TECHNCIAL COMPUTING / HPC SOFTWARE TOOLS & AI SOLUTIONS

Accelerate AI, HPC, Enterprise & Cloud Applications

April 2019
@ CiTIUS: Centro Singular de Investigación en Tecnoloxías da Información

Intel Computing Performance and Software Products (CPDP)
Edmund Preiss

Agenda

- **Intel Software Development Tools**

- Intel optimized AI Solutions

Intel® Software Developer Tools & SDKs

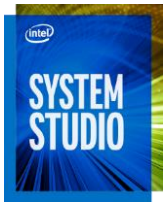
Intel® Parallel Studio XE

Comprehensive Enterprise , HPC Tools suite

Shared and distributed memory systems

Code creation and versatile SW Analysis Tools

Intel® System Studio



Embedded Tools Suite

Comprehensive, all-in-one, cross-platform system & IoT development tool suite

Simplifies system bring-up, boosts performance and power efficiency, strengthens system reliability

Intel® Media Server Studio

Media Encode/Decode Tools

Media SDK

Graphics Perf Analyzer

Computer Vision SDK

Open CL SDK

Context SDK

OpenVINO™

Machine Learning / Deep Learning Inference

Computer Vision SDK

Deep Learning (DL) Deployment Toolkit

Deep Learning Algorithms

Optimized DL Frameworks

What's Inside Intel® Parallel Studio XE

Comprehensive Software Development Tool Suite

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

PROFESSIONAL EDITION

ANALYZE
Analysis Tools

Intel® VTune™ Amplifier
Performance Profiler

Intel® Inspector
Memory & Thread Debugger

Intel® Advisor
Vectorization Optimization & Thread Prototyping

CLUSTER EDITION

SCALE
Cluster Tools

Intel® MPI Library
Message Passing Interface Library

Intel® Trace Analyzer & Collector
MPI Tuning & Analysis

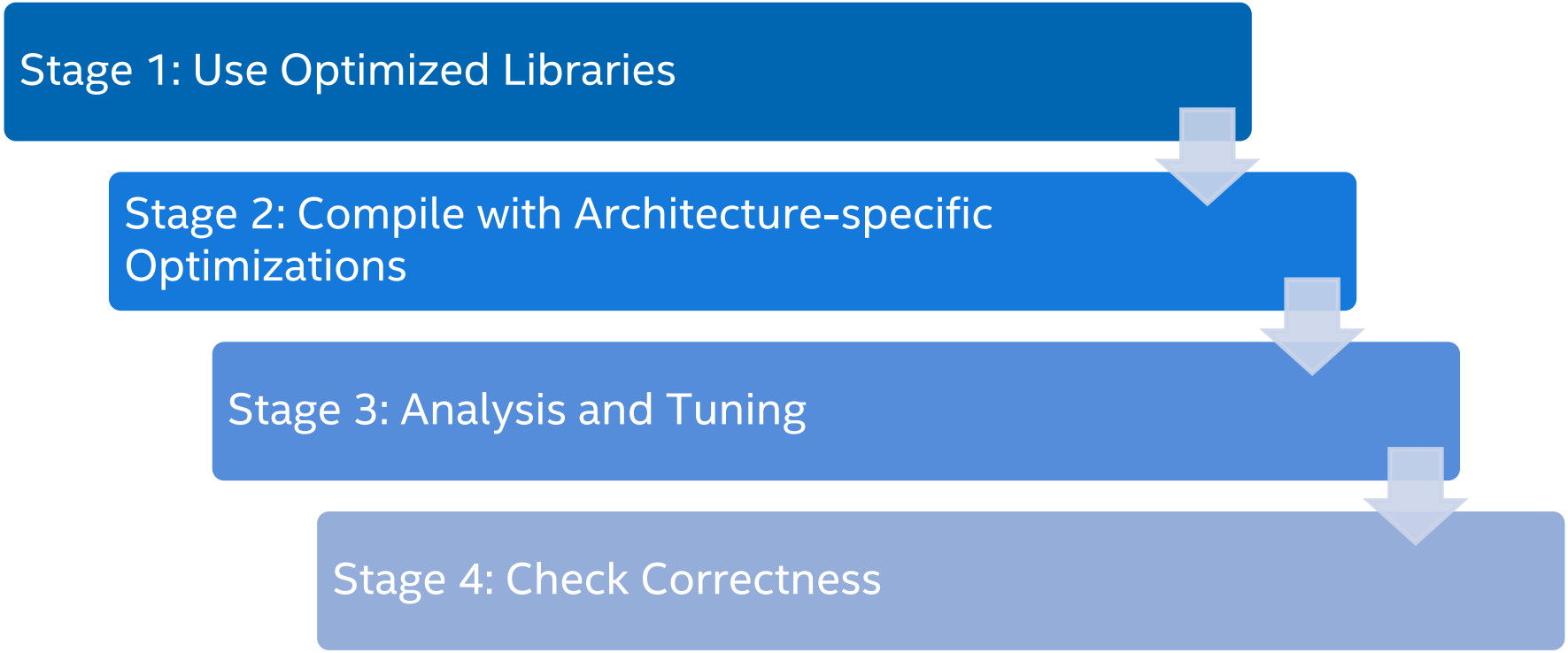
Intel® Cluster Checker
Cluster Diagnostic Expert System

