Intel AI

INTRODUCTION TO ARTIFICIAL INTELLIGENCE USING INTEL® HARDWARE PLATFORM

Dr. Fabio Baruffa

Sr. Technical Consulting Engineer, Intel IAGS

NAVIGATING THE AI PERFORMANCE PACKAGE

INTRODUCTION TO AI
Overview of Deep Learning Software

INTEL® XEON® SCALABLE PROCESSORS
Second generation: Cascade Lake

INTEL® DEEP LEARNING BOOST
Intel® AVX-512 Vector Neural Network Instructions (VNNI)

NAVIGATING THE AI PERFORMANCE PACKAGE

INTRODUCTION TO AI

Overview of Deep Learning Software

- What are AI, Machine Learning, and Deep Learning?
- Deep Learning Software breakdown
- Popular AI Neural Networks and their uses
- Intel's AI software tools

WHAT IS AI?

Regression

Classification

Clustering

Decision Trees

Data Generation

Image Processing

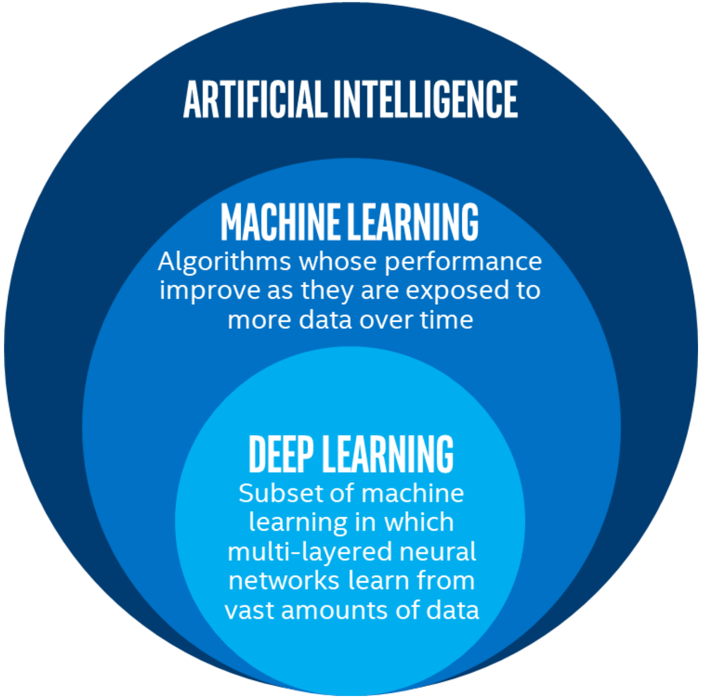
Speech Processing

Natural Language Processing

Recommender Systems

Adversarial Networks

Reinforcement Learning



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

No one size fits all approach to AI

UNSUPERVISED LEARNING EXAMPLE

MACHINE LEARNING

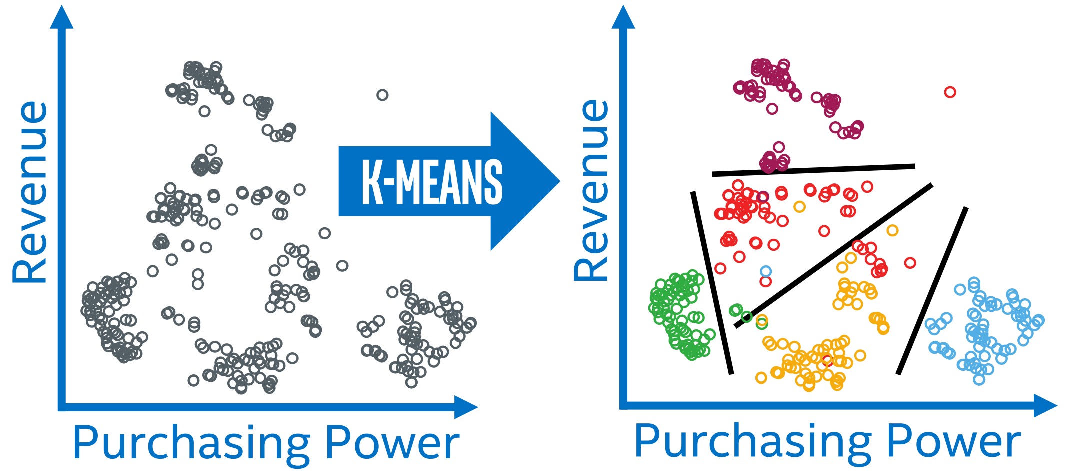
- Regression
- Classification
- **Clustering**
- Decision Trees
- Data Generation
- Image Processing

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

SUPERVISED LEARNING EXAMPLE

MACHINE LEARNING

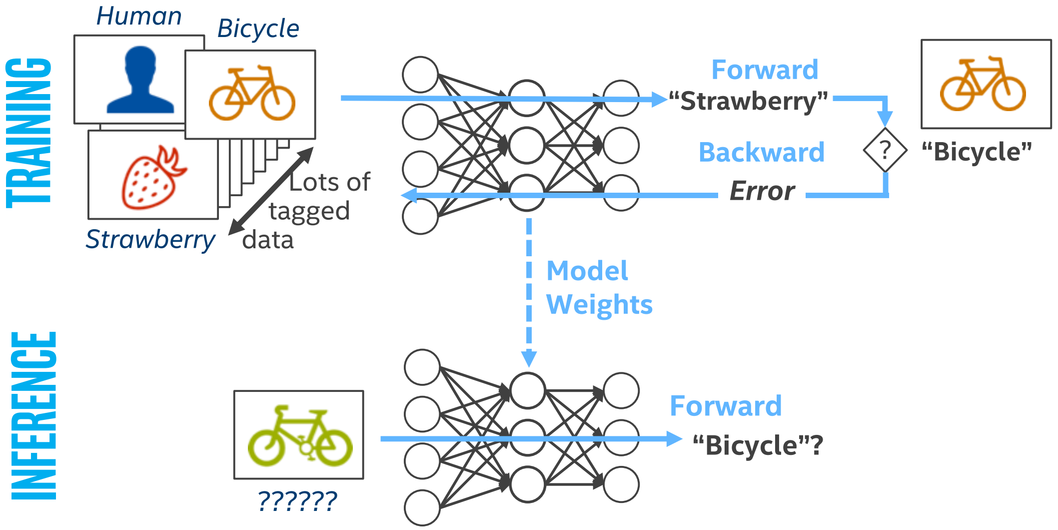
- Regression
- Classification
- Clustering
- Decision Trees
- Data Generation
- **Image Processing**

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

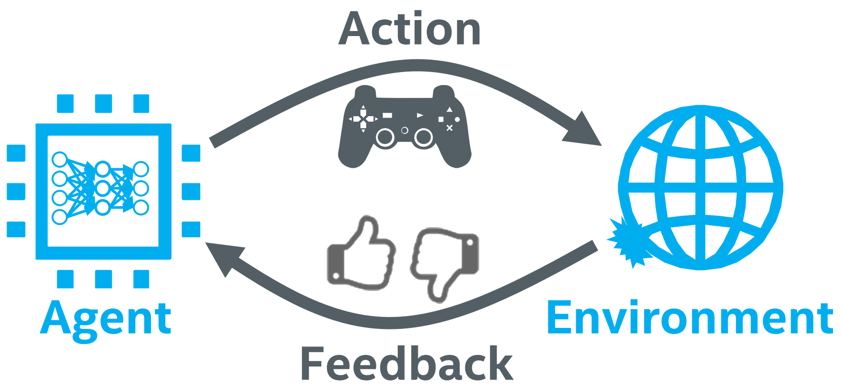
- Regression
- Classification
- Clustering
- Decision Trees
- Data Generation

