059a018cd2634d94195140cf2d8790a75a), model: https://github.com/intel/caffe/blob/master/models/intel_optimized_models/int8/resnet50_int8_full_conv.prototxt, BS=8, synthetic Data:3x224x224, 14 instance/2 socket, Datatype: INT8

BKMs for running multi-stream configurations on Xeon: https://www.intel.ai/wp-content/uploads/sites/69/TensorFlow_Best_Practices_Intel_Xeon_AI-HPC_v1.1_Q119.pdf

Configuration Details (Cont'd)

Configuration: AI Performance – Software + Hardware

INFERENCE using FP32 Batch Size Caffe GoogleNet v1 128 AlexNet 256.

The benchmark results may need to be revised as additional testing is conducted. The results depend on the specific platform configurations and workloads utilized in the testing, and may not be applicable to any particular user's components, computer system or workloads. The results are not necessarily representative of other benchmarks and other benchmark results may show greater or lesser impact from mitigations. 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.

Configurations for Inference throughput

Platform :2 socket Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz / 28 cores HT ON , Turbo ON Total Memory 376.28GB (12slots / 32 GB / 2666 MHz),4 instances of the framework, CentOS Linux-7.3.1611-Core , SSD sda RS3WC080 HDD 744.1GB,sdb RS3WC080 HDD 1.5TB,sdc RS3WC080 HDD 5.5TB , Deep Learning Framework caffe version: a3d5b022fe026e9092fc7abc7654b1162ab9940d Topology:GoogleNet v1 BIOS:SE5C620.86B.00.01.0004.071220170215 MKLDNN: version: 464c268e544bae26f9b85a2acb9122c766a4c396 NoDataLayer. Measured: 1449.9 imgs/sec vs Platform: 2S Intel® Xeon® CPU E5-2699 v3 @ 2.30GHz (18 cores), HT enabled, turbo disabled, scaling governor set to "performance" via intel_pstate driver, 64GB DDR4-2133 ECC RAM. BIOS: SE5C610.86B.01.01.0024.021320191901, CentOS Linux-7.5.1804(Core) kernel 3.10.0-862.3.2.el7.x86_64, SSD sdb INTEL SSDSC2BW24 SSD 223.6GB. Framework BVLC-Caffe: https://github.com/BVLC/caffe, Inference & 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. BVLC Caffe (http://github.com/BVLC/caffe), revision 2a1c552b66f026c7508d390b526f2495ed3be594

Configuration for training throughput:

Platform :2 socket Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz / 28 cores HT ON , Turbo ON Total Memory 376.28GB (12slots / 32 GB / 2666 MHz),4 instances of the framework, CentOS Linux-7.3.1611-Core , SSD sda RS3WC080 HDD 744.1GB,sdb RS3WC080 HDD 1.5TB,sdc RS3WC080 HDD 5.5TB , Deep Learning Framework caffe version: a3d5b022fe026e9092fc7abc765b1162ab9940d Topology:alexnet BIOS:SE5C620.86B.00.01.0004.071220170215 MKLDNN: version: 464c268e544bae26f9b85a2acb9122c766a4c396 NoDataLayer. Measured: 1257 imgs/sec vs Platform: 2S Intel® Xeon® CPU E5-2699 v3 @ 2.30GHz (18 cores), HT enabled, turbo disabled, scaling governor set to "performance" via intel_pstate driver, 64GB DDR4-2133 ECC RAM. BIOS: SE5C610.86B.01.01.0024.021320191901, CentOS Linux-7.5.1804(Core) kernel 3.10.0-862.3.2.el7.x86_64, SSD sdb INTEL SSDSC2BW24 SSD 223.6GB. Framework BVLC-Caffe: https://github.com/BVLC/caffe, Inference & 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. BVLC Caffe (http://github.com/BVLC/caffe), revision 2a1c552b66f026c7508d390b526f2495ed3be594

CONFIGURATION DETAILS (CONT'D)

Configuration: AI Performance – Software + Hardware

1.4x training throughput improvement in August 2019:

Tested by Intel as of measured August 2nd 2019. Processor: 2 socket Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz / 28 cores HT ON , Turbo ON Total Memory 376.46GB (12slots / 32 GB / 2666 MHz). CentOS Linux-7.3.1611-Core kernel 3.10.0-693.11.6.el7.x86_64, SSD sda RS3WC080 HDD 744.1GB,sdb RS3WC080 HDD 1.5TB,sdc RS3WC080 HDD 5.5TB , Deep Learning Framework Intel® Optimizations for caffe version:a3d5b022fe026e9092fc7abc7654b1162ab9940d Topology::resnet_50 BIOS:SE5C620.86B.00.01.0013.030920190427 MKLDNN: version: 464c268e544bae26f9b85a2acb9122c766a4c396 NoDataLayer. Measured: 123 imgs/sec vs Intel tested July 11th 2017 Platform: 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. 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 2019.0.20170425. Caffe run with "numactl -l".

