intel Software

GridKa School 2019

KIT, Campus North, FTU

INTEL'S HARDWARE & SOFTWARE SOLUTIONS - DIRECTIONS FOR ARTIFICIAL INTELLIGENCE

Edmund Preiss

Business Development Manager, EMEA
Intel Computing Performance and Software Products (CPDP)

Agenda

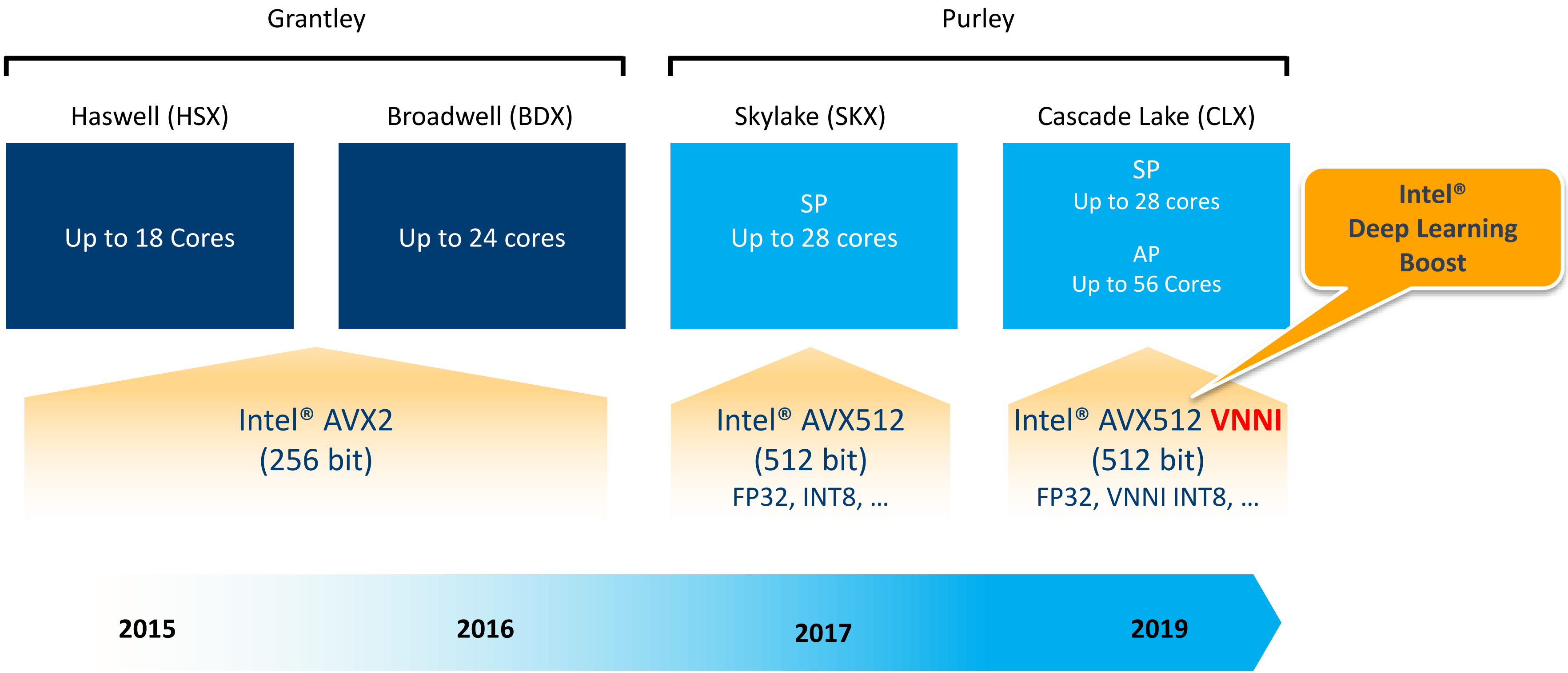
Intel Xeon Family – Latest Status

Intel Software Development Tools

- Intel Parallel Studio XE (IPS XE) Tool Suites
- News on the IPS XE 2020 Edition
- Upcoming **oneAPI** development tools concept

Intel optimized AI Solutions and Directions

EVOLUTION OF INTEL® XEON® PLATFORM



Grantley

Purley

Haswell (HSX)

Up to 18 Cores

Broadwell (BDX)

Up to 24 cores

Skylake (SKX)

SP
Up to 28 cores

Cascade Lake (CLX)

SP
Up to 28 cores

AP
Up to 56 Cores

Intel®
Deep Learning
Boost

Intel® AVX2
(256 bit)

Intel® AVX512
(512 bit)
FP32, INT8, …

Intel® AVX512 **VNNI**
(512 bit)
FP32, VNNI INT8, …

2015

2016

2017

2019

Optimization Notice

Agenda

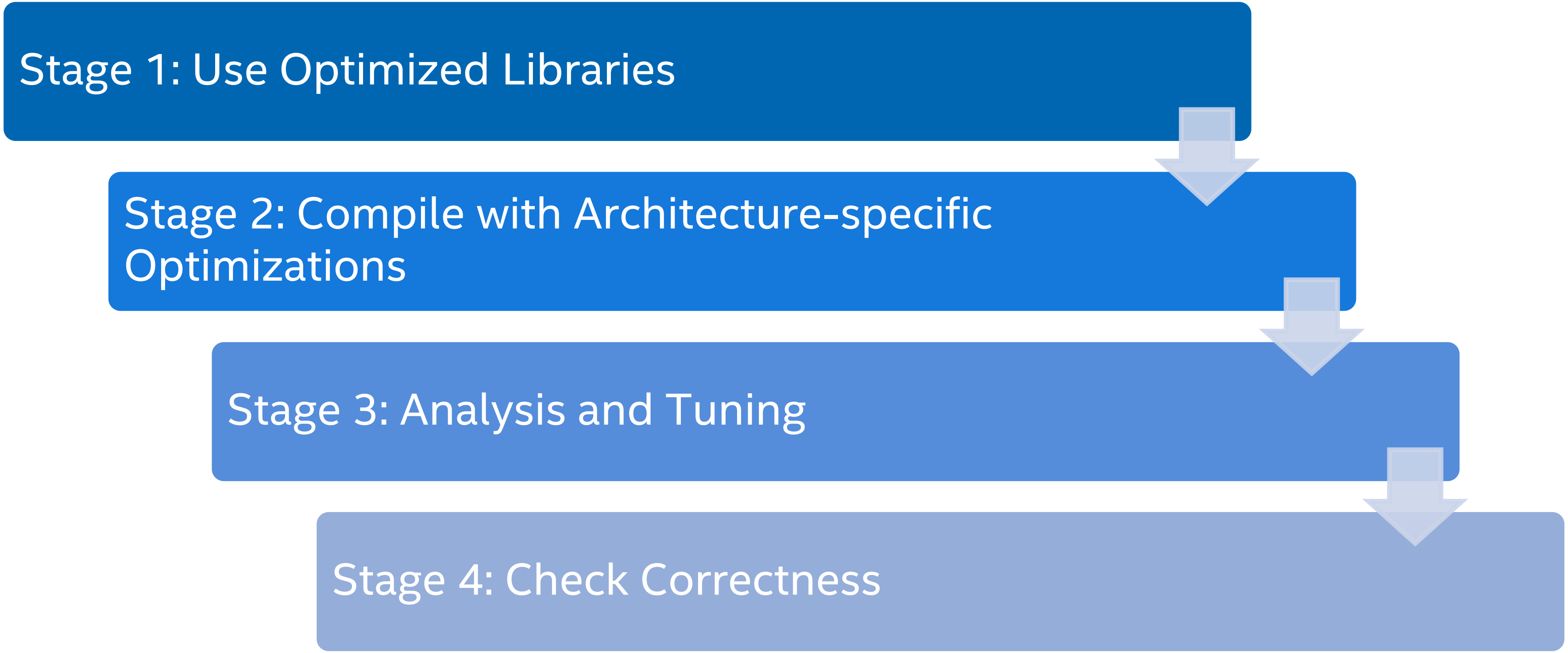
Intel Xeon Family Update

Intel Software Development Tools

- Intel Parallel Studio XE (IPS XE) Tool Suites
- News on the IPS XE 2020 Edition
- Upcoming **oneAPI** development tools concept

Intel optimized AI Solutions and Directions

Code Modernization



Stage 1: Use Optimized Libraries

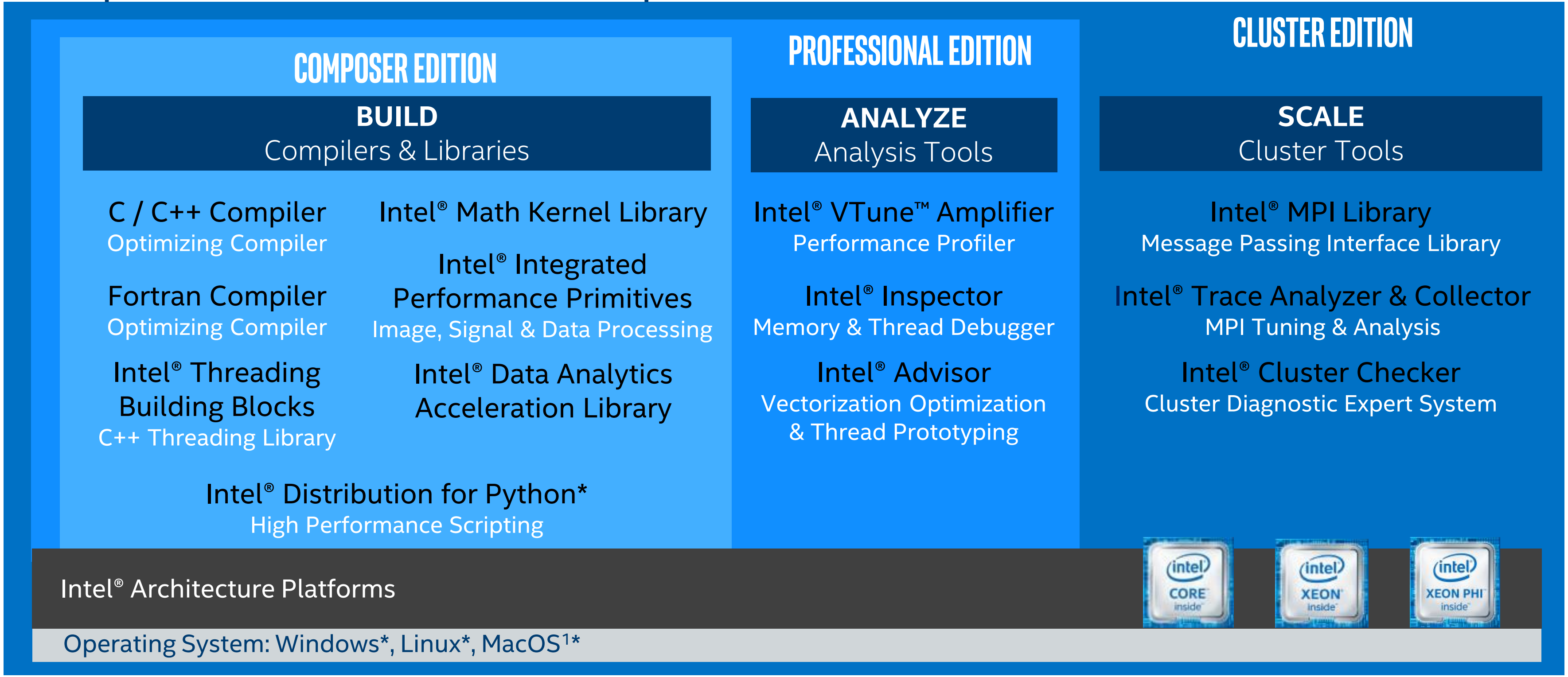
Stage 2: Compile with Architecture-specific Optimizations