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

DEEP LEARNING GLOSSARY

LIBRARY

MKL-DNN
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 & deep learning algorithms, topologies & frameworks

FRAMEWORK

TensorFlow
BigDL for Spark
mxnet
PaddlePaddle
PYTORCH
Caffe

Open-source software environments that facilitate deep learning model development & 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

DEEP LEARNING USAGES & KEY TOPOLOGIES

Image Recognition

Resnet-50
Inception V3
MobileNet
SqueezeNet

Language Translation

GNMT

Object Detection

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

Text to Speech

Wavenet

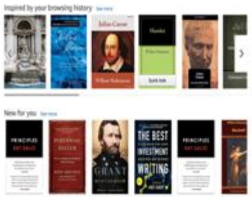
Understand Legalese

Image Segmentation

Mask R-CNN

Recommendation System

Wide & Deep, NCF

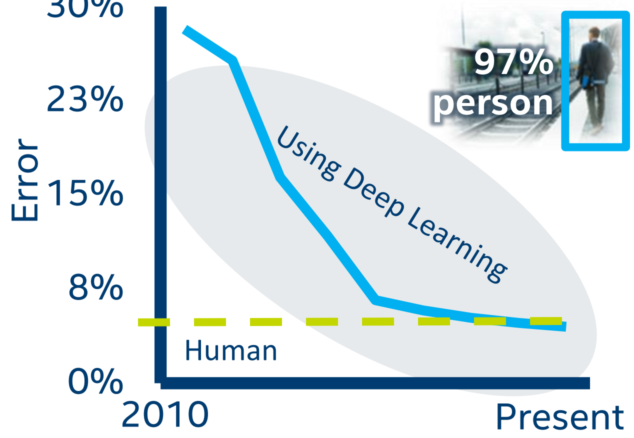


There are many deep learning usages and topologies for each

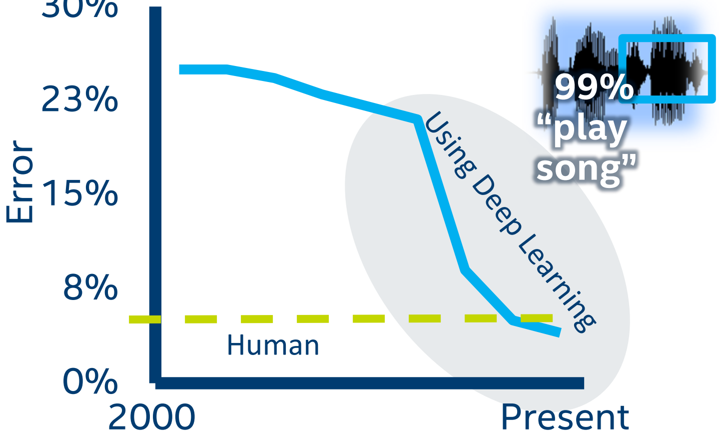
DEEP LEARNING BREAKTHROUGHS

Machines able to meet or exceed human image & speech recognition

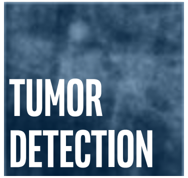
IMAGE RECOGNITION



SPEECH RECOGNITION

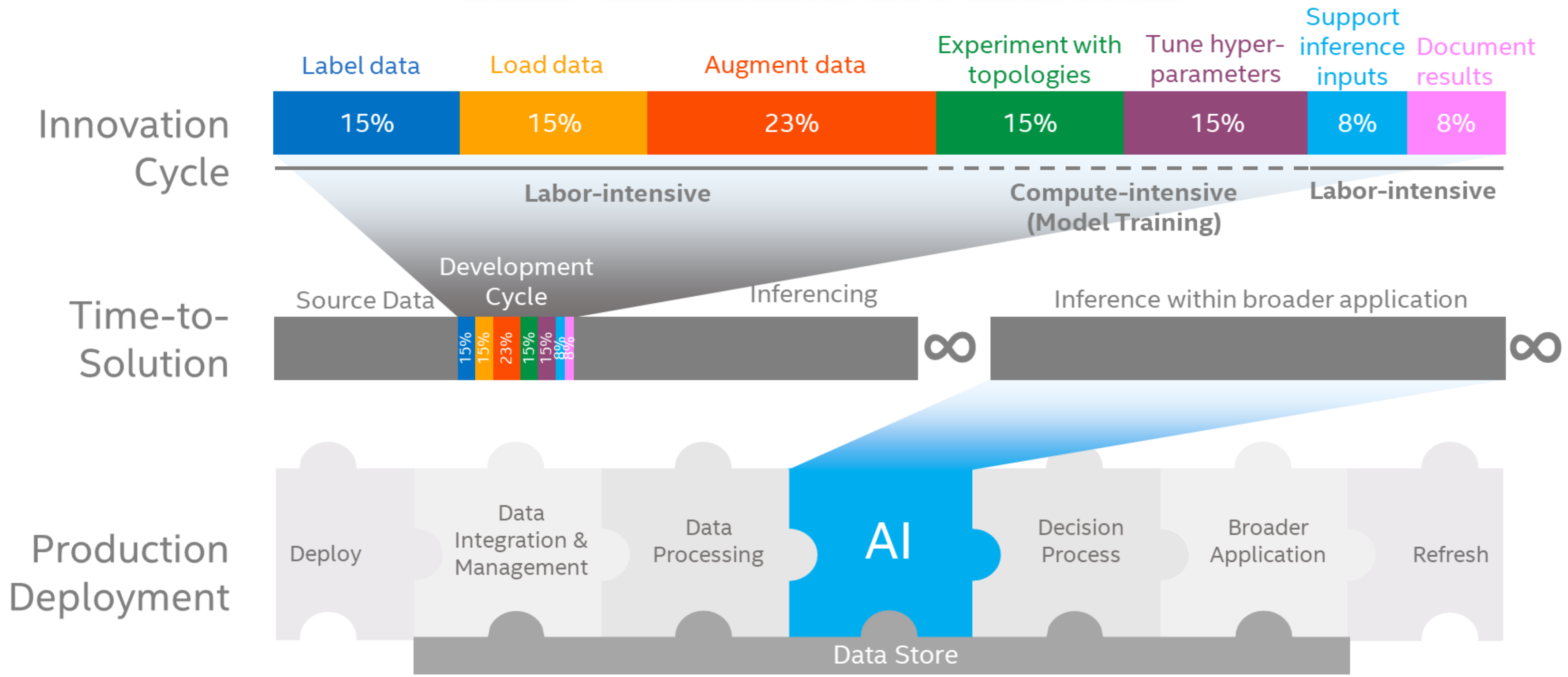


e.g.