5.4x inference throughput improvement in August 2019:

Tested by Intel as of measured July 26th 2019 :2 socket Intel(R) Xeon(R) Platinum 8180 CPU @ 2.50GHz / 28 cores HT ON , Turbo ON Total Memory 376.46GB (12slots / 32 GB / 2666 MHz). CentOS Linux-7.3.1611-Core, kernel: 3.10.0-862.3.3.el7.x86_64, SSD sda RS3WC080 HDD 744.1GB,sdb RS3WC080 HDD 1.5TB,sdc RS3WC080 HDD 5.5TB , Deep Learning Framework Intel® Optimized caffe version:a3d5b022fe026e9092fc7abc7654b1162ab9940d Topology::resnet_50_v1 BIOS:SE5C620.86B.00.01.0013.030920190427 MKLDNN: version:464c268e544bae26f9b85a2acb9122c766a4c396 instances: 2 instances socket:2 (Results on Intel® Xeon® Scalable Processor were measured running multiple instances of the framework. Methodology described here: https://software.intel.com/en-us/articles/boosting-deep-learning-training-inference-performance-on-xeon-and-xeon-phi)   NoDataLayer. Datatype: INT8 Batchsize=64 Measured: 1233.39 imgs/sec vs Tested by Intel as of July 11th 2017:2S 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. 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 (ResNet-50). Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2019.0.20170425. Caffe run with "numactl -l".

11X inference thoughput improvement with CascadeLake:

Future Intel Xeon Scalable processor (codename Cascade Lake) 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 vs Tested by Intel as of July 11th 2017: 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. 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 (ResNet-50),. Intel C++ compiler ver. 17.0.2 20170213, Intel MKL small libraries version 2019.0.20170425. Caffe run with "numactl -l".

Configuration Details (Cont'd)

Intel Arria 10 – 1150 FPGA energy efficiency on Caffe/AlexNet up to 25 img/s/w with FP16 at 297MHz

Vanilla AlexNet Classification Implementation as specified by http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf, Training Parameters taken from Caffe open-source Framework are 224x224x3 Input, 1000x1 Output, FP16 with Shared Block-Exponents, All compute layers (incl. Fully Connected) done on the FPGA except for Softmax, Arria 10-1150 FPGA, -1 Speed Grade on Altera PCIe DevKit with x72 DDR4 @ 1333 MHz, Power measured through on-board power monitor (FPGA POWER ONLY), ACDS 16.1 Internal Builds + OpenCL SDK 16.1 Internal Build, Compute machine is an HP Z620 Workstation, Xeon E5-1660 at 3.3 GHz with 32GB RAM. The Xeon is not used for compute.

Config for –Optimized Deep Learning Frameworks and Toolkits

3.0x and 1.87x performance boost with MxNet on ResNet-50: Tested by Intel as of 1/30/2019. 2 socket Intel® Xeon® Platinum 8280 Processor, 28 cores HT On Turbo ON Total Memory 384 GB (12 slots/ 32GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0271.120720180605 (ucode:0x4000013),CentOS 7.6, 4.19.5-1.el7.elrepo.x86_64, Deep Learning Framework: MxNet https://github.com/apache/incubator-mxnet/ -b master da5242b732de39ad47d8ecee582f261ba5935fa9, Compiler: gcc 4.8.5,MKL DNN version: v0.17, ResNet50: https://github.com/apache/incubator-MXNet/blob/master/python/MXNet/gluon/model_zoo/vision/resnet.py, BS=64, synthetic data, 2 instance/2 socket, Datatype: INT8 vs Tested by Intel as of 1/30/2019. 2 socket Intel® Xeon® Platinum 8180 Processor, 28 cores HT On Turbo ON Total Memory 384 GB (12 slots/ 32GB/ 2633 MHz), BIOS: SE5C620.86B.0D.01.0286.121520181757, CentOS 7.6, 4.19.5-1.el7.elrepo.x86_64, Deep Learning Framework: MxNet https://github.com/apache/incubator-mxnet/ -b master da5242b732de39ad47d8ecee582f261ba5935fa9, Compiler: gcc 4.8.5,MKL DNN version: v0.17, ResNet50: https://github.com/apache/incubator-MXNet/blob/master/python/MXNet/gluon/model_zoo/vision/resnet.py, BS=64, synthetic data, 2 instance/2 socket, Datatype: INT8 and FP32