Stage 3: Analysis and Tuning

Stage 4: Check Correctness

What's Inside Intel® Parallel Studio XE

Comprehensive Software Development Tool Suite



CLUSTER EDITION

PROFESSIONAL EDITION

COMPOSER EDITION

BUILD
Compilers & Libraries

C / C++ Compiler
Optimizing Compiler

Fortran Compiler
Optimizing Compiler

Intel® Threading Building Blocks
C++ Threading Library

Intel® Math Kernel Library

Intel® Integrated Performance Primitives
Image, Signal & Data Processing

Intel® Data Analytics Acceleration Library

Intel® Distribution for Python*
High Performance Scripting

ANALYZE
Analysis Tools

Intel® VTune™ Amplifier
Performance Profiler

Intel® Inspector
Memory & Thread Debugger

Intel® Advisor
Vectorization Optimization & Thread Prototyping

SCALE
Cluster Tools

Intel® MPI Library
Message Passing Interface Library

Intel® Trace Analyzer & Collector
MPI Tuning & Analysis

Intel® Cluster Checker
Cluster Diagnostic Expert System

Intel® Architecture Platforms

Operating System: Windows*, Linux*, MacOS1*

More Power for Your Code - software.intel.com/intel-parallel-studio-xe

SELECTED INTEL PARALLEL STUDIO XE HIGHLIGHTS

What's Inside Intel® Math Kernel Library

LINEAR ALGEBRA

- BLAS
- LAPACK
- ScaLAPACK
- Sparse BLAS
- Iterative sparse solvers
- PARDISO*

FFTS

- Multidimensional
- FFTW interfaces

VECTOR RNGS

- Congruential
- Wichmann-Hill
- Mersenne Twister
- Sobol
- Neirderreiter
- Non-deterministic

SUMMARY STATISTICS

- Kurtosis
- Variation coefficient
- Order statistics
- Min/max
- Variance-covariance

VECTOR MATH

- Trigonometric
- Hyperbolic
- Exponential
- Log
- Power
- Root

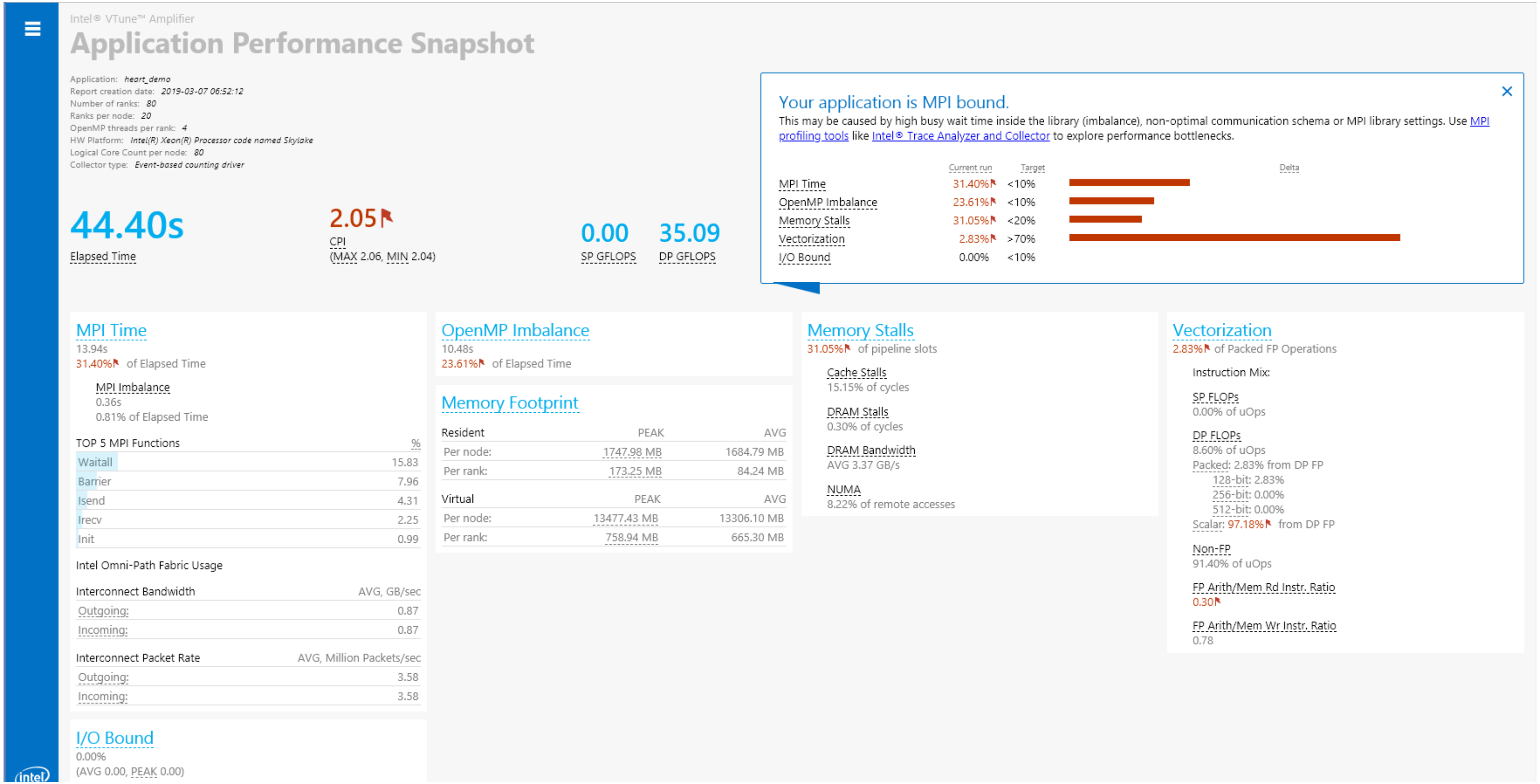
& MORE

- Splines
- Interpolation
- Trust Region
- Fast Poisson Solver

[1] Available only in Intel® Parallel Studio Composer Edition.

Optimization Notice

Application Performance Snapshot (VTune Amplifier)



Intel® VTune™ Amplifier

Application Performance Snapshot

Application: *heart_demo*
Report creation date: *2019-03-07 06:52:12*
Number of ranks: *80*
Ranks per node: *20*
OpenMP threads per rank: *4*
HW Platform: *Intel(R) Xeon(R) Processor code named Skylake*
Logical Core Count per node: *80*
Collector type: *Event-based counting driver*

44.40s Elapsed Time

2.05 ⚑ CPI (MAX 2.06, MIN 2.04)

0.00 SP GFLOPS

35.09 DP GFLOPS

Your application is MPI bound.

This may be caused by high busy wait time inside the library (imbalance), non-optimal communication schema or MPI library settings. Use MPI profiling tools like Intel® Trace Analyzer and Collector to explore performance bottlenecks.

	Current run	Target	Delta
MPI Time	31.40% ⚑	<10%	
OpenMP Imbalance	23.61% ⚑	<10%	
Memory Stalls	31.05% ⚑	<20%	
Vectorization	2.83% ⚑	>70%	
I/O Bound	0.00%	<10%	

MPI Time

13.94s
31.40% ⚑ of Elapsed Time

MPI Imbalance
0.36s
0.81% of Elapsed Time

TOP 5 MPI Functions	%
Waitall	15.83
Barrier	7.96
Isend	4.31
Irecv	2.25
Init	0.99

Intel Omni-Path Fabric Usage

Interconnect Bandwidth	AVG, GB/sec
Outgoing:	0.87
Incoming:	0.87

Interconnect Packet Rate	AVG, Million Packets/sec
Outgoing:	3.58
Incoming:	3.58

I/O Bound

0.00%
(AVG 0.00, PEAK 0.00)

OpenMP Imbalance

10.48s
23.61% ⚑ of Elapsed Time

Memory Footprint

Resident	PEAK	AVG
Per node:	1747.98 MB	1684.79 MB
Per rank:	173.25 MB	84.24 MB

Virtual	PEAK	AVG
Per node:	13477.43 MB	13306.10 MB
Per rank:	758.94 MB	665.30 MB

Memory Stalls

31.05% ⚑ of pipeline slots

Cache Stalls
15.15% of cycles

DRAM Stalls
0.30% of cycles

DRAM Bandwidth
AVG 3.37 GB/s

NUMA
8.22% of remote accesses

Vectorization

2.83% ⚑ of Packed FP Operations

Instruction Mix:

SP FLOPs
0.00% of uOps

DP FLOPs
8.60% of uOps
Packed: 2.83% from DP FP
128-bit: 2.83%
256-bit: 0.00%
512-bit: 0.00%
Scalar: 97.18% ⚑ from DP FP

Non-FP
91.40% of uOps

FP Arith/Mem Rd Instr. Ratio
0.30 ⚑

FP Arith/Mem Wr Instr. Ratio
0.78

Find Effective Optimization Strategies

Intel Advisor: Cache-aware roofline analysis

Roofs Show Platform Limits

Memory, cache & compute limits

Dots Are Loops

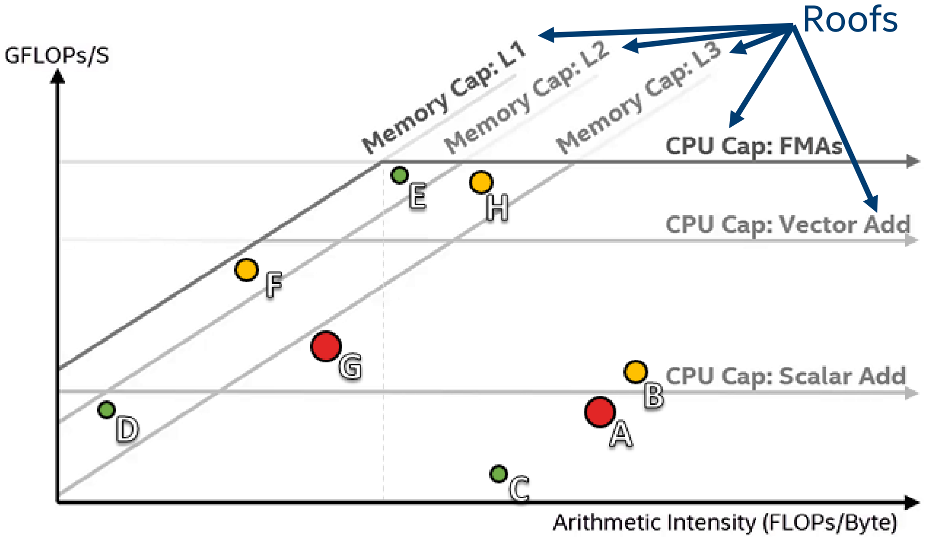
Bigger, red dots take more time so optimization has a bigger impact