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)

DEEP LEARNING IN PRACTICE



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

Visit: www.intel.ai/technology

SPEED UP DEVELOPMENT

using open AI software

MACHINE LEARNING | DEEP LEARNING

TOOLKITS
App developers

Analytics Zoo
Open source platform for building E2E Analytics & AI applications on Apache Spark* with distributed TensorFlow*, Keras*, BigDL

OpenVINO
Deep learning inference deployment on CPU/GPU/FPGA/VPU for Caffe*, TensorFlow*, MXNet*, ONNX*, Kaldi*

NAUTA
Open source, scalable, and extensible distributed deep learning platform built on Kubernetes (BETA)

LIBRARIES
Data scientists

Python
- Scikit-learn
- Pandas
- NumPy

R
- Cart
- Random Forest
- e1071

Distributed
- MlLib (on Spark)
- Mahout

Intel-optimized Frameworks

TensorFlow* Caffe2* ONNX* mxnet* PyTorch* BigDL*

And more framework optimizations underway including PaddlePaddle*, Chainer*, CNTK* & others

KERNELS
Library developers

Intel® Distribution for Python*
Intel distribution optimized for machine learning

Intel® Data Analytics Acceleration Library (Intel® DAAL)
High performance machine learning & data analytics library

Intel® Math Kernel Library for Deep Neural Networks (Intel® MKL-DNN)
Open source DNN functions for CPU / integrated graphics

nGraph
Open source compiler for deep learning model computations optimized for multiple devices (CPU, GPU, NNP) from multiple frameworks (TF, MXNet, ONNX)

NAVIGATING THE AI PERFORMANCE PACKAGE

INTRODUCTION TO AI

Overview of Deep Learning Software

INTEL® XEON® SCALABLE PROCESSORS

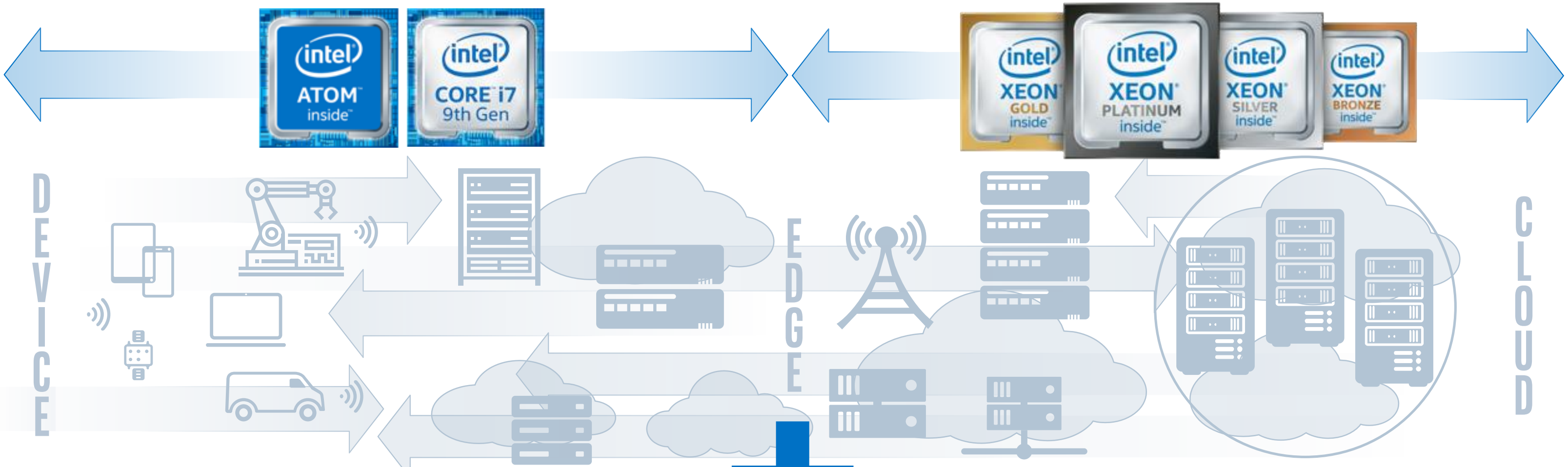
Second generation: Cascade Lake

- Deploy AI Everywhere on Intel® Architecture
- Intel® AVX-512 (Advanced Vector Instructions)

Visit: www.intel.ai/technology

DEPLOY AI ANYWHERE

with unprecedented hardware choice



IF NEEDED

AUTOMATED DRIVING	DEDICATED MEDIA/VISION	FLEXIBLE ACCELERATION	DEDICATED DL INFERENCE	DEDICATED DL TRAINING	GRAPHICS, MEDIA & ANALYTICS
MOBILEYE An Intel Company	intel MOVIDIUS inside	intel FPGA	2H'19 intel NERVANA inside NNP-I	2H'19 intel NERVANA inside NNP-T	FUTURE Intel GPU

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

HARDWARE

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

intel AI

	MAINSTREAM		INTENSIVE
DEEP LEARNING → TRAINING	intel XEON PLATINUM inside	+	intel NERVANA inside (COMING 2019)
DEEP LEARNING → INFERENCE	intel ATOM inside; intel CORE 8th Gen; intel XEON PLATINUM inside	+	intel STRATIX 10 inside; intel CORE i7 8th Gen; intel IRIS; intel MOVIDIUS inside; intel GNA inside
MOST OTHER AI	intel ATOM inside; intel CORE 8th Gen; intel XEON PLATINUM inside		