Intel® Architecture Platforms

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

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

Code Modernization



INTEL® PARALLEL STUDIO XE TOOLS DETAILS

BUILD

Intel® C++ Compiler
Intel® Fortran Compiler
Intel® Distribution for Python*
Intel® Math Kernel Library
Intel® Integrated Performance Primitives
Intel® Threading Building Blocks
Intel® Data Analytics Acceleration Library
Included in Composer Edition

ANALYZE

Intel® VTune™ Amplifier
Intel® Advisor
Intel® Inspector

Part of the Professional Edition

SCALE

Intel® MPI Library
Intel® Trace Analyzer & Collector
Intel® Cluster Checker

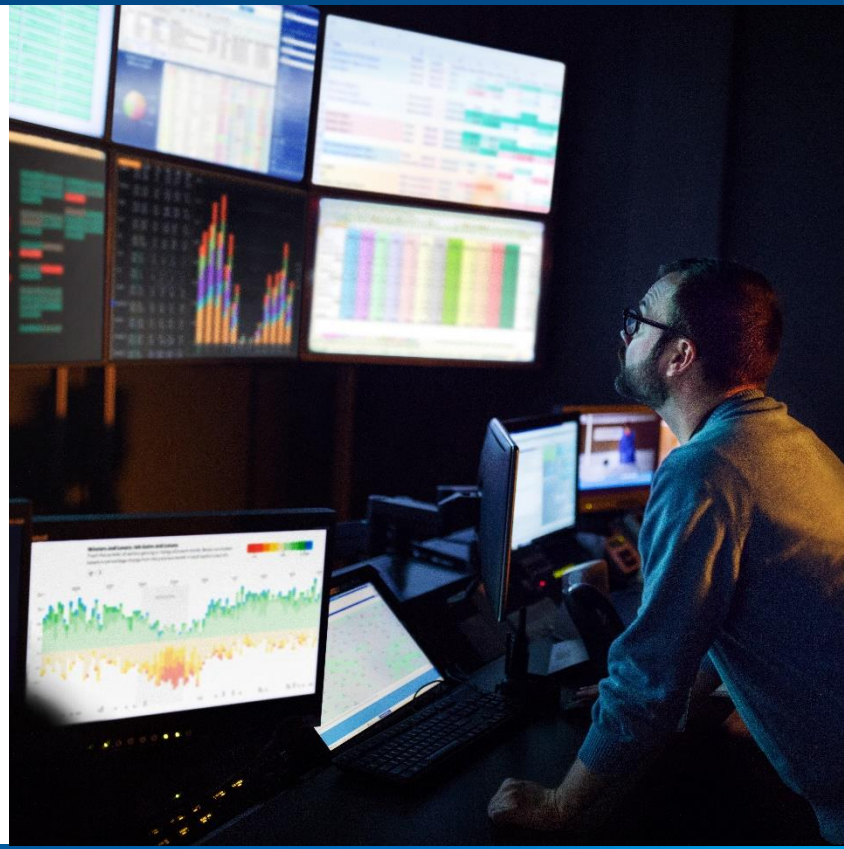
Part of the Cluster Edition

Fast, Scalable, Parallel Code with Intel® Compilers

Deliver Industry-leading C/C++ Code Performance, Unleash the Power of the latest Intel® Processors