Dots farther from a roof have more room for improvement

Higher Dot = Higher GFLOPs/sec

Optimization moves dots up

Algorithmic changes move dots horizontally



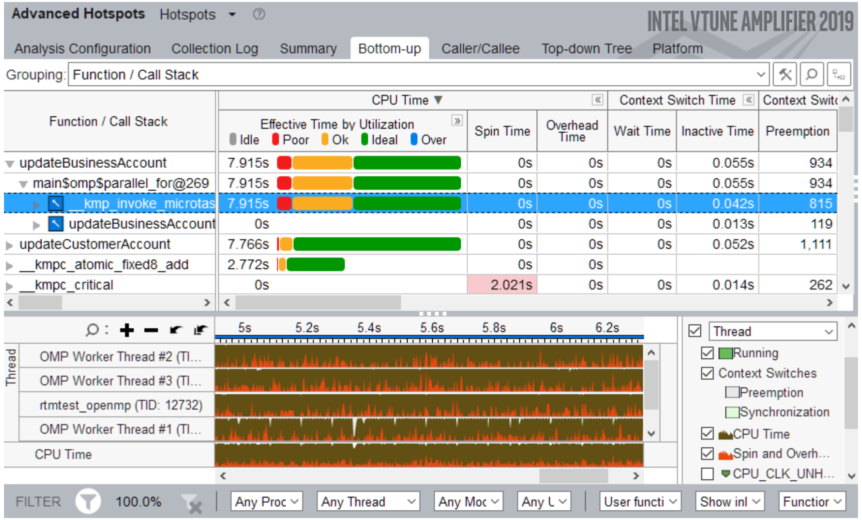
Which loops should we optimize?

- A and G are the best candidates
- B has room to improve, but will have less impact
- E, C, D, and H are poor candidates

Roofline tutorial video

Optimization Notice

Analyze & Tune Application Performance & Scalability with Intel® VTune™ Amplifier—Performance Profiler



Grouping: Function / Call Stack

Function / Call Stack	CPU Time ▼			Context Switch Time		Context Switc
	Effective Time by Utilization Idle Poor Ok Ideal Over	Spin Time	Overhead Time	Wait Time	Inactive Time	Preemption
updateBusinessAccount	7.915s	0s	0s	0s	0.055s	934
main$omp$parallel_for@269	7.915s	0s	0s	0s	0.055s	934
__kmp_invoke_microtas	7.915s	0s	0s	0s	0.042s	815
updateBusinessAccount	0s	0s	0s	0s	0.013s	119
updateCustomerAccount	7.766s	0s	0s	0s	0.052s	1,111
__kmpc_atomic_fixed8_add	2.772s	0s	0s			
__kmpc_critical	0s	2.021s	0s	0s	0.014s	262

Fast, Scalable Code, Faster

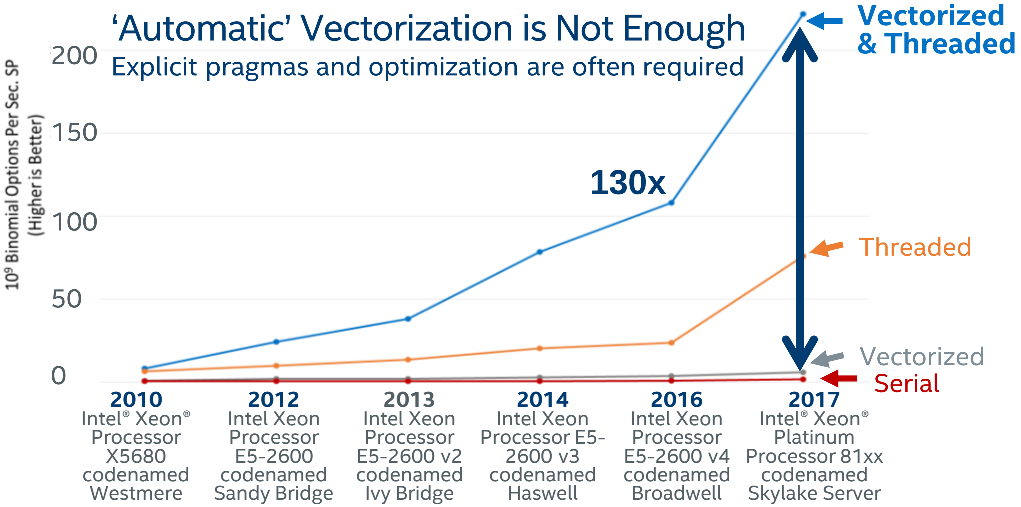
- Accurately profile C, C++, Java*, Python*, Go*, or any mix
- Optimize CPU/GPU, threading, memory, cache, storage & more
- Save time: rich analysis leads to insight

What's New in 2019 Release (Highlights)

- Simplified workflow for easier tuning
- I/O Analysis—Tune SPDK storage & DPDK network performance
- New Platform Profiler—Get insights into platform-level performance, identify memory & storage bottlenecks & imbalances

Optimize Vectorization & Threading with Intel® Advisor

Performance Increases Scale with Each New Hardware Generation



Modern Performant Code

- Vectorized for Intel® Advanced Vector Extensions (Intel® AVX-512 & Intel® AVX)
- Efficient memory access
- Threaded

Capabilities

- Adds & optimizes vectorization
- Analyzes memory patterns
- Quickly prototypes threading

Benchmark: Binomial Options Pricing Model software.intel.com/en-us/articles/binomial-options-pricing-model-code-for-intel-xeon-phi-coprocessor

Performance results are based on testing as of August 2017 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. See Vectorize & Thread or Performance Dies Configurations for 2010-2017 Benchmarks in Backup. Testing by Intel as of August 2017.

Learn More: http: intel.ly/advisor

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, & SSSE3 instruction sets & 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

Get Breakthrough Vectorization Performance

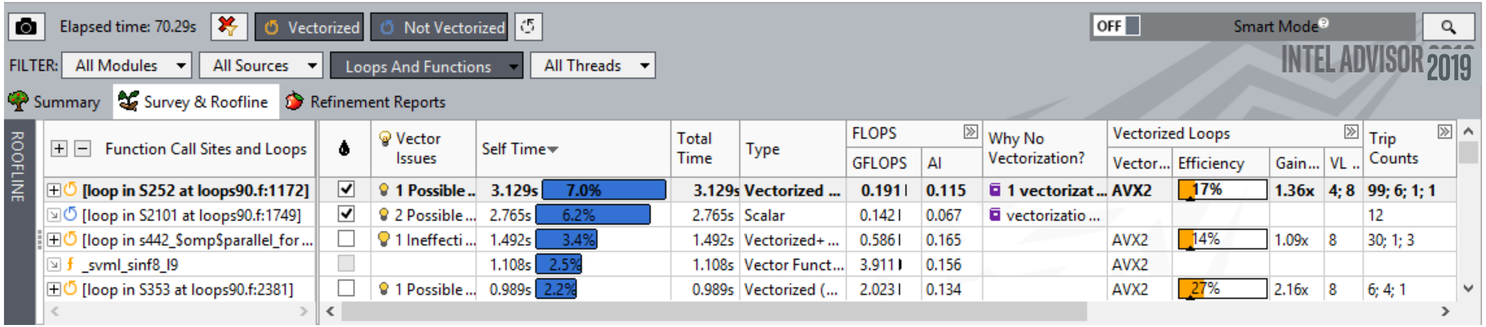
Intel® Advisor—Vectorization Advisor

Faster Vectorization Optimization

- Vectorize where it will pay off most
- Quickly ID what is blocking vectorization
- Tips for effective vectorization
- Safely force compiler vectorization
- Optimize memory stride

Data & Guidance You Need

- Compiler diagnostics + Performance Data + SIMD efficiency
- Detect problems & recommend fixes
- Loop-Carried Dependency Analysis
- Memory Access Patterns Analysis



Elapsed time: 70.29s

FILTER: All Modules | All Sources | Loops And Functions | All Threads

Function Call Sites and Loops	●	Vector Issues	Self Time	Total Time	Type	FLOPS		Why No Vectorization?	Vectorized Loops				Trip Counts
						GFLOPS	AI		Vector...	Efficiency	Gain...	VL ..	
[loop in S252 at loops90.f:1172]	☑	**1 Possible ..**	**3.129s** 7.0%	**3.129s**	**Vectorized ...**	**0.191**	**0.115**	**1 vectorizat ...**	**AVX2**	17%	**1.36x**	**4; 8**	**99; 6; 1; 1**
[loop in S2101 at loops90.f:1749]	☑	2 Possible ...	2.765s 6.2%	2.765s	Scalar	0.142	0.067	vectorizatio ...					12
[loop in s442_ompparallel_for ...	☐	1 Ineffecti ...	1.492s 3.4%	1.492s	Vectorized+ ...	0.586	0.165		AVX2	14%	1.09x	8	30; 1; 3
_svml_sinf8_l9	☐		1.108s 2.5%	1.108s	Vector Funct...	3.911	0.156		AVX2				
[loop in S353 at loops90.f:2381]	☐	1 Possible ...	0.989s 2.2%	0.989s	Vectorized (...	2.023	0.134		AVX2	27%	2.16x	8	6; 4; 1

Optimize for Intel® Advanced Vector Extensions 512 (Intel® AVX-512) with or without access to Intel AVX-512 hardware

Optimization Notice

Agenda

Intel Xeon Family Update

Intel Software Development Tools

- Intel Parallel Studio XE (IPS XE) Tool Suites
- News on the IPS XE 2020 Edition
- Upcoming **oneAPI** development tools concept

Intel optimized AI Solutions and Directions

Key Updates for Intel® Parallel Studio XE 2020

Speed Artificial Intelligence Inferencing - Intel® Compiler and analyzer support for Vector Neural Instructions (VNNI) in Cascade Lake/AP platform

512GB DIMMs with Persistent Memory – Identify, Optimize & Tune Platforms for Intel® Optane™ Persistent Memory with Intel® VTune™ Amplifier

Extended Coarse Grain Profiling – Platform level collection and analysis in Intel® VTune™ Amplifier

Cache Simulation Insights for Vectorization - Roofline analysis for L1, L2 L3, DRAM in Intel® Advisor

Expanded standard support — More Fortran 2018 features & Expanded support of C++17 with initial C++20 support

Latest Processor Support - Intel® Xeon® Scalable Processors (codenamed Cascade Lake / Cascade Lake AP)