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

INTRODUCING THE ADVANCED PERFORMANCE OF INTEL® XEON® PLATINUM 9200 PROCESSORS

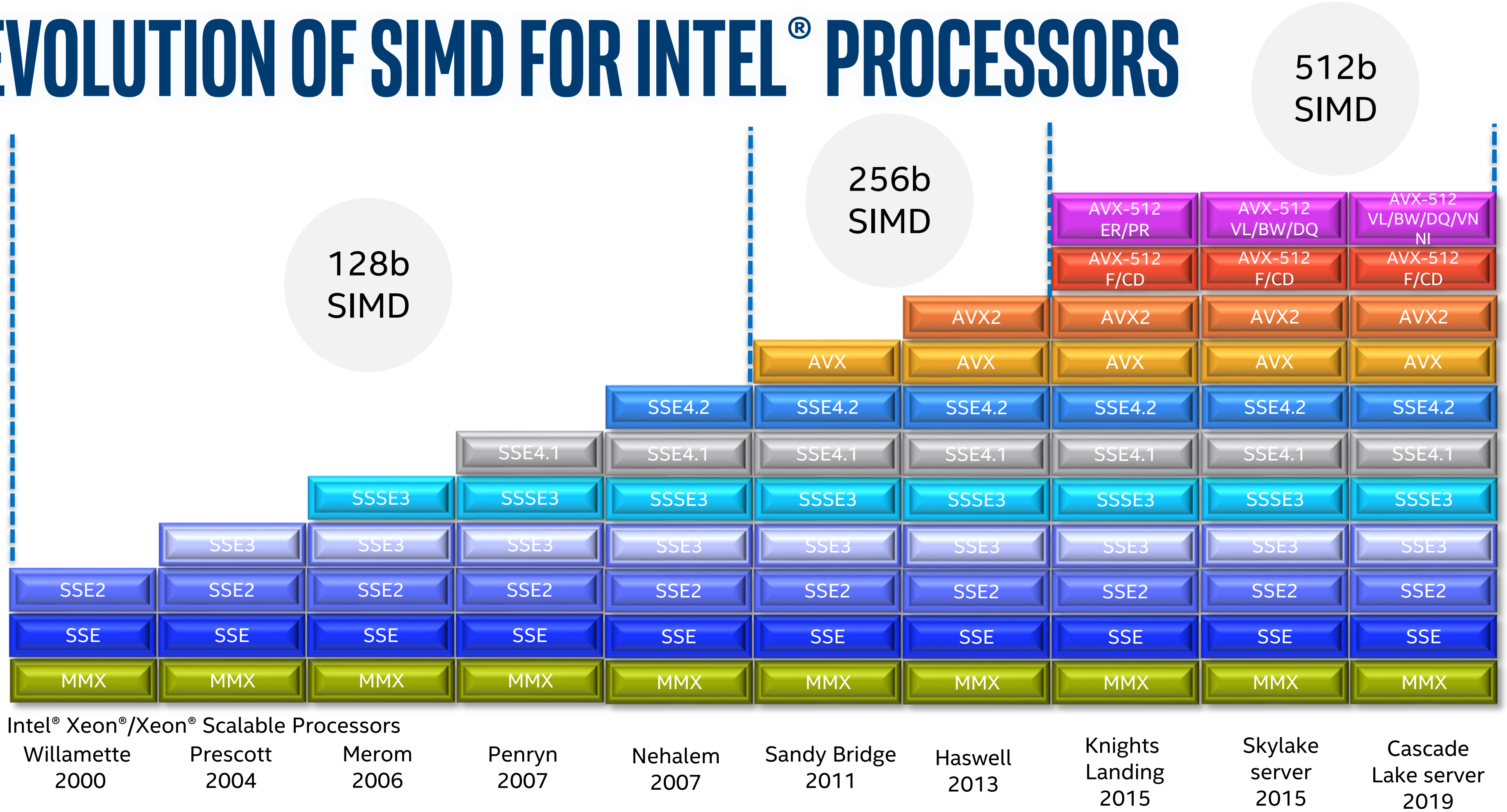
known as Cascade Lake

LEADERSHIP XEON PERFORMANCE

- HIGH PERFORMANCE COMPUTING
- ANALYTICS & ARTIFICIAL INTELLIGENCE
- HIGH DENSITY INFRASTRUCTURE

- UP TO **112** CORES 2S SYSTEM
- UP TO **2X** MORE COMPUTE DENSITY
- UP TO **3.8** GHZ INTEL® TURBO BOOST TECHNOLOGY 2.0
- UP TO **3** TB DDR4-2933 MT/s 2S SYSTEM

EVOLUTION OF SIMD FOR INTEL® PROCESSORS

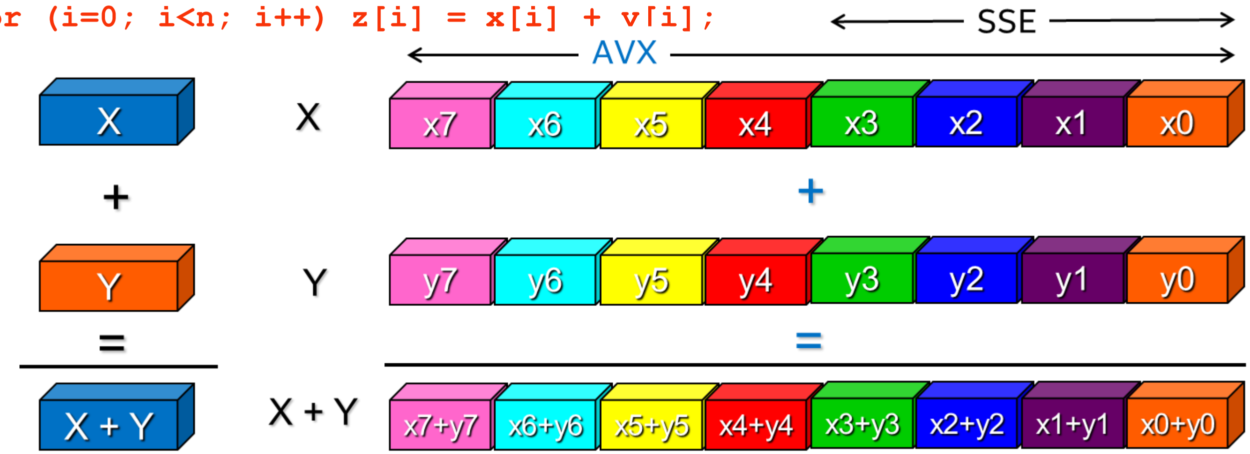


SIMD PROCESSING

Single instruction multiple data (**SIMD**) allows to execute the same operation on multiple data elements using larger registers.

- Scalar mode
  - one instruction produces one result
  - E.g. vadd**ss**, (vadd**sd**)
- Vector (SIMD) mode
  - one instruction can produce multiple results
  - E.g. vadd**ps**, (vadd**pd**)

```
for (i=0; i<n; i++) z[i] = x[i] + y[i];
```



- SSE (128 Bits reg.): -> 4 floats
- AVX (256 Bits reg.): -> 8 floats
- AVX512 (512 Bits reg.): -> 16 floats

AVX-512 COMPRESS AND EXPAND

VCOMPRESSPD|PS|D|Q	Store sparse packed floating-point values into dense memory
VEXPANDPD|PS|D|Q	Load sparse packed floating-point values from dense memory

double/single-precision/doubleword/quadword

```
vcompresspd YMMWORD PTR [rsi+rax*8]{k1}, ymm1
```

COMPRESS LOOP PATTERN

AUTO-VECTORIZATION

https://godbolt.org/z/x7gNfb

```
int compress(double *a, double * __restrict b, int na)
{
    int nb = 0;
    for (int ia=0; ia <na; ia++)
    {
        if (a[ia] > 0.)
            b[nb++] = a[ia];

    }

    return nb;
}
```

COMPRESS LOOP PATTERN

AUTO-VECTORIZATION

https://godbolt.org/z/x7gNfb

```
int compress(double *a, double *
__restrict b, int na)
{
    int nb = 0;
    for (int ia=0; ia <na; ia++)
    {
        if (a[ia] > 0.)
            b[nb++] = a[ia];

    }

    return nb;
}
```

Targeting Intel® AVX2

-xcore-avx2 -qopt-report-file=stderr -qopt-report-phase=vec

LOOP BEGIN

remark #15344: loop was not vectorized: vector dependence prevents vectorization.

remark #15346: vector dependence: assumed FLOW dependence between b[nb] (7:4) and a[ia] (7:4)

LOOP END

Targeting Intel® AVX-512

-xcore-avx512 -qopt-report-file=stderr -qopt-report-phase=vec

LOOP BEGIN

remark #15300: LOOP WAS VECTORIZED

LOOP END

COMPRESS LOOP PATTERN

AUTO-VECTORIZATION

https://godbolt.org/z/x7gNfb

```
int compress(double *a, double * __restrict b, int na)
{
    int nb = 0;
    for (int ia=0; ia <na; ia++)
    {
        if (a[ia] > 0.)
            b[nb++] = a[ia];

    }
}
```

```
movsxd      rax, eax
xor         r11d, r11d
kmovw       r8d, k1
popcnt      r11d, r8d
vcompresspd YMMWORD PTR [rsi+rax*8]{k1}, ymm1
add         eax, r11d
```

Key Take Aways

Compress/Expand loop pattern doesn’t vectorize on architectures like Intel® AVX2 and the previous ones and does with Intel® AVX512