3.7x and 2.1x performance boost with Pytorch ResNet-50: Tested by Intel as of 2/25/2019. 2 socket Intel® Xeon® Platinum 8280 Processor, 28 cores HT On Turbo ON Total Memory 384 GB (12 slots/ 32GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0271.120720180605 (ucode: 0x4000013), Ubuntu 18.04.1 LTS, kernel 4.15.0-45-generic, SSD 1x sda INTEL SSDSC2BA80 SSD 745.2GB, 3X INTEL SSDPE2KX040T7 SSD 3.7TB , Deep Learning Framework: Pytorch with ONNX/Caffe2 backend: https://github.com/pytorch/pytorch.git (commit: 4ac91b2d64eeea5ca21083831db5950dc08441d6)and Pull Request link: https://github.com/pytorch/pytorch/pull/17464 (submitted for upstreaming), gcc (Ubuntu 7.3.0-27ubuntu1~18.04) 7.3.0, MKL DNN version: v0.17.3 (commit hash: 0c3cb94999919d33e4875177fdef662bd9413dd4), ResNet-50: https://github.com/intel/optimized-models/tree/master/pytorch, BS=512, synthetic data, 2 instance/2 socket, Datatype: INT8 vs Tested by Intel as of 2/25/2019. 2 socket Intel® Xeon® Platinum 8180 Processor, 28 cores HT On Turbo ON Total Memory 192 GB (12 slots/ 16GB/ 2666 MHz), BIOS: SE5C620.86B.00.01.0015.110720180833 (ucode: 0x200004d), CentOS 7.5, 3.10.0-693.el7.x86_64, Intel® SSD DC S4500 SERIES SSDSC2KB480G7 2.5'' 6Gb/s SATA SSD 480G, Deep Learning Framework: : https://github.com/pytorch/pytorch.git (commit:4ac91b2d64eeea5ca21083831db5950dc08441d6)and Pull Request link: https://github.com/pytorch/pytorch/pull/17464 (submitted for upstreaming), gcc (Red Hat 5.3.1-6) 5.3.1 20160406, MKL DNN version: v0.17.3 (commit hash: 0c3cb94999919d33e4875177fdef662bd9413dd4), ResNet-50: https://github.com/intel/optimized-models/tree/master/pytorch, BS=512, synthetic data, 2 instance/2 socket, Datatype: INT8&FP32

3.9x and 1.8x performance boost with TensorFlow ResNet-50:Tested by Intel as of 3/1/2019. 2 socket Intel® Xeon® Platinum 8280 Processor, 28 cores HT On Turbo ON Total Memory 384 GB (12 slots/ 32GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0271.120720180605 (ucode:0x4000013),CentOS 7.6, 4.19.5-1.el7.elrepo.x86_64, Deep Learning Framework: https://hub.docker.com/r/intelaipg/intel-optimized-tensorflow:PR25765-devel-mkl (https://github.com/tensorflow/tensorflow.git commit: 6f2eaa3b99c241a9c09c345e1029513bc4cd470a + Pull Request PR 25765, PR submitted for upstreaming) Compiler: gcc 6.3.0,MKL DNN version: v0.17, ResNet50: https://github.com/IntelAI/models/tree/master/models/image_recognition/tensorflow/resnet50, (commit: 87261e70a902513f934413f009364c4f2eed6642) BS=128, synthetic data, 2 instance/2 socket, Datatype: INT8 vs Tested by Intel as of 3/1/2019. 2 socket Intel® Xeon® Platinum 8180 Processor, 28 cores HT On Turbo ON Total Memory 384 GB (12 slots/ 32GB/ 2633 MHz), BIOS: SE5C620.86B.0D.01.0286.121520181757, CentOS 7.6, 4.19.5-1.el7.elrepo.x86_64, Deep Learning Framework: https://hub.docker.com/r/intelaipg/intel-optimized-tensorflow:PR25765-devel-mkl 6f2eaa3b99c241a9c09c345e1029513bc4cd470a + PR25765, PR submitted for upstreaming) Compiler: gcc 6.3.0,MKL DNN version: v0.17, ResNet50: https://github.com/IntelAI/models/tree/master/models/image_recognition/tensorflow/resnet50 , (commit: 87261e70a902513f934413f009364c4f2eed6642) BS=128, synthetic data, 2 instance/2 socket, Datatype: FP32 & INT8