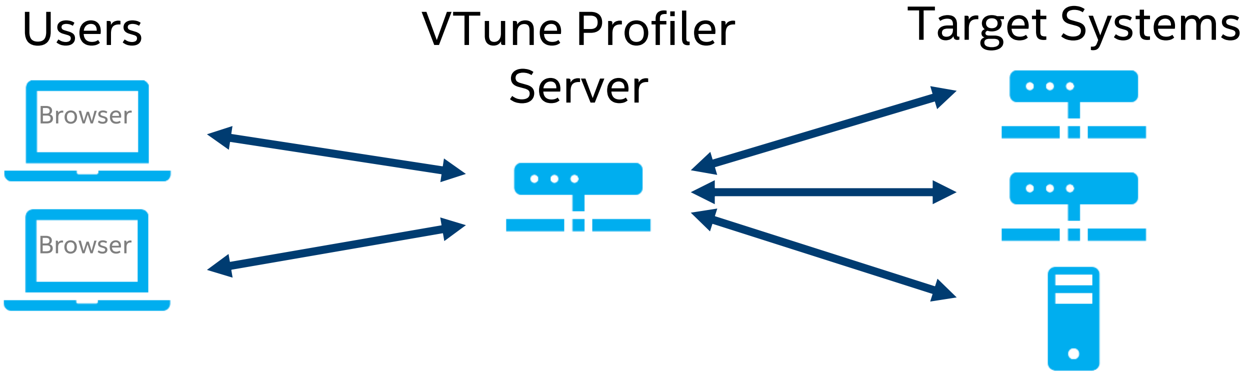
New OS Support – Clear Linux & Amazon Linux 2*

* Supported features of tools and libraries may vary by instances and configurations

Intel® VTune™ Profiler Server Architecture

Just launch your browser and go

2020 Preview Feature



Easier profiling

- **Access with a web browser** – no install required by users
- **Share results** – all results available to all users with server access
- **Profile any system on the network** – server installs collector on the target

Optimization Notice

What’s Coming in Intel® Parallel Studio XE 2020

Coming Q3’2019

- **Intel® Math Kernel Library**
  - Increased AVX512 Optimizations for Complex Vector Math Functions
  - Strided Vector Math API
- **Intel® Data Analytics Acceleration Library**
  - Performance improvements and feature improvements such as
    - Gradient Boosted Trees for large dimensional data sets
    - Extended Z-score support for PCA algorithm
    - XGBoost accelerated with DAAL

Agenda

Intel Xeon Family Update

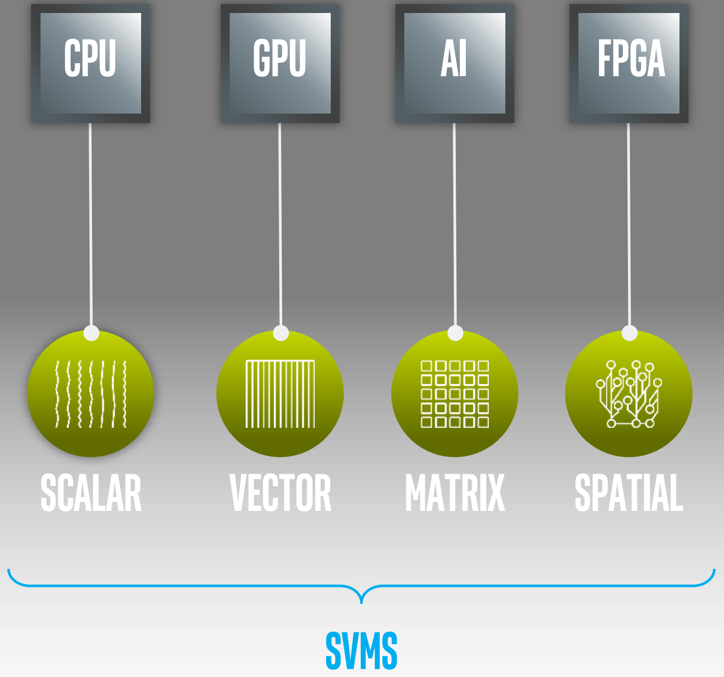
Intel Software Development Tools

- Intel Parallel Studio XE (IPS XE) Tool Suites
- News on the IPS XE 2020 Edition
- Upcoming **oneAPI** development tools concept

Intel optimized AI Solutions and Directions

DIVERSE WORKLOADS REQUIRE DIVERSE ARCHITECTURES

The future is a **diverse** mix of scalar, vector, matrix, and spatial **architectures** deployed in CPU, GPU, AI, FPGA and other accelerators



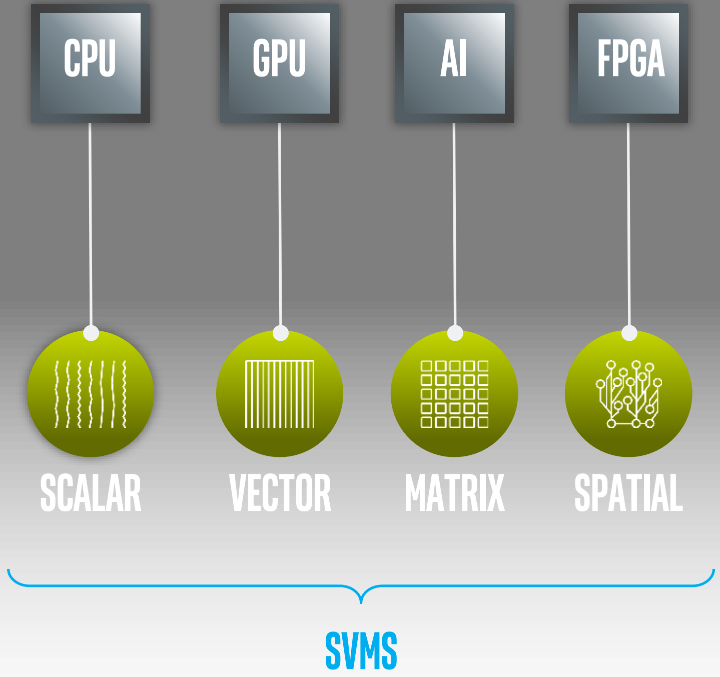
PROGRAMMING CHALLENGE

Diverse set of data-centric hardware

No common programming language or APIs

Inconsistent tool support across platforms

Each platform requires unique software investment



INTEL'S ONE API CORE CONCEPT

Project oneAPI delivers a unified programming model to simplify development across diverse architectures

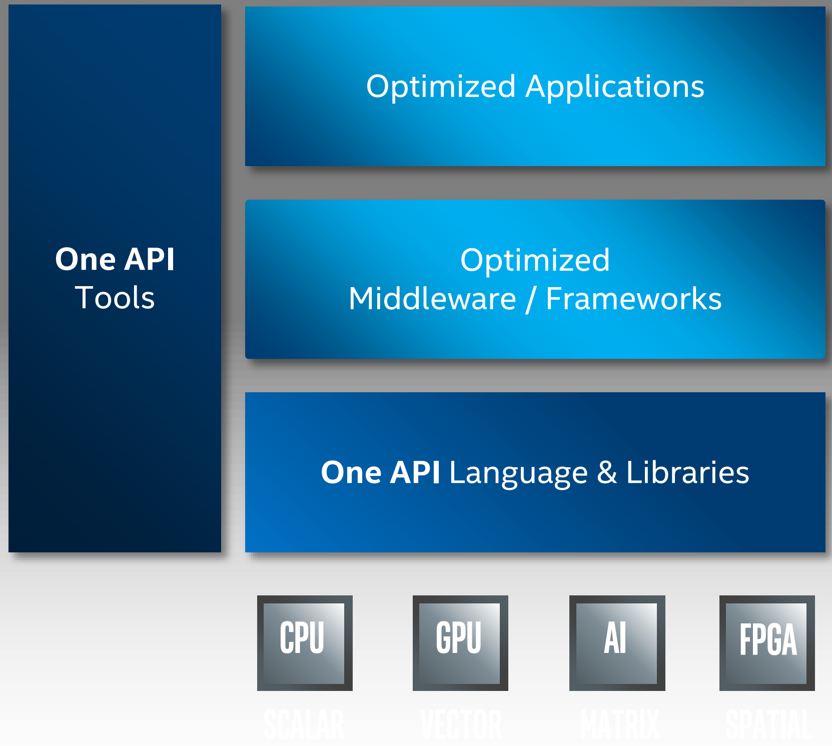
Common developer experience across Scalar, Vector, Matrix and Spatial (SVMS) architecture

Unified and simplified language and libraries for expressing parallelism

Uncompromised native high-level language performance

Support for CPU, GPU, AI and FPGA

Based on industry standards and open specifications



One API Tools

Optimized Applications

Optimized Middleware / Frameworks

One API Language & Libraries

CPU

SCALAR

GPU

VECTOR

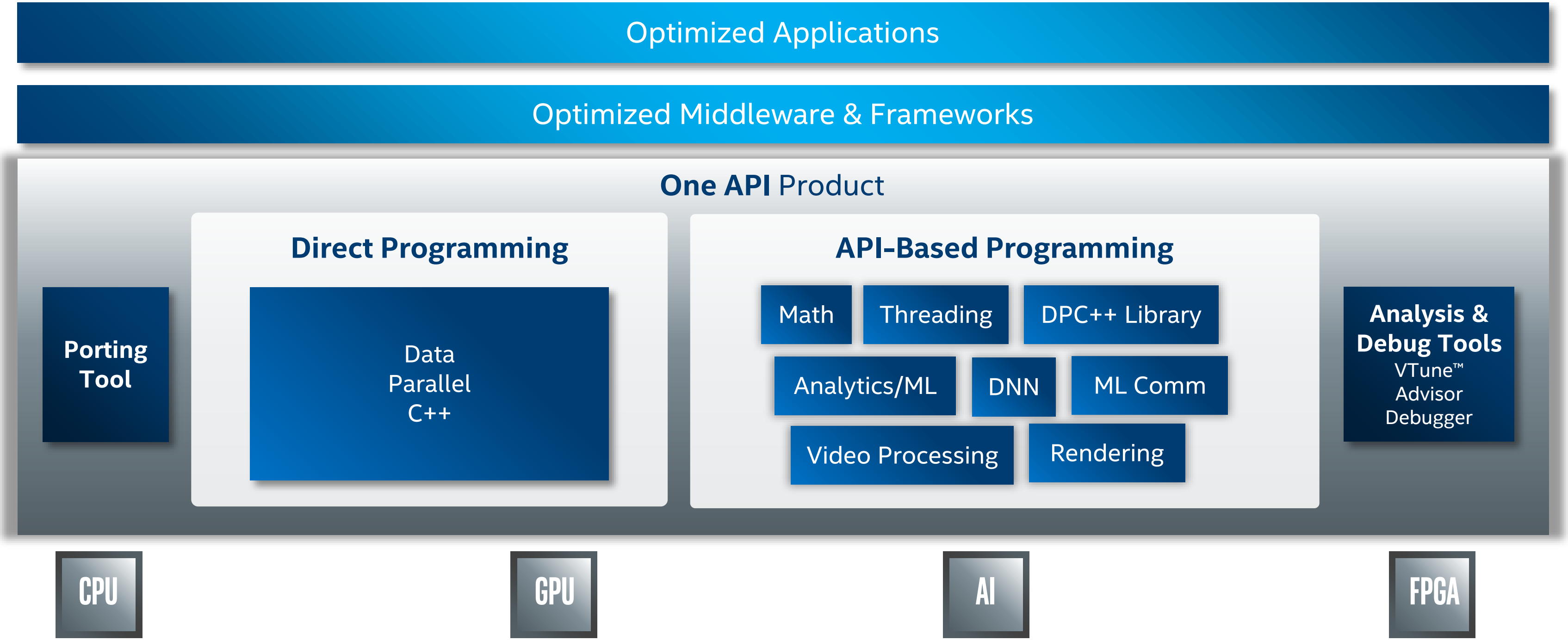
AI

MATRIX

FPGA

SPATIAL

ONE API FOR CROSS-ARCHITECTURE PERFORMANCE



Optimized Applications

Optimized Middleware & Frameworks

One API Product

Porting Tool

Direct Programming

Data Parallel C++

API-Based Programming

Math

Threading

DPC++ Library

Analytics/ML

DNN

ML Comm

Video Processing

Rendering

Analysis & Debug Tools
VTune™
Advisor
Debugger

CPU

GPU

AI

FPGA

Some capabilities may differ per architecture.