NAVIGATING THE AI PERFORMANCE PACKAGE

INTRODUCTION TO AI

Overview of Deep Learning Software

INTEL® XEON® SCALABLE PROCESSORS

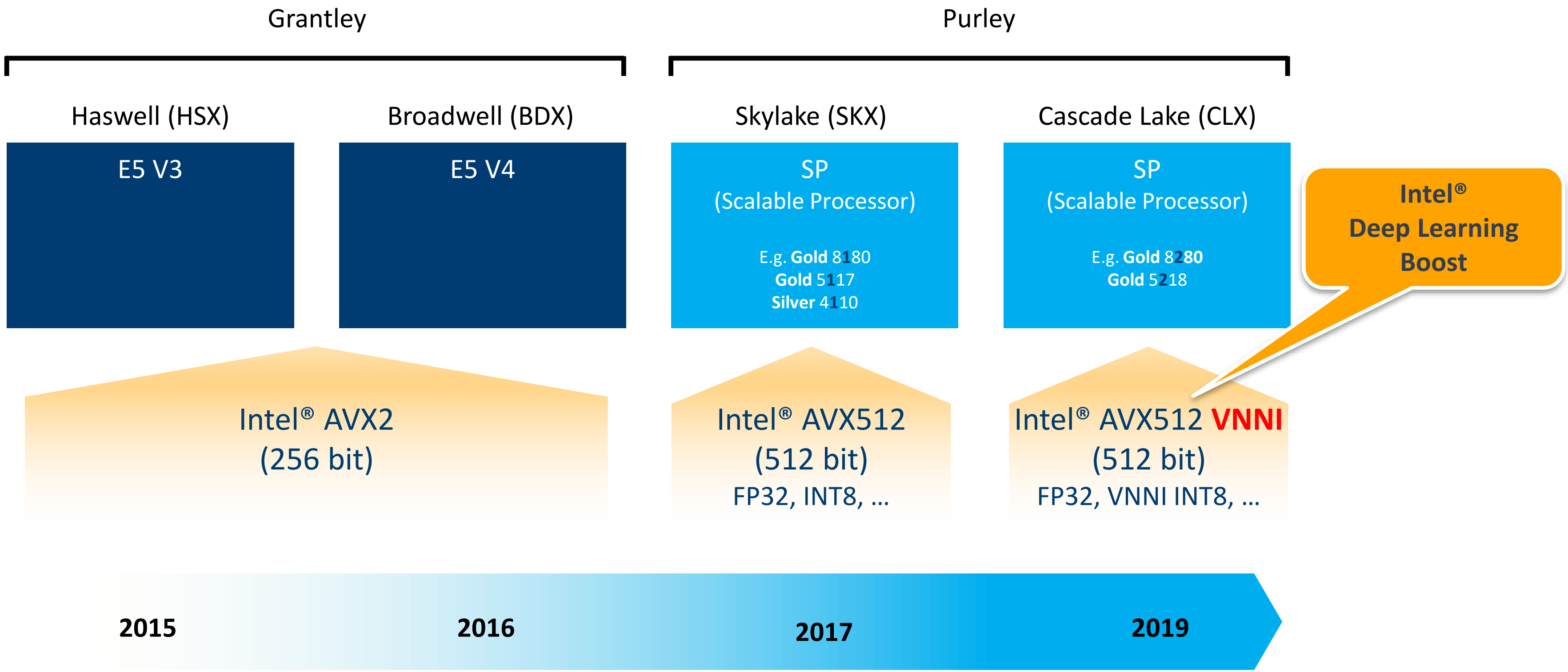
Second generation: Cascade Lake

INTEL® DEEP LEARNING BOOST

Intel® AVX-512 Vector Neural Network Instructions (VNNI)

- Boost Deep Learning Inference with VNNI

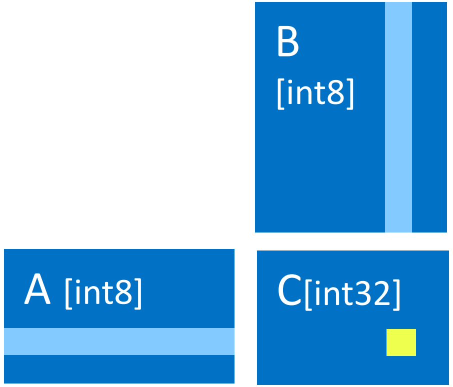
FAST EVOLUTION OF AI CAPABILITY ON INTEL® XEON® PLATFORM



- ***Intel® Deep Learning Boost* is a new set of AVX-512 instructions *designed to deliver significant, more efficient Deep Learning (Inference) acceleration on second generation Intel® Xeon® Scalable processor (codename "Cascade Lake")***

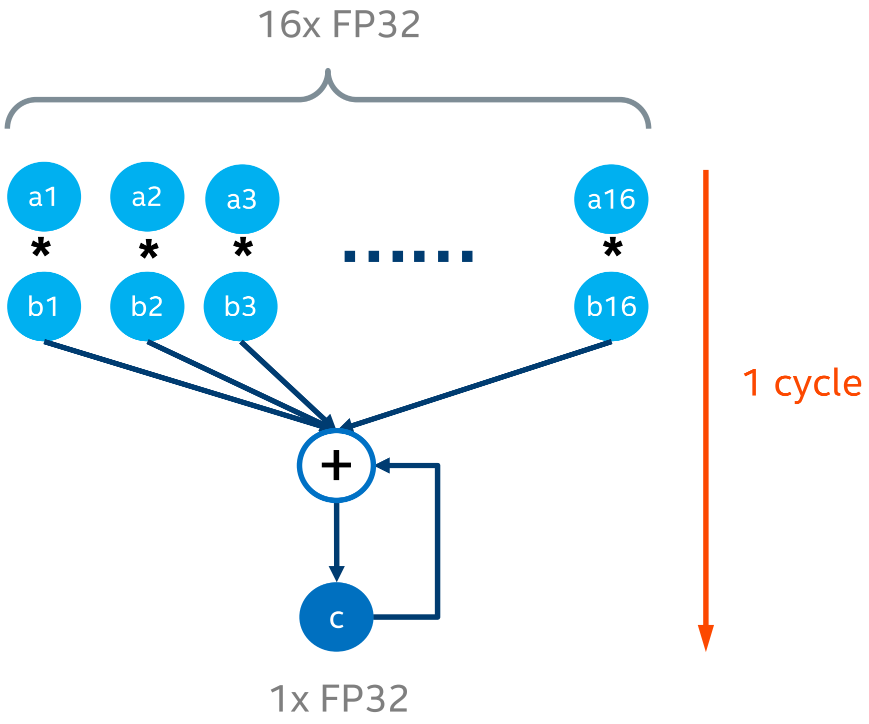
DEEP LEARNING FOUNDATIONS

- Matrix Multiplies are the foundation of many DL applications
  - **Multiply** a row*column values, **accumulate** into a single value
- Traditional HPC and many AI training workloads use floating point
  - Massive dynamic range of values (FP32 goes up to ~2^128)
- Why INT8 for Inference?
  - More power efficient per operation due to smaller multiplies
  - Reduces pressure on cache and memory subsystem
  - Precision and dynamic range sufficient for many models
- What's different about INT8?
  - Much smaller dynamic range than FP32: 256 values
    - Requires *accumulation into INT32* to avoid overflow
      (FP handles this "for free" w/ large dynamic range)



Matrix Multiply
A x B = C

FAST EVOLUTION OF AI CAPABILITY ON XEON PLATFORM



<u>For Example: Dual-Socket Xeon SP Gold 6148</u>

Total Peak FP32 TFLOPS

= (16 + 16) // 16 multiplies + 16 adds
x 2 // 2 FMA engines per core
x 20 // 20 physical cores per CPU
x 2 // 2 CPUs per server
x 0.0022 // 2.2 GHz AVX512 @20 cores active
= ***5.6 TFLOPS***

NUMERIC PRECISION

- Precision is a measure of the detail used for numerical values primarily measured in bits

				Signed Range
FP32	s	8 bit exp	23 bit mantissa	$\pm1.18\times10^{-38}$ to $\pm3.4\times10^{38}$
BF16	s	8 bit exp	7 bit mantissa	$\pm1.18\times10^{-38}$ to $\pm3.4\times10^{38}$
FP16	s	5 bit exp	10 bit mantissa	$\pm6.10 \times 10^{-5}$ To $\pm$ 65504
INT16	s		15 bit mantissa	-32,768 to +32,767
INT8	s		7 bit mantissa	-128 to 127

DL Compute in High precision - High processing time & high accuracy in results

DL Compute in Lower precision - Faster results but potentially less accurate

Speed accuracy trade off

PRECISION FOR DEEP LEARNING

- Precision is a measure of the detail used for numerical values primarily measured in bits

FP32 TRAINING → BF16 BIT TRAINING

Better cache usage
Avoids bandwidth bottlenecks
Maximizes compute resources
Lesser silicon area & power