3.9x and 1.9x performance boost with OpenVino™ ResNet-50: Tested by Intel as of 1/30/2019. 2 socket Intel® Xeon® Platinum 8280 Processor, 28 cores HT On Turbo ON Total Memory 384 GB (12 slots/ 32GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0271.120720180605 (ucode:0x4000013), Linux-4.15.0-43-generic-x86_64-with-debian-buster-sid, Compiler: gcc (Ubuntu 7.3.0-27ubuntu1~18.04) 7.3.0, Deep Learning ToolKit: OpenVINO R5 (DLDTK Version:1.0.19154 , AIXPRT CP (Community Preview) benchmark (https://www.principledtechnologies.com/benchmarkxprt/aixprt/) BS=64, Imagenet images, 1 instance/2 socket, Datatype: INT8 vs Tested by Intel as of 1/30/2019. 2 socket Intel® Xeon® Platinum 8180 Processor, 28 cores HT On Turbo ON Total Memory 192 GB (12 slots/ 16GB/ 2633 MHz), BIOS: SE5C620.86B.0D.01.0271.120720180605, Linux-4.15.0-29-generic-x86_64-with-Ubuntu-18.04-bionic, Compiler: gcc (Ubuntu 7.3.0-27ubuntu1~18.04) 7.3.0, Deep Learning ToolKit: OpenVINO R5 (DLDTK Version:1.0.19154), AIXPRT CP (Community Preview) benchmark (https://www.principledtechnologies.com/benchmarkxprt/aixprt/) BS=64, Imagenet images, 1 instance/2 socket, Datatype: INT8 and FP32

4.0x and 2.3x performance boost with Intel® Optimizations for Caffe ResNet-50: Tested by Intel as of 2/20/2019. 2 socket Intel® Xeon® Platinum 8280 Processor, 28 cores HT On Turbo ON Total Memory 384 GB (12 slots/ 32GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0271.120720180605 (ucode: 0x4000013), Ubuntu 18.04.1 LTS, kernel 4.15.0-45-generic, SSD 1x sda INTEL SSDSC2BA80 SSD 745.2GB, 3X INTEL SSDPE2KX040T7 SSD 3.7TB , Deep Learning Framework: Intel® Optimization for Caffe version: 1.1.3 (commit hash: 7010334f159da247db3fe3a9d96a3116ca06b09a) , ICC version 18.0.1, MKL DNN version: v0.17 (commit hash: 830a10059a018cd2634d94195140cf2d8790a75a, model: https://github.com/intel/caffe/blob/master/models/intel_optimized_models/int8/resnet50_int8_full_conv.prototxt, BS=64, syntheticData, 2 instance/2 socket, Datatype: INT8 vs Tested by Intel as of 2/21/2019. 2 socket Intel® Xeon® Platinum 8180 Processor, 28 cores HT On Turbo ON Total Memory 192 GB (12 slots/ 16GB/ 2666 MHz), BIOS: SE5C620.86B.00.01.0015.110720180833 (ucode: 0x200004d), CentOS 7.5, 3.10.0-693.el7.x86_64, Intel® SSD DC S4500 SERIES SSDSC2KB480G7 2.5'' 6Gb/s SATA SSD 480G, , Deep Learning Framework: Intel® Optimization for Caffe version: 1.1.3 (commit hash: 7010334f159da247db3fe3a9d96a3116ca06b09a) , ICC version 18.0.1, MKL DNN version: v0.17 (commit hash: 830a10059a018cd2634d94195140cf2d8790a75a, model: https://github.com/intel/caffe/blob/master/models/intel_optimized_models/benchmark/resnet_50/deploy.prototxt, BS=64, synthetic Data, 2 instance/2 socket, Datatype: INT8 and FP32

Disclaimer

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 information go to **www.intel.com/benchmarks**. Performance results are based on testing as of Oct 31, 2019 and may not reflect all publicly available security updates. See configuration disclosure for details. No product or component can be absolutely secure.

Product	**Intel Keem Bay VPU**	**NVIDIA Jetson TX2**	**Huawei Atlas 200 (Ascend 310)**	**NVIDIA Xavier AGX**
Testing as of	10/31/2019	10/30/19	8/25/19	10/22/19
Precision	INT8	FP16	INT8	INT8
Batch Size	1	1	1	1
Sparsity	50% weight sparsity	N/A	N/A	N/A
Product Type	Keem Bay EA CRB Dev kit (preproduction)	Jetson Developer kit	Atlas 200 Developer kit	Jetson Developer kit
Mode	N/A	nvpmodel 0 Fixed Freq	N/A	nvpmodel 0 Fixed Freq
Memory	4GB	8GB	8GB	16GB
Processor	ARM* A53 x 4	ARM*v8 Processor rev 3 (v8l) × 4	ARM* A53 x 8	ARM*v8 Processor rev 0 (v8l) × 2
Graphics	N/A	NVIDIA Tegra X2 (nvgpu)/integrated	N/A	NVIDIA Tegra Xavier (nvgpu)/integrated
OS	Ubuntu 18.04 Kernel 1.18 (64-bit) on Host Yocto Linux 5.3.0 RC8 on KMB	Ubuntu 18.04 LTS (64-bit)	Ubuntu 16.04	Ubuntu 18.04 LTS (64-bit)
Hard Disk	N/A	32GB	32GB	32GB
Software	Performance demo firmware	JetPack: 4.2.2	MindSpore Studio, DDK B883	JetPack: 4.2.1
Listed TDP	N/A	10W	20W	30W

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at www.intel.com.