Agenda

Intel Xeon Family Update

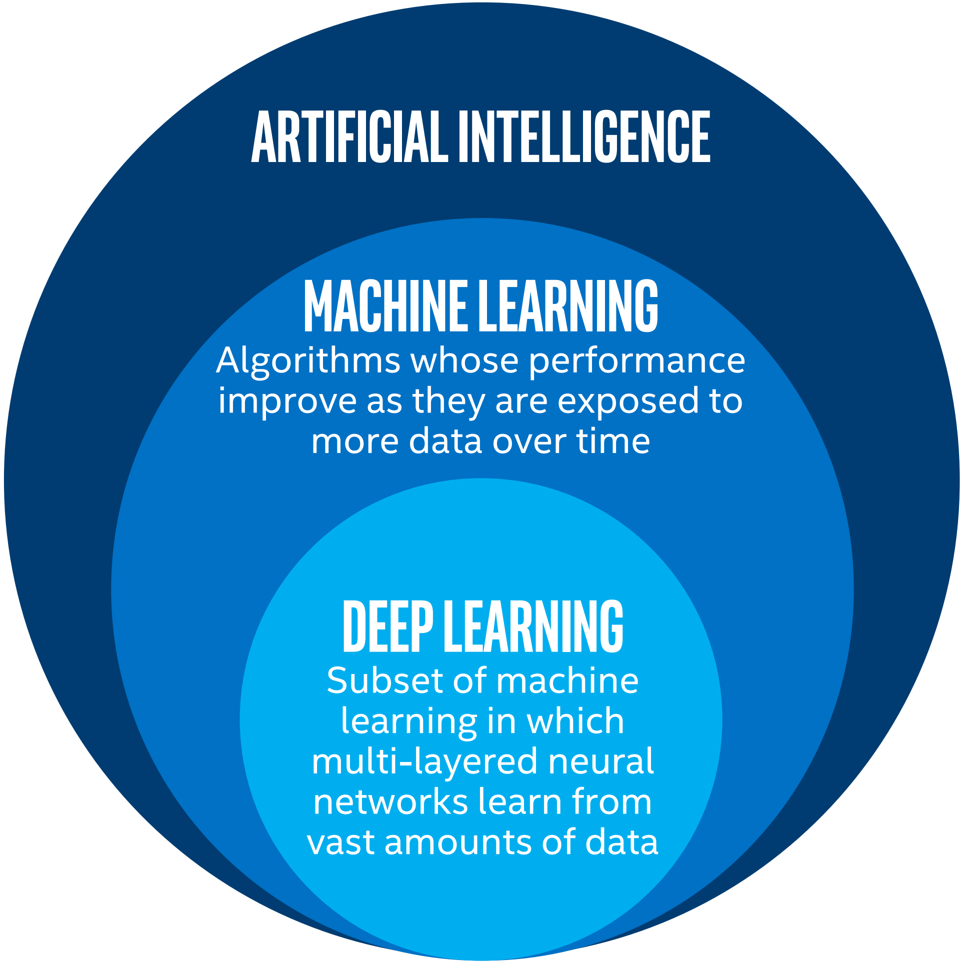
Intel Software Development Tools

- Intel Parallel Studio XE (IPS XE) Tool Suites
- News on the IPS XE 2020 Edition
- Upcoming **oneAPI** development tools concept

Intel optimized AI Solutions and Directions

ARTIFICIAL INTELLIGENCE

is the ability of machines to learn from experience, without explicit programming, in order to perform cognitive functions associated with the human mind



ARTIFICIAL INTELLIGENCE

MACHINE LEARNING

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

DEEP LEARNING

Subset of machine learning in which multi-layered neural networks learn from vast amounts of data

ARTIFICIAL INTELLIGENCE

SOLUTIONS

Solution Architects

ARTIFICIAL INTELLIGENCE

AI Solutions Catalog (Public & Internal)

Platforms · *Finance* · *Healthcare* · *Energy* · *Industrial* · *Transport* · *Retail* · *Home* · *More...*

TOOLKITS

App Developers

DEEP LEARNING DEPLOYMENT

OpenVINO™ †

Open Visual Inference & Neural Network Optimization toolkit for inference deployment on CPU, processor graphics, FPGA & VPU using TF, Caffe & MXNet**

Intel® Movidius™ SDK

Optimized inference deployment for all Intel® Movidius™ VPUs using TensorFlow & Caffe**

DEEP LEARNING

COMING SOON!

Intel® Deep Learning Studio‡

Open-source tool to compress deep learning development cycle

LIBRARIES

Data Scientists

MACHINE LEARNING LIBRARIES

Python	R	Distributed
• Scikit-learn	• Cart	• MlLib (on Spark)
• Pandas	• Random Forest	• Mahout
• NumPy	• e1071	

DEEP LEARNING FRAMEWORKS

COMING SOON!

Now optimized for CPU				Optimizations in progress		
TensorFlow*	MXNet*	Caffe*	BigDL/Spark*	Caffe2*	PyTorch*	PaddlePaddle*

FOUNDATION

Library Developers

ANALYTICS, MACHINE & DEEP LEARNING PRIMITIVES

Python

Intel distribution optimized for machine learning

DAAL

Intel® Data Analytics Acceleration Library (for machine learning)

MKL-DNN **clDNN**

Open-source deep neural network functions for CPU, processor graphics

DEEP LEARNING GRAPH COMPILER

Intel® nGraph™ Compiler (Alpha)

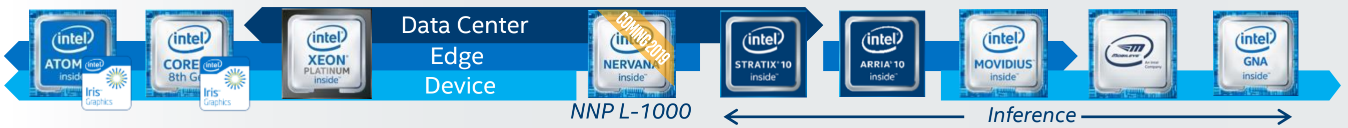
Open-sourced compiler for deep learning model computations optimized for multiple devices (CPU, GPU, NNP) using multiple frameworks (TF, MXNet, ONNX)

HARDWARE

IT System Architects

AI FOUNDATION

DEEP LEARNING ACCELERATORS



† Formerly the Intel® Computer Vision SDK
*Other names and brands may be claimed as the property of others.
All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

Optimization Notice

AI.INTEL.COM

Components Comparison : Intel MKL-DNN vs Intel MKL

MKL-DNN (Open Source)

- Convolution
- Pooling
- ReLU
- Inner Product
- Normalization

MKL (Math Kernel Library)

Linear Algebra

- BLAS
- LAPACK
- ScaLAPACK
- Sparse BLAS
- Sparse Solvers
- Iterative
- PARDISO*
- Cluster Sparse Solver

Fast Fourier Transforms

- Multidimensional
- FFTW interfaces
- Cluster FFT

Vector Math

- Trigonometric
- Hyperbolic
- Exponential
- Log
- Power
- Root
- Vector RNGs

Summary Statistics

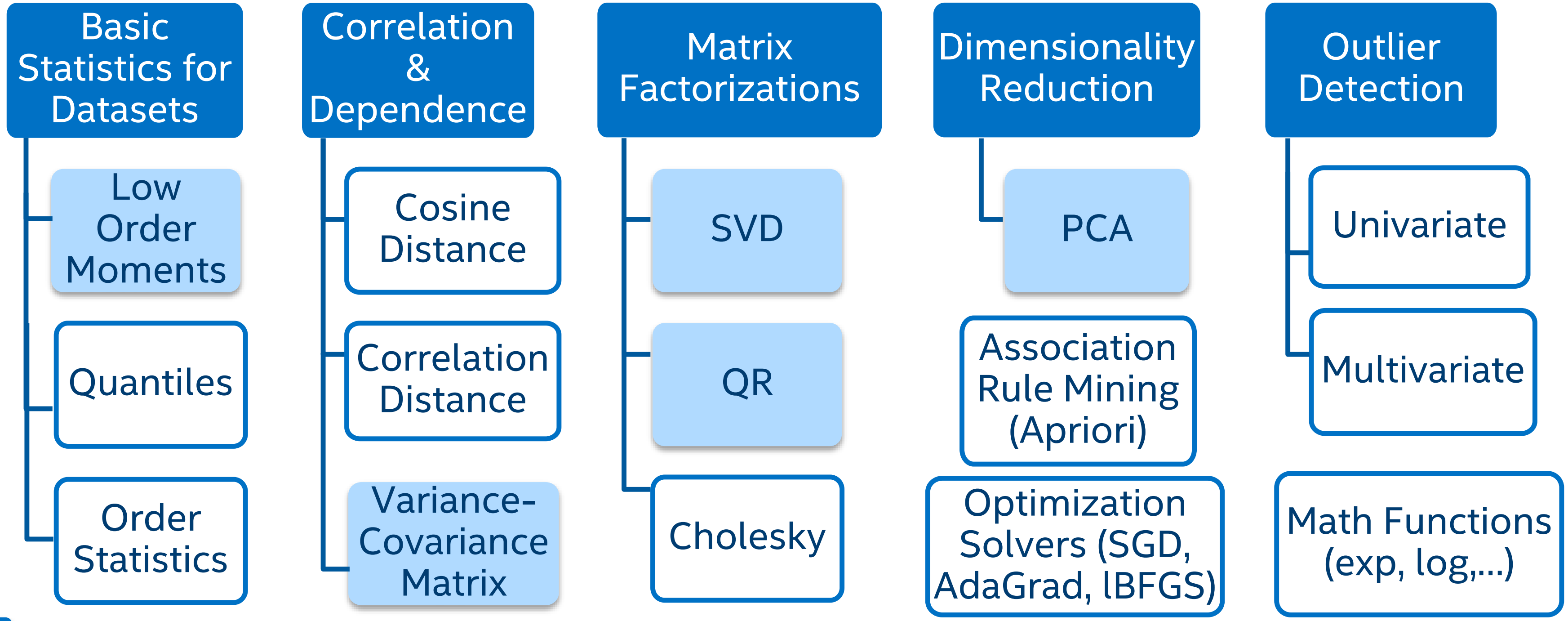
- Kurtosis
- Variation coefficient
- Order statistics
- Min/max
- Variance-covariance

And More…

- Splines
- Interpolation
- Trust Region
- Fast Poisson Solver

Data Transformation & Analysis Algorithms

Intel® Data Analytics Acceleration Library



Basic Statistics for Datasets
- Low Order Moments
- Quantiles
- Order Statistics

Correlation & Dependence
- Cosine Distance
- Correlation Distance
- Variance-Covariance Matrix

Matrix Factorizations
- SVD
- QR
- Cholesky

Dimensionality Reduction
- PCA

Association Rule Mining (Apriori)

Optimization Solvers (SGD, AdaGrad, lBFGS)

Outlier Detection
- Univariate
- Multivariate

Math Functions (exp, log,...)