FP32 INFERENCE → 8 BIT INFERENCE

NEED - FP32 WEIGHTS → INT 8 WEIGHTS FOR INFERENCE

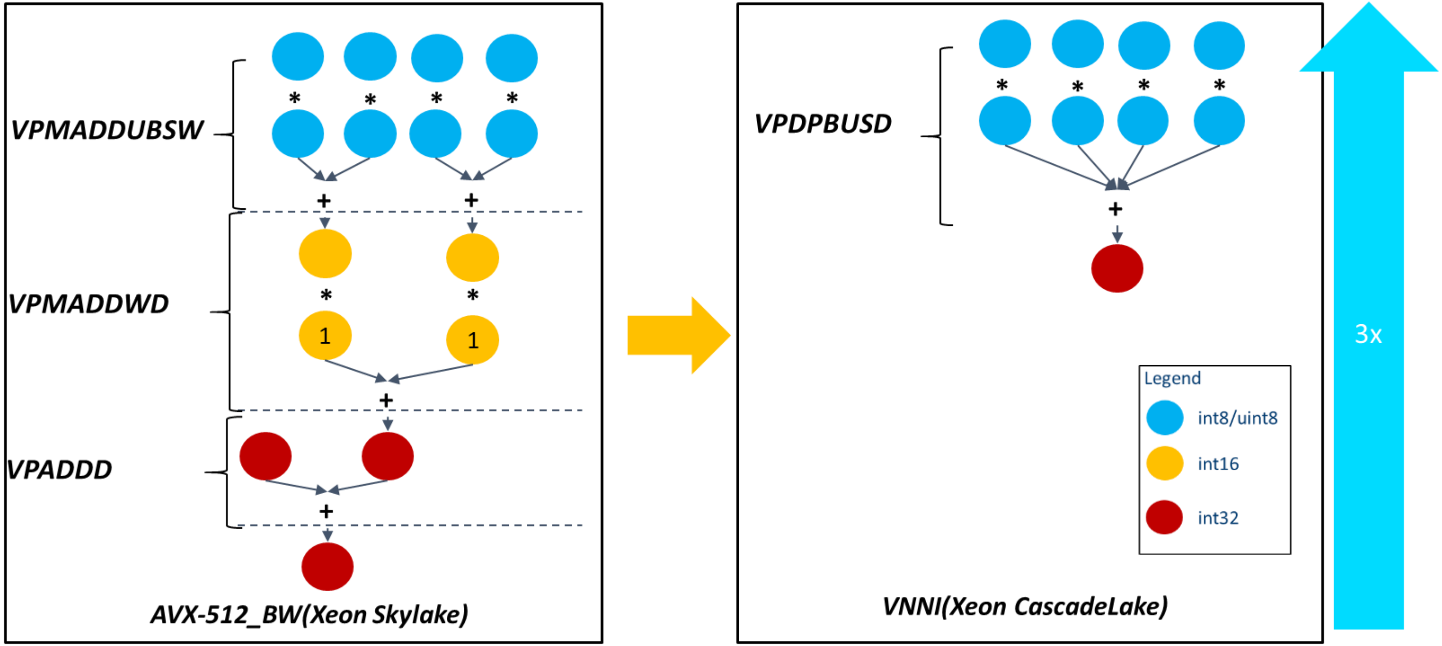
BENEFIT - REDUCE MODEL SIZE AND ENERGY CONSUMPTION

CHALLENGE - POSSIBLE DEGRADATION IN PREDICTIVE PERFORMANCE

SOLUTION - QUANTIZATION WITH MINIMAL LOSS OF INFORMATION

DEPLOYMENT - OPENVINO TOOLKIT OR DIRECT FRAMEWORK (TF)

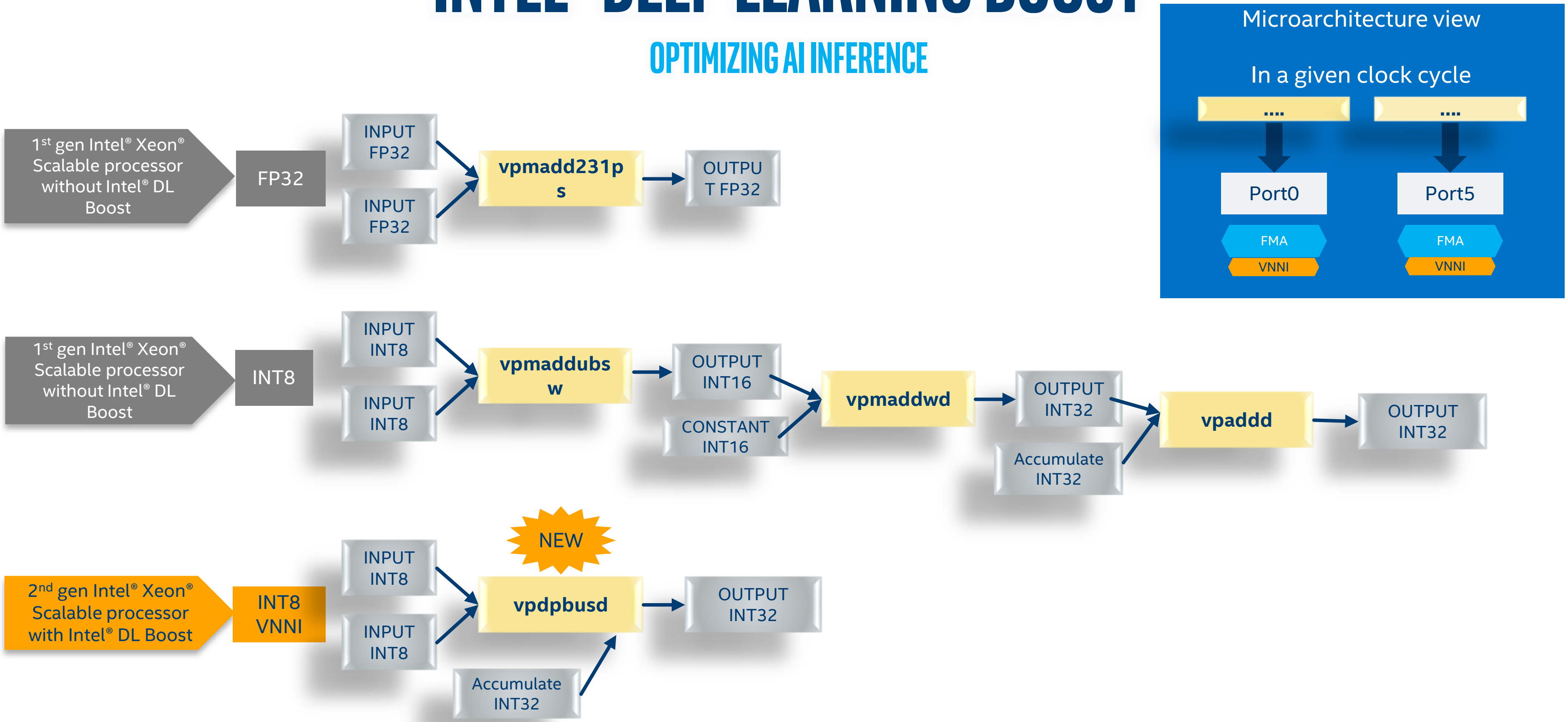
INSTRUCTIONS FUSION



VPMADDUBSW + VPMADDWD + VPADDD
fused into
VPDPBUSD (3x peak OPs)

INTEL® DEEP LEARNING BOOST

OPTIMIZING AI INFERENCE



[Optimization Notice](#)

SO WHAT?