- Develop optimized and vectorized code for Intel® architectures, including Intel Atom®, Intel® Core™ and Intel® Xeon® processors
- Achieve Superior Parallel Performance—Vectorize & thread your code (using OpenMP*) to take full advantage of the latest SIMD-enabled hardware, including Intel® Advanced Vector Extensions 512 (Intel® AVX-512)
- Leverage latest language and OpenMP* standards, and compatibility with leading compilers and IDEs

Learn More: software.intel.com/intel-compilers



What's New for Intel® C++ Compilers 19.0

Additional C++17 Standard feature support

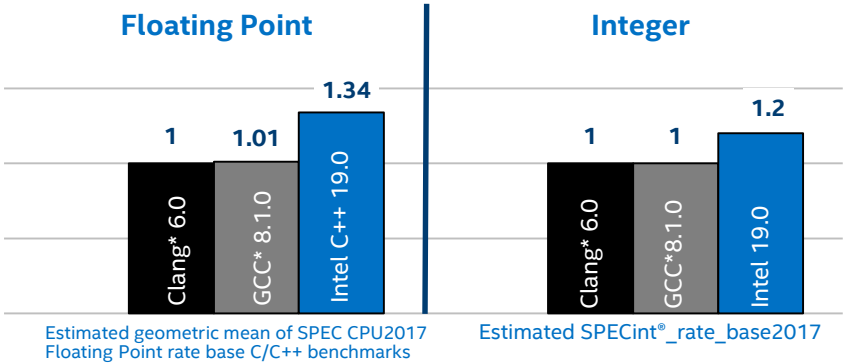
- Enjoy improvements to lambda and constant expression support
- Improved GNU C++ and Microsoft C++ compiler compatibility

Standards-driven parallelization for C++ developers

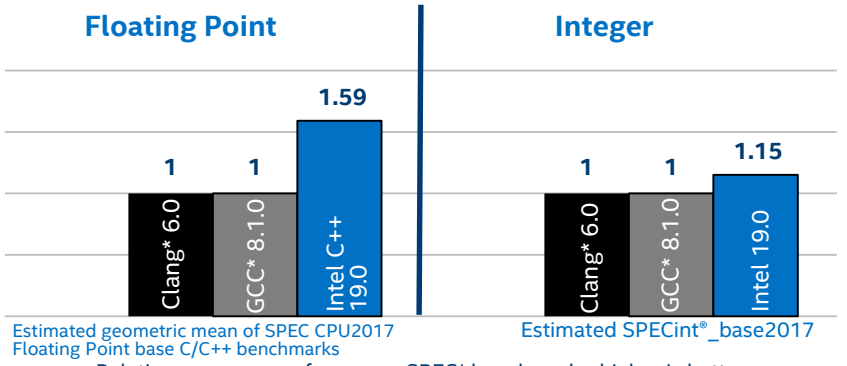
- Partial OpenMP Version 5.0 support
- Modernize your code by using the latest parallelization specifications

[1]OpenMP 5 is currently a draft

Systems Performance Advantage as Measured by SPEC* on Intel® Xeon™ Processors—Intel® C++ Compiler



Relative geomean performance, SPEC* benchmark - higher is better



Relative geomean performance, SPEC* benchmark - higher is better

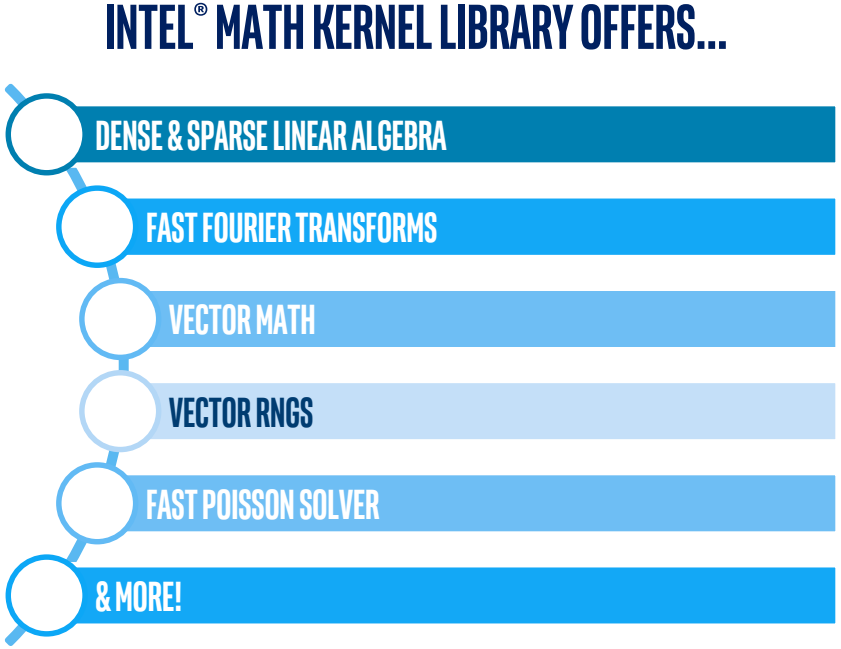
Performance results are based on testing as of Aug. 26, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. Software and workloads used in performance tests may have been optimized for performance only on Intel 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, see Performance Benchmark Test Disclosure.