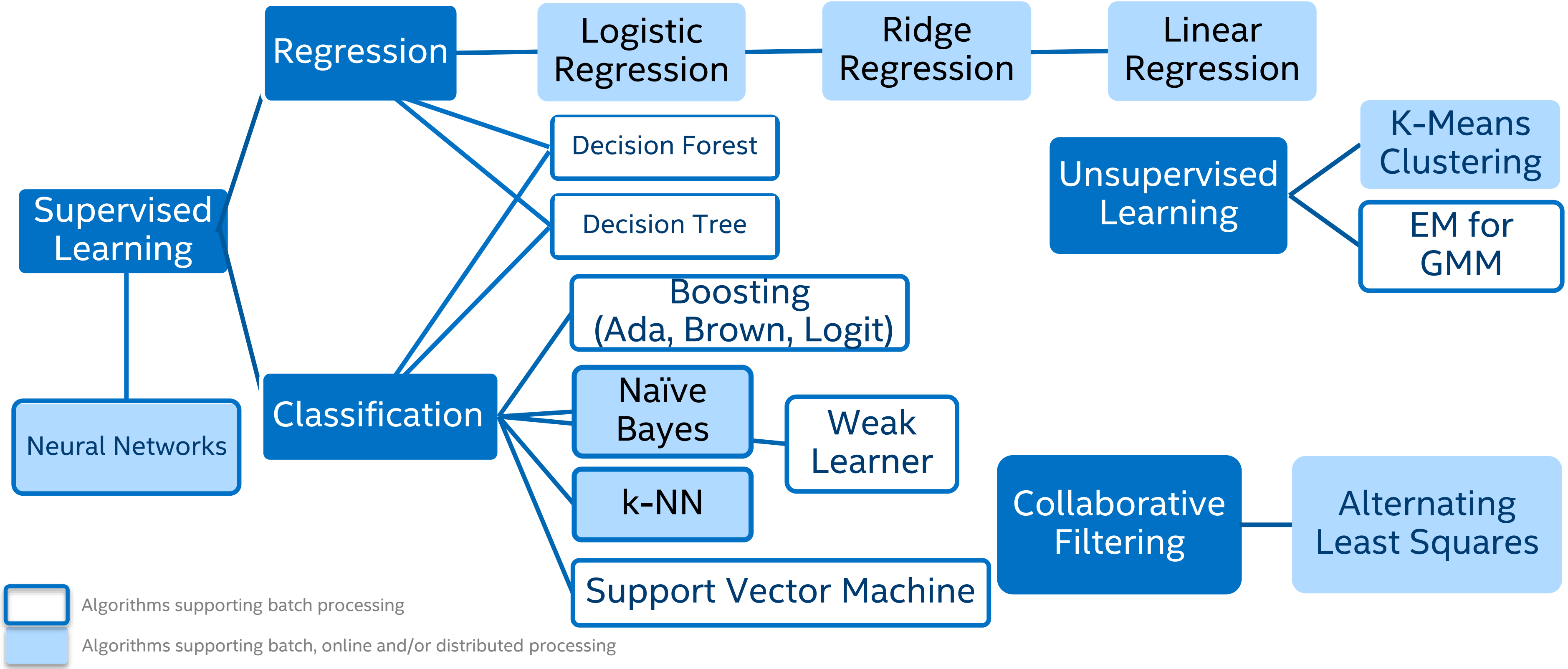
□ Algorithms supporting batch processing

🟦 Algorithms supporting batch, online and/or distributed processing

Optimization Notice

Machine Learning Algorithms

Intel® Data Analytics Acceleration Library



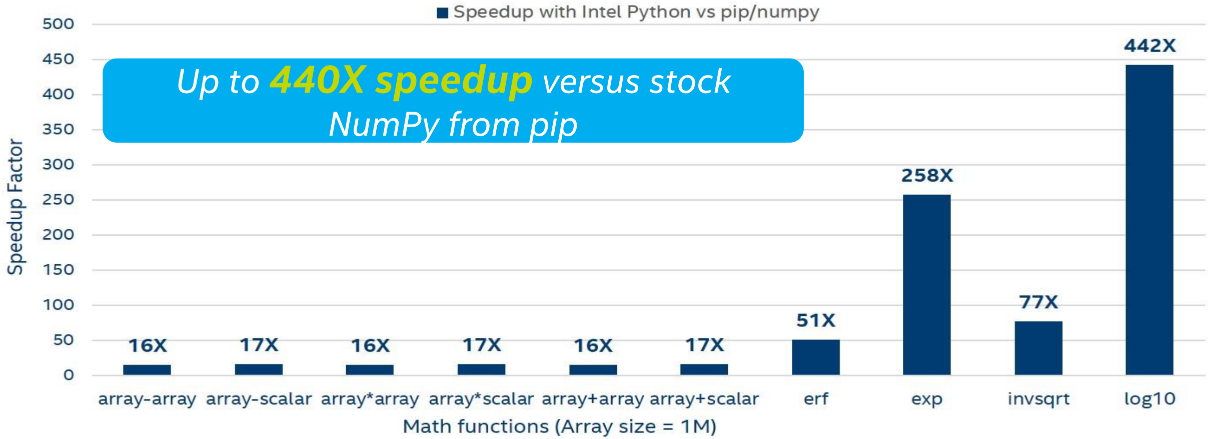
□ Algorithms supporting batch processing

▨ Algorithms supporting batch, online and/or distributed processing

Optimization Notice

Faster Python* with Intel® Distribution for Python

Intel® Distribution for Python* Performance Speedups for Select Math Functions on Intel® Xeon™ Processors



*Up to **440X speedup** versus stock NumPy from pip*

Configuration: Hardware: Intel® Xeon® CPU E5-2699 v4 @ 2.20GHz (2 sockets, 22 cores per socket, 1 thread per core – HT is off), 256GB DDR4 @ 2400MHz. Software: Stock: CentOS Linux release 7.3.1611 (Core), python 3.6.2, pip 9.0.1, numpy 1.13.1, scipy 0.19.1, scikit-learn 0.19.0. Intel® Distribution for Python* 2018 Gold: mkl 2018.0.0 intel_4, daal 2018.0.0.20170814, numpy 1.13.1 py36_intel_15, openmp 2018.0.0 intel_7, scipy 0.19.1 np113py36_intel_11, scikit-learn 0.18.2 np113py36_intel_3

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 http://www.intel.com/performance. Benchmark Source: Intel Corporation.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804. *Other brands and names are the property of their respective owners.

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

Software & workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark & MobileMark, are measured using specific computer systems, components, software, operations & functions. Any change to any of those factors may cause the results to vary. You should consult other information & 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 http://www.intel.com/performance.

Optimization Notice [2]Paid versions only.

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

Intel-Optimized AI Frameworks

Popular DL Frameworks are now optimized for CPU!

CHOOSE YOUR FAVORITE FRAMEWORK

TensorFlow™ *

Caffe *

mxnet *

BigDL FOR APACHE Spark™ *

PYTORCH

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

More under optimization: Caffe2 * PYTORCH * PaddlePaddle *

SEE ALSO: Machine Learning Libraries for Python (Scikit-learn, Pandas, NumPy), R (Cart, randomForest, e1071), Distributed (MlLib on Spark, Mahout)
**Limited availability today*
Other names and brands may be claimed as the property of others.

INTEL® XEON® PROCESSOR PLATFORM PERFORMANCE

Hardware plus optimized software

INFERENCE THROUGHPUT

198x

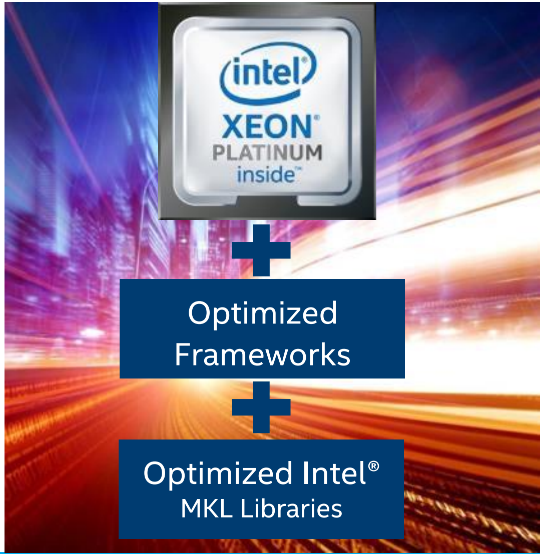
Intel® Xeon® Platinum 8180 Processor
higher Intel optimized Caffe GoogleNet v1 with Intel® MKL
inference throughput compared to
Intel® Xeon® Processor E5-2699 v3 with BVLC-Caffe

TRAINING THROUGHPUT

127x

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

Inference and training throughput uses FP32 instructions



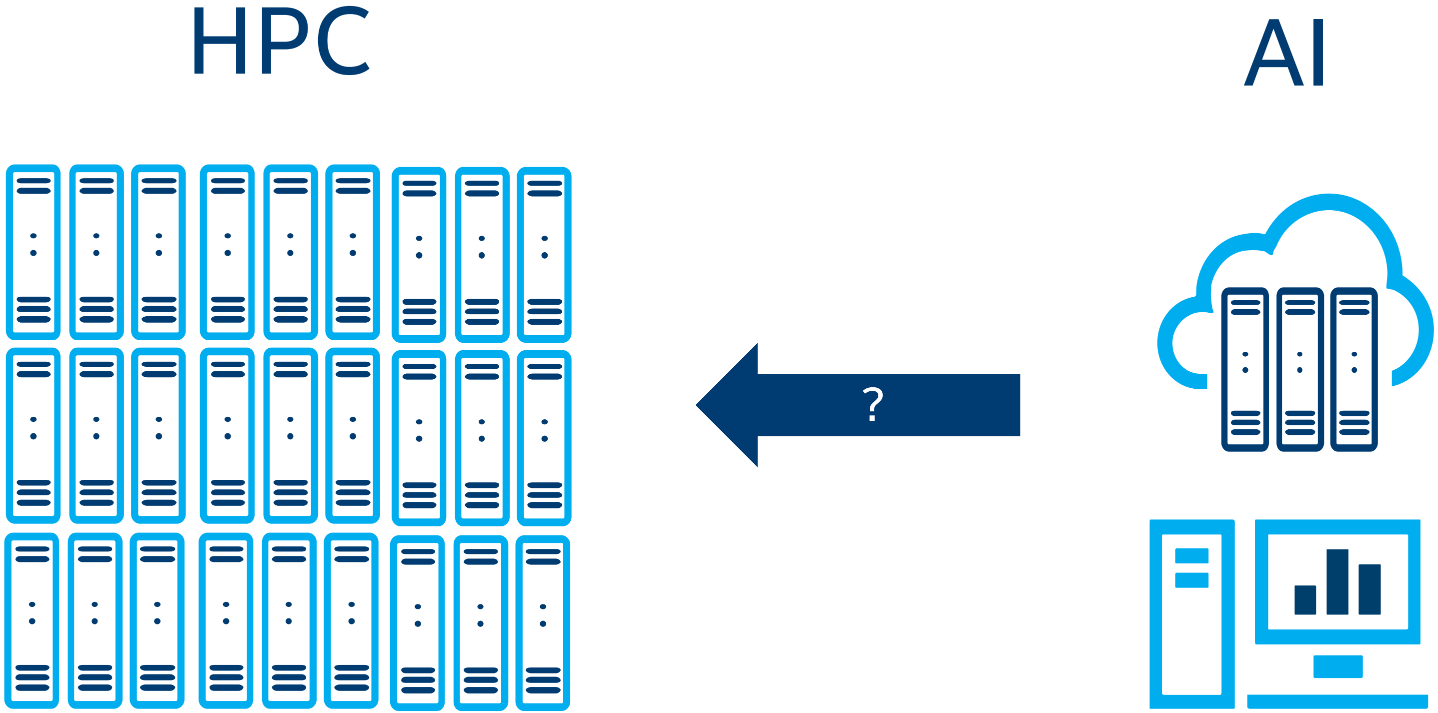
Deliver significant AI performance with hardware and software optimizations on Intel® Xeon® Scalable Family