AVX512_VNNI is a new set of AVX-512 instructions to boost Deep Learning performance

- VNNI includes FMA instructions for:
  - 8-bit multiplies with 32-bit accumulates (u8 x s8 ⇒ s32)
  - 16-bit multiplies with 32-bit accumulates (s16 x s16 ⇒ s32)
- Theoretical peak compute gains are:
  - 4x int8 OPS over fp32 OPS and ¼ memory requirements
  - 2x int16 OPS over fp32 OPS and ½ memory requirements
- Ice Lake and future microarchitectures will have AVX512_VNNI

ENABLING INTEL® DL BOOST ON CASCADE LAKE

THEORETICAL IMPROVEMENTS: FP32 VS. INT8 & DL BOOST

UP TO 4X BOOST IN MAC/CYCLE

UP TO 4X IMPROVED PERFORMANCE / WATT

DECREASED MEMORY BANDWIDTH

IMPROVED CACHE PERFORMANCE

UP NEXT: MICROBENCHMARKING WITH INTEL® MKL-DNN'S

Workloads

Topologies

TensorFlow mxnet Caffe2 Caffe PaddlePaddle

Frameworks

Intel® MKL-DNN Libraries

Intel® Processors

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.. 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.

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
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

Configuration details – Optimized Deep learning frameworks and tool kits

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

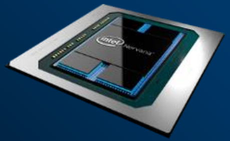
3.9x and 1.8x performance boost with TensorFlow ResNet-50:Tested by Intel as of 3/1/2019. 2 socket Intel® Xeon® Platinum 8280 Processor, 28 cores HT On Turbo ON Total Memory 384 GB (12 slots/ 32GB/ 2933 MHz), BIOS: SE5C620.86B.0D.01.0271.120720180605 (ucode:0x4000013),CentOS 7.6, 4.19.5-1.el7.elrepo.x86_64, 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

ONEAPI

INTEL DATA-CENTRIC HARDWARE: HIGH PERFORMANCE, FLEXIBLE OPTIONS

General Purpose CPU	Programmable Data Parallel Accelerator	FPGA	Domain Optimized Accelerator
intel XEON inside	Intel® Processor Graphics & Future Products		 Intel Neural Network Processor

GENERAL PURPOSE — HARDWARE — WORKLOAD OPTIMIZED

Provide optimal performance over the widest variety of workloads

Deliver highest performance per $/Watt/U/Rack for critical applications

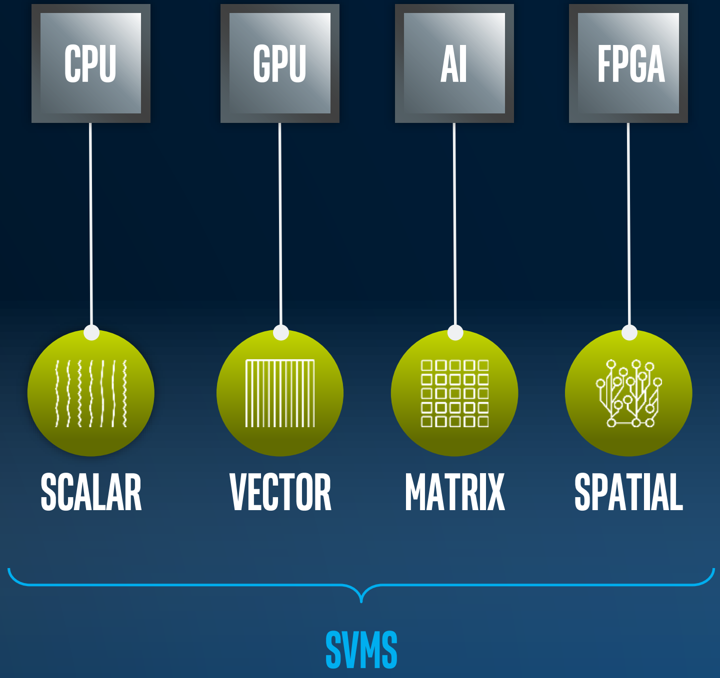
PROGRAMMING CHALLENGE

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 ONEAPI CORE CONCEPT

oneAPI is a project to deliver a unified programming model to simplify development across diverse architectures

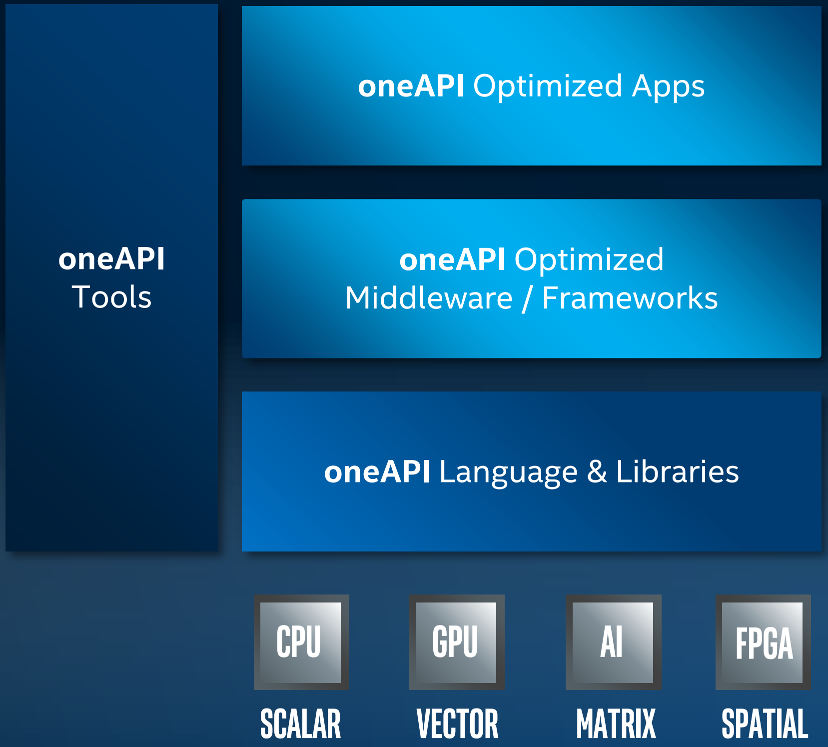
Common developer experience across Scalar, Vector, Matrix and Spatial (SVMS) architecture