Up to 191X Intel® Xeon® Platinum 8180 Processor higher Intel optimized Caffe Resnet50 with Intel® MKL inference throughput compared to Intel® Xeon® Processor E5-2699 v3 with BVLC-Caffe
Up to 93X Intel® Xeon® Platinum 8180 Processor higher Intel optimized Caffe Resnet50 with Intel® MKL training throughput compared to Intel® Xeon® Processor E5-2699 v3 with BVLC-Caffe

Optimization Notice

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

Use your Intel Architecture based HPC Infrastructure ---> Shorten Deep Learning Training Time



AI (ML & DL) SOFTWARE STACK FOR INTEL® PROCESSORS

OpenVINO™ TOOLKIT
Chainer
Caffe
PaddlePaddle
BigDL
PYTORCH
Caffe2
mxnet
TensorFlow

Intel MKL

Intel MKL-DNN

Intel Processors

Distributed DL Training

Uber's open source Distributed training framework for TensorFlow

Intel MPI

Intel MLSL

Intel Processors

More details on Intel and AI at GridKa School

Join the Intel hands-on Training : Thursday afternoon at 13:15

Title : Enhance Machine Learning Performance with Intel® Software tools

1st Session:

- Introduction of Intel tools for ML and DL
  - DLBoost, VNNI instructions ; Intel distribution for Python

2nd session:

- Classical ML
  - Numpy and MKL, K-Means, clustering and DAAL4PY Distributed ML algorithms

3rd session:

- Intel MKL for Deep Neural Network Intel optimized Framework and Tensorflow
- Distributed Tensorflow with Horovod

THANK YOU

Legal Disclaimer & Optimization Notice

Performance results are based on testing as of August 2017 to September 2018 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.

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

Optimization Notice

Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.
Notice revision #20110804

intel®

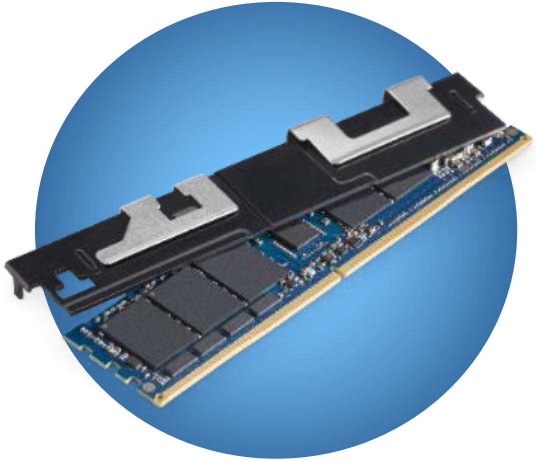
Software

INTEL® OPTANE™ DC PERSISTENT MEMORY

THE BEST OF BOTH WORLDS WITH INTEL® OPTANE™ DC PERSISTENT MEMORY

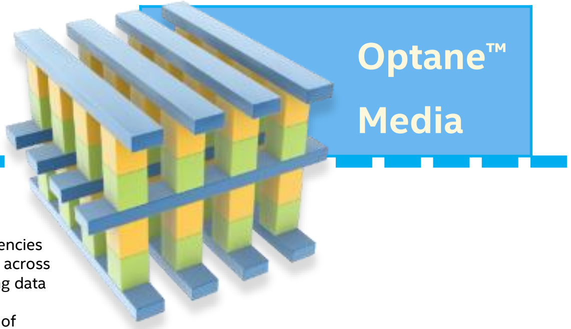
DRAM ATTRIBUTES

Performance comparable to DRAM at *low latencies*[1]



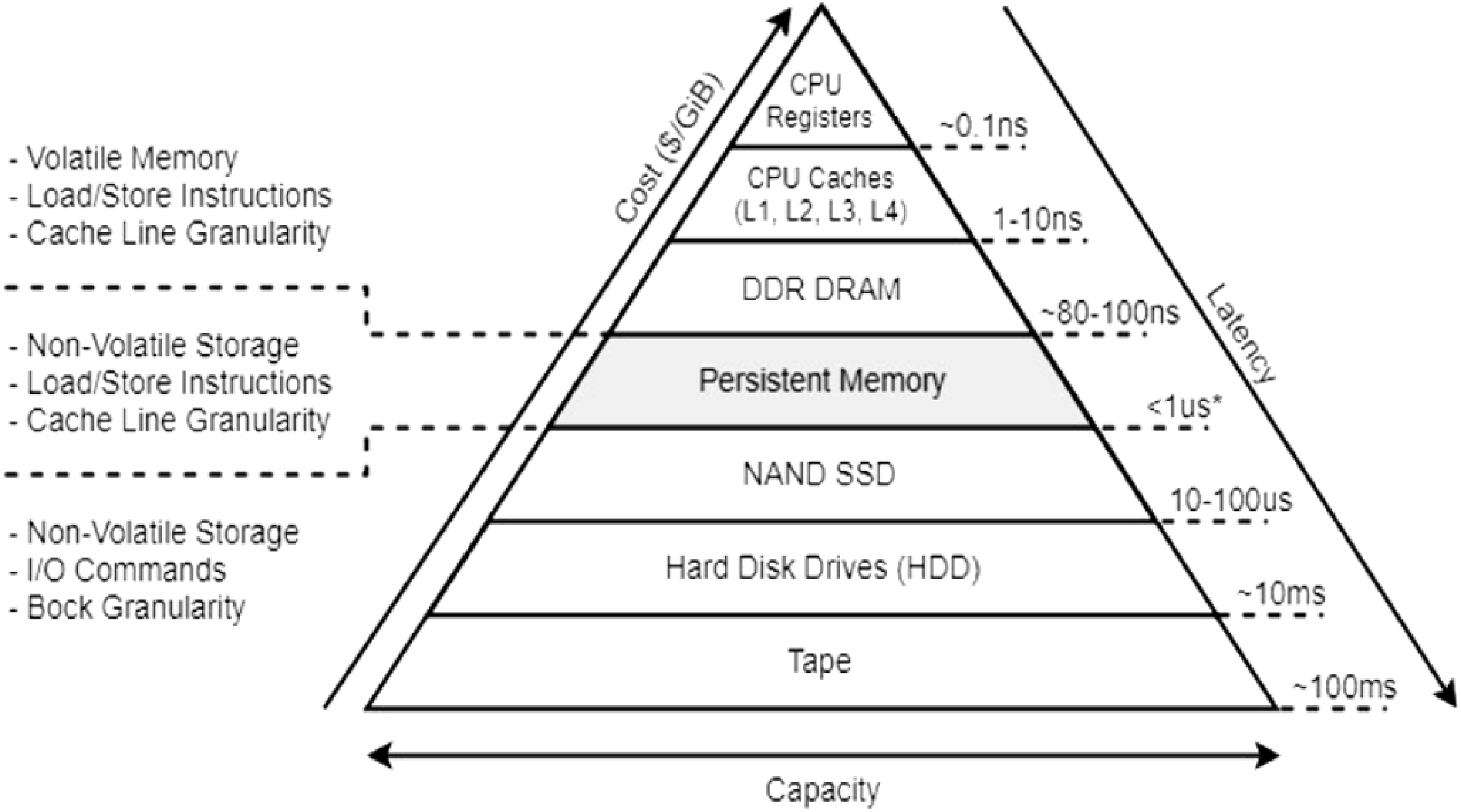
NAND SSD ATTRIBUTES

Data persistence with higher capacity than DRAM[2]



1. "*Fast performance comparable to DRAM*" - Intel persistent memory is expected to perform at latencies near DDR4 DRAM. Benchmarks and proof points forthcoming. "*low latencies*" - Data transferred across the memory bus causes latencies to be orders of magnitude lower when compared to transferring data across PCIe or I/O bus' to NAND/Hard Disk. Benchmarks and proof points forthcoming.
2. Intel persistent memory offers 3 different capacities – 128GB, 256GB, 512GB. Individual DIMMs of DDR4 DRAM max out at 256GB.

Latency Estimates for different Memory and Storage Devices

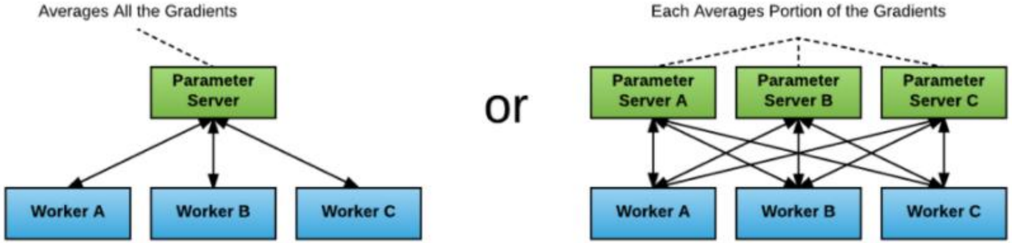


Optimization Notice

Distributed TensorFlow™ Compare

Distributed Tensorflow with Parameter Server

With Parameter Server

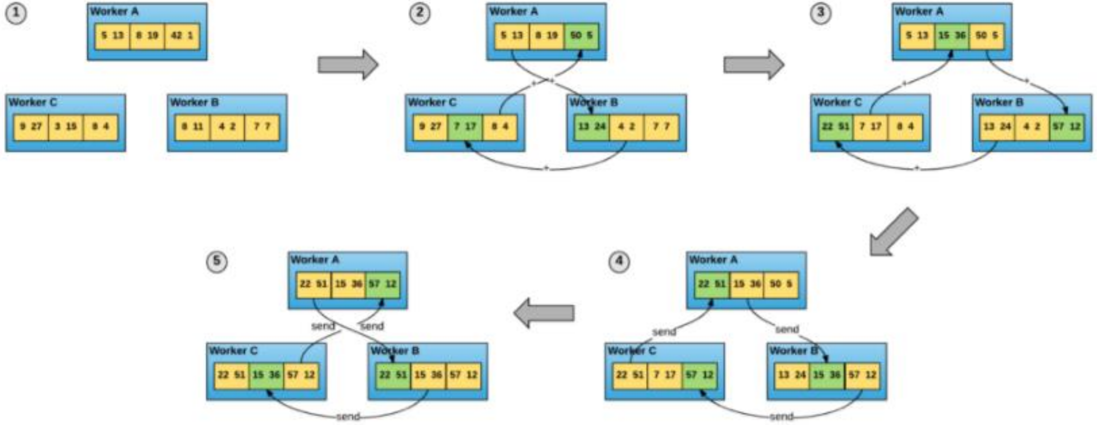


The parameter server model for distributed training jobs can be configured with different ratios of parameter servers to workers, each with different performance profiles.

HOROVOD

Uber's open source Distributed training framework for TensorFlow

No Parameter Server



The ring all-reduce algorithm allows worker nodes to average gradients and disperse them to all nodes without the need for a parameter server.

Source: https://eng.uber.com/horovod/

UNSUPERVISED LEARNING EXAMPLE

MACHINE LEARNING

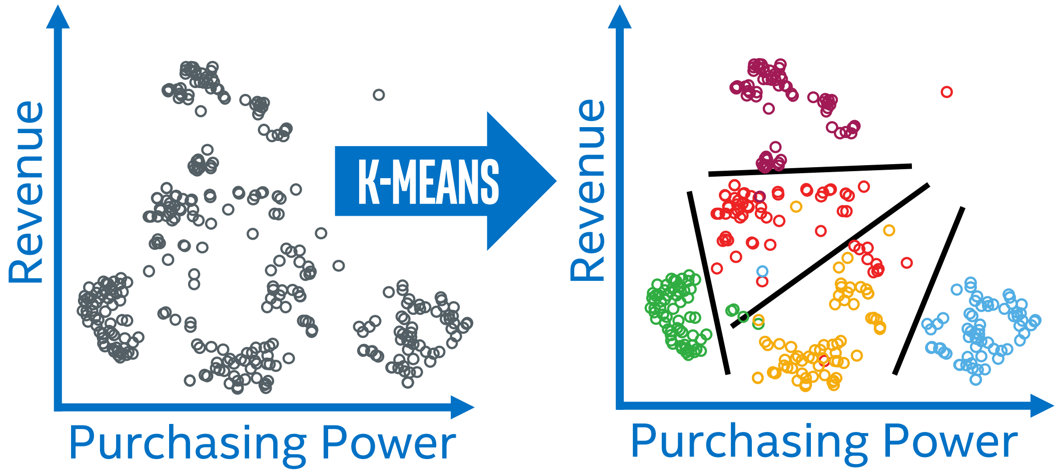
- Regression
- Classification
- **Clustering**
- Decision Trees
- Data Generation
- Image Processing

DEEP LEARNING

- Speech Processing
- Natural Language Processing
- Recommender Systems
- Adversarial Networks
- Reinforcement Learning

MACHINE LEARNING (CLUSTERING)

An 'Unsupervised Learning' Example



Choose the right AI approach for your challenge

Optimization Notice

SUPERVISED LEARNING EXAMPLE

MACHINE LEARNING

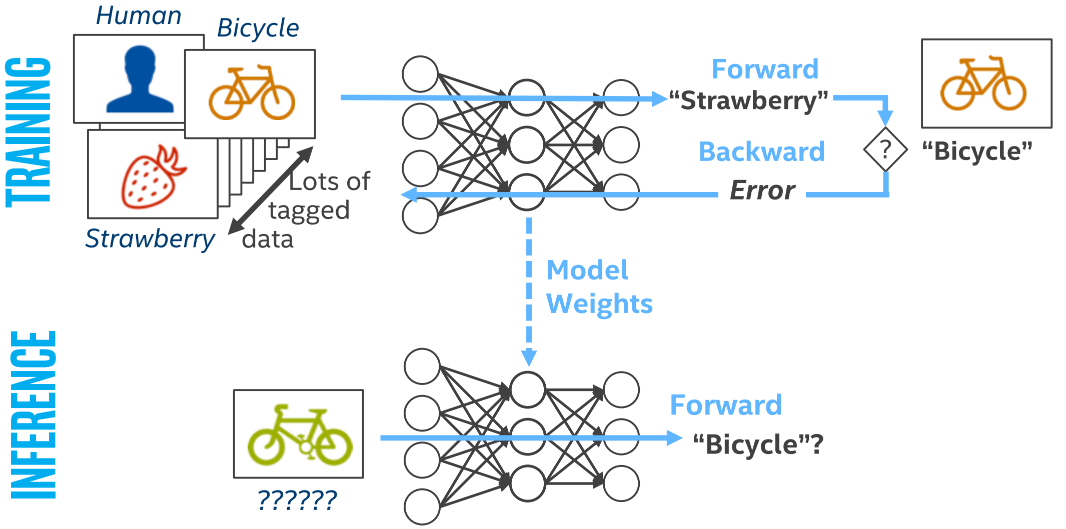
- Regression
- Classification
- Clustering
- Decision Trees
- Data Generation
- **Image Processing**

DEEP LEARNING

- Speech Processing
- Natural Language Processing
- Recommender Systems
- Adversarial Networks
- Reinforcement Learning

DEEP LEARNING (IMAGE RECOGNITION)

A 'Supervised Learning' Example



Choose the right AI approach for your challenge

REINFORCEMENT LEARNING EXAMPLE

MACHINE LEARNING

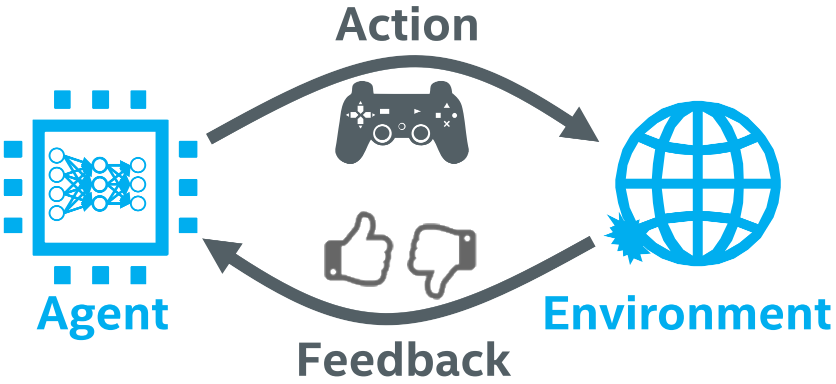
- Regression
- Classification
- Clustering
- Decision Trees
- Data Generation

DEEP LEARNING

- Image Processing
- Speech Processing
- Natural Language Processing
- Recommender Systems
- Adversarial Networks
- Reinforcement Learning

DEEP LEARNING (REINFORCEMENT LEARNING)

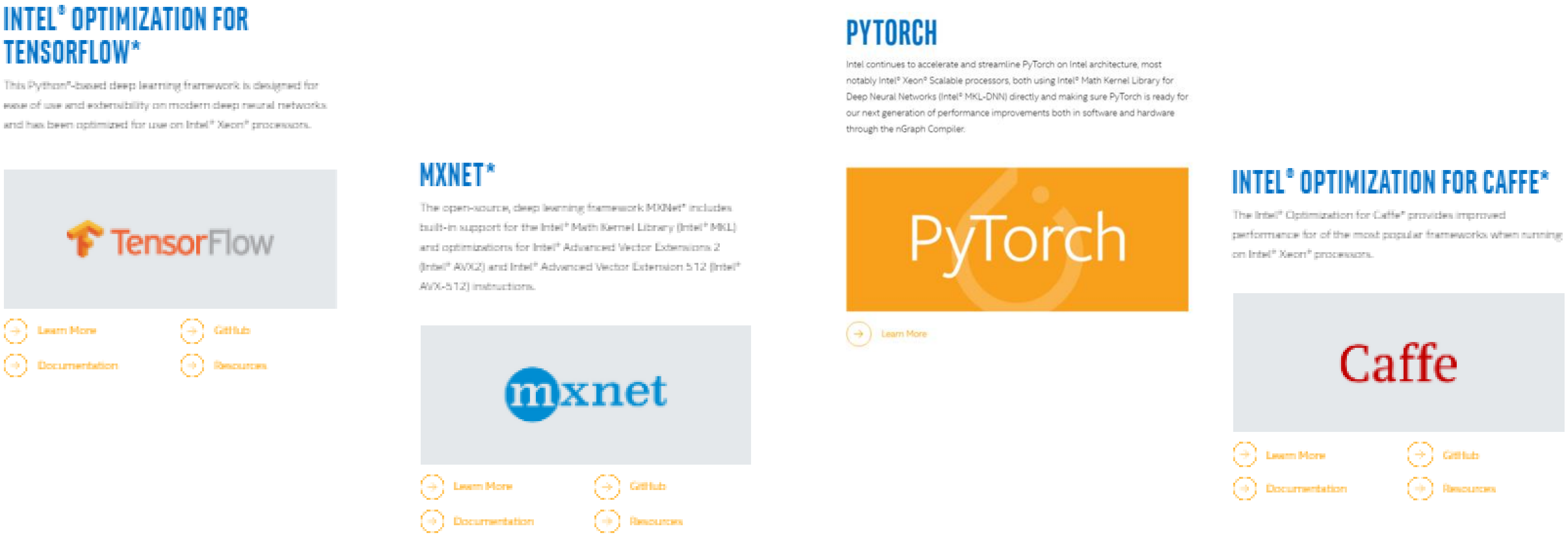
A 'Deep Reinforcement Learning' Example



Choose the right AI approach for your challenge

Intel-Optimized Frameworks: How To Get?

Check out our intel.ai for the framework optimizations page



INTEL® OPTIMIZATION FOR TENSORFLOW*

This Python*-based deep learning framework is designed for ease of use and extensibility on modern deep neural networks and has been optimized for use on Intel® Xeon® processors.

Learn More
GitHub
Documentation
Resources

MXNET*

The open-source, deep learning framework MXNet* includes built-in support for the Intel® Math Kernel Library (Intel® MKL) and optimizations for Intel® Advanced Vector Extensions 2 (Intel® AVX2) and Intel® Advanced Vector Extension 512 (Intel® AVX-512) instructions.

Learn More
GitHub
Documentation
Resources

PYTORCH

Intel continues to accelerate and streamline PyTorch on Intel architecture, most notably Intel® Xeon® Scalable processors, both using Intel® Math Kernel Library for Deep Neural Networks (Intel® MKL-DNN) directly and making sure PyTorch is ready for our next generation of performance improvements both in software and hardware through the nGraph Compiler.

Learn More

INTEL® OPTIMIZATION FOR CAFFE*

The Intel® Optimization for Caffe* provides improved performance for of the most popular frameworks when running on Intel® Xeon® processors.

Learn More
GitHub
Documentation
Resources

Installation example for optimized Tensorflow / Keras for a Medial Sample application
https://docs.google.com/document/d/1AAsWLSfYBx-pYfvzFQxOYi4Z1zx3TAEB5534s61Lf8w/edit?usp=sharing

Topology Examples : https://www.intel.ai/framework-optimizations