Unified and simplified language and libraries for expressing parallelism

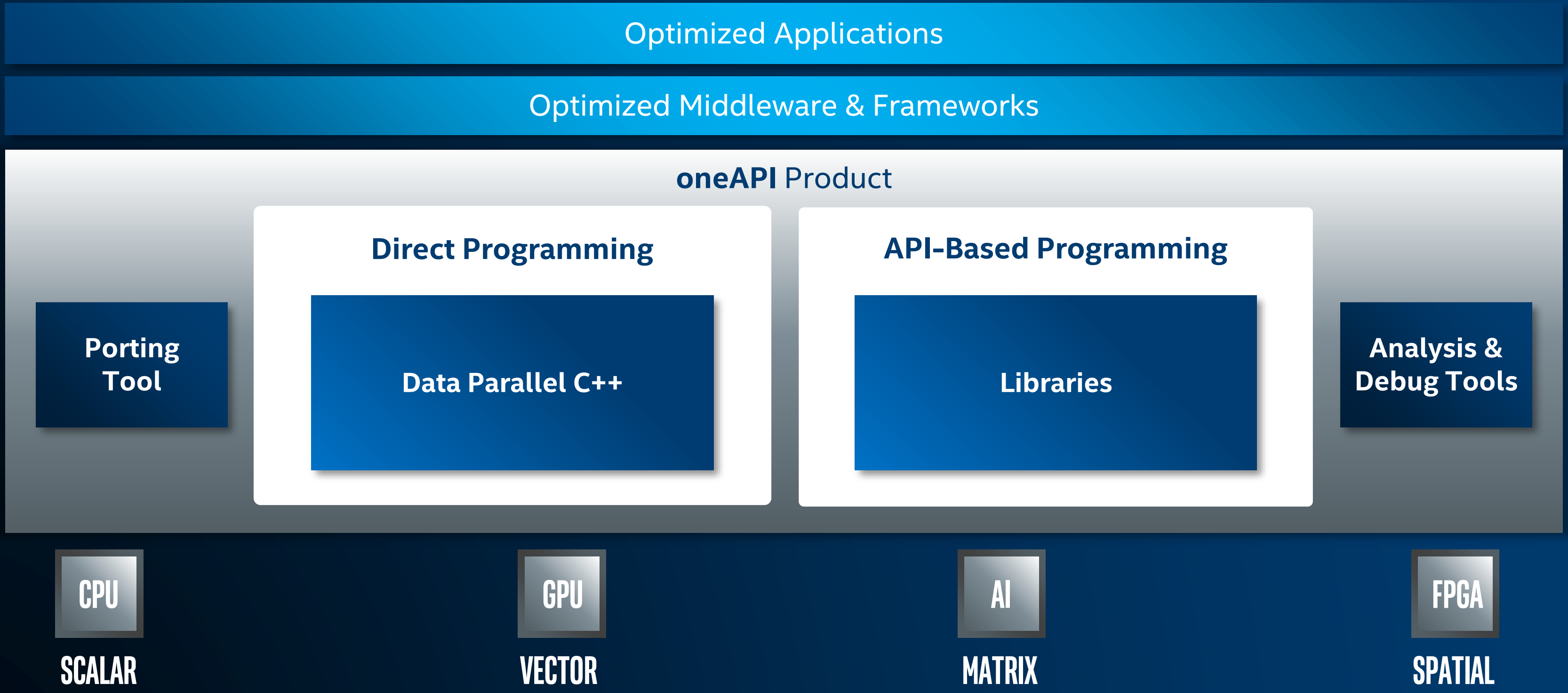
Uncompromised native high-level language performance

Support for CPU, GPU, AI and FPGA

Based on industry standards and open specifications



ONEAPI FOR CROSS-ARCHITECTURE PERFORMANCE



Some capabilities may differ per architecture.

Optimization Notice

Legal Disclaimer & 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 www.intel.com/benchmarks.

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

BACKUP

QUANTIZATION

FP32 WEIGHTS TRAINING → 8 BIT INFERENCE

REDUCE PRECISION & KEEP MODEL ACCURACY

POST TRAINING QUANTIZATION

Train normally, capture FP32 weights; convert to low precision before running inference and calibrate to improve accuracy

FP32 WEIGHTS TRAINING → Collect statistics for the activation to find quantization factor

Calibrate using subset data to improve accuracy, Convert FP32 weights to INT8 | 8 BIT INFERENCE

QUANTIZATION AWARE TRAINING

Simulate the effect of quantization in the forward and backward passes using FAKE quantization

FP32 WEIGHTS TRAINING | captured FP32 weights are quantized to int8 at each iteration after the weight updates → Convert FP32 weights to INT8 | 8 BIT INFERENCE

DATA PARALLEL C++

STANDARDS-BASED, CROSS-ARCHITECTURE LANGUAGE

Language to deliver uncompromised parallel programming productivity and performance across CPUs and accelerators

- Allows code reuse across hardware targets, while permitting custom tuning for a specific accelerator

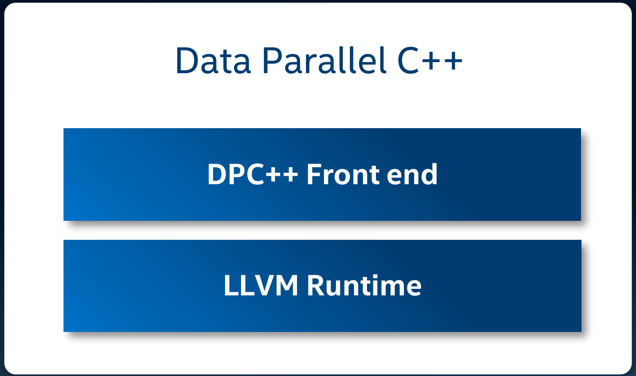
Based on C++

- Delivers C++ productivity benefits, using common and familiar C and C++ constructs
- Incorporates SYCL* from the Khronos* Group to support data parallelism and heterogeneous programming

Language enhancements being driven through community project

- Extensions to simplify data parallel programming
- Open and cooperative development for continued evolution

Builds upon Intel’s years of experience in architecture and compilers



POWERFUL LIBRARIES FOR DATA-CENTRIC FUNCTIONS

Key domain-specific functions to accelerate compute intensive workloads

Custom-coded for uncompromised performance on SVMS *(Scalar, Vector, Matrix, Spatial)* architectures

API-Based Programming

- Math
- Threading
- DPC++ Library
- Analytics/ML
- DNN
- ML Comm
- Video Processing
- Rendering

ADVANCED ANALYSIS & DEBUG TOOLS

Productive performance analysis across SVMS architectures

Intel® VTune™ Profiler

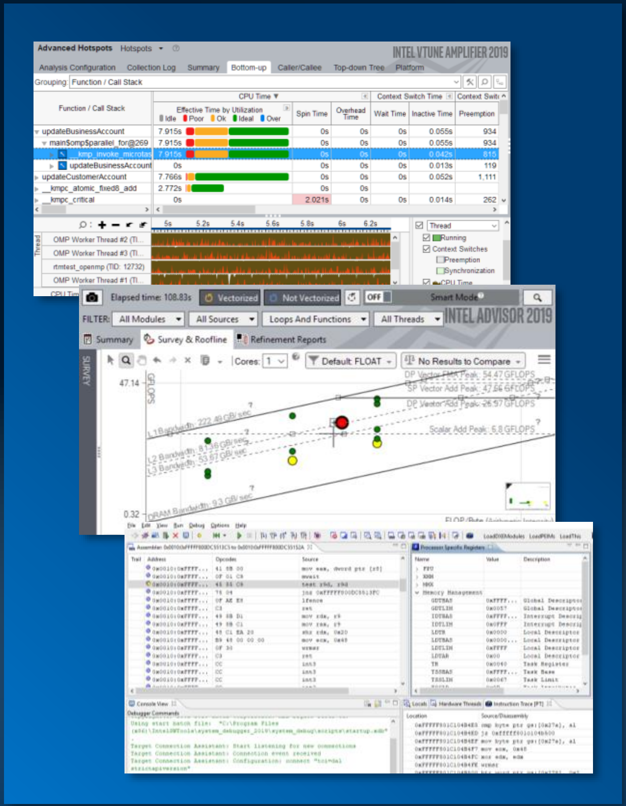
Profiler to analyze CPU and accelerator performance of compute, threading, memory, storage, and more

Intel® Advisor

Design assistant to provide advice on offload, threading, and vectorization

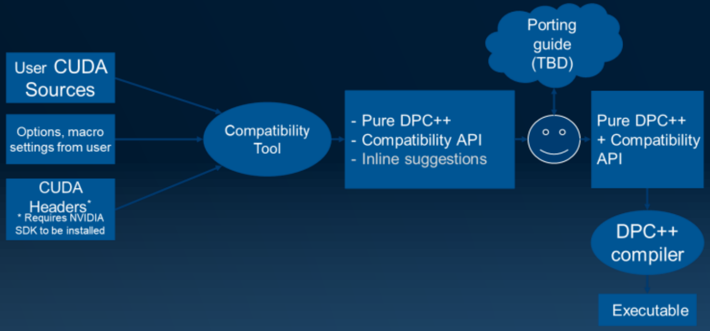
Debugger

Application debugger for fast code debug on CPUs and accelerators



COMPATIBILITY TOOL

Facilitate addressing multiple hardware choices through a modern language like DPC++



Software