Testing by Intel as of Aug. 26, 2018. Configuration: Linux hardware: Intel® Xeon® Platinum 8180 CPU @ 2.50GHz, 384 GB RAM, HyperThreading is on. Software: Intel compilers 19.0, GCC 8.1.0. PGI 18.5, Clang/LLVM 6.0. Linux OS: Red Hat Enterprise Linux Server release 7.4 (Maipo), 3.10.0-693.el7.x86_64. SPEC* Benchmark (www.spec.org). SmartHeap 10 was used for CXX tests when measuring SPECint® benchmarks.SPECint®_rate_base_2017 compiler switches: SmartHeap 10 were used for C++ tests. Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=3. GCC 8.1.0 -march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto. Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast -funroll-loops –flto.SPECfp®_rate_base_2017 compiler switches: Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-prefetch -ffinite-math-only -qopt-mem-layout-trans=3. GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto. Clang 6.0: -march=znver1 -mfpmath=sse -Ofast -funroll-loops –flto.SPECint®_speed_base_2017 compiler switches: SmartHeap 10 were used for C++ tests. Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=3 -qopenmp. GCC 8.1.0 '-march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp. Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp=libomp. SPECfp®_speed_base_2017 compiler switches: Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-prefetch -ffinite-math-only -qopenmp. GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto -fopenmp. Clang 6.0: -march=skylake-avx512 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp=libomp.

Testing by Intel as of Aug. 26, 2018. Configuration: Linux hardware: Intel® Xeon® Platinum 8180 CPU @ 2.50GHz, 384 GB RAM, HyperThreading is on. Software: Intel compilers 19.0, GCC 8.1.0. PGI 18.5, Clang/LLVM 6.0. Linux OS: Red Hat Enterprise Linux Server release 7.4 (Maipo), 3.10.0-693.el7.x86_64. SPEC* Benchmark (www.spec.org). SmartHeap 10 was used for CXX tests when measuring SPECint® benchmarks.SPECint®_rate_base_2017 compiler switches: SmartHeap 10 were used for C++ tests. Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=3. GCC 8.1.0 -march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto. Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast -funroll-loops –flto.SPECfp®_rate_base_2017 compiler switches: Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-prefetch -ffinite-math-only -qopt-mem-layout-trans=3. GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto. Clang 6.0: -march=znver1 -mfpmath=sse -Ofast -funroll-loops –flto.SPECint®_speed_base_2017 compiler switches: SmartHeap 10 were used for C++ tests. Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-mem-layout-trans=3 -qopenmp. GCC 8.1.0 '-march=znver1 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp. Clang 6.0: -march=core-avx2 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp=libomp. SPECfp®_speed_base_2017 compiler switches: Intel C/C++ compiler 19.0: -xCORE-AVX512 -ipo -O3 -no-prec-div -qopt-prefetch -ffinite-math-only -qopenmp. GCC 8.1.0: -march=skylake-avx512 -mfpmath=sse -Ofast -fno-associative-math -funroll-loops -flto -fopenmp. Clang 6.0: -march=skylake-avx512 -mfpmath=sse -Ofast -funroll-loops -flto -fopenmp=libomp.

Fast, Scalable Code with Intel® Math Kernel Library (Intel® MKL)

- Speeds computations for machine learning applications through highly-optimized, threaded & vectorized math functions
- Provides key functionality for dense & sparse linear algebra (BLAS, LAPACK, PARDISO), FFTs, vector math, summary statistics, splines & more
- Dispatches optimized code for each processor automatically without the need to branch code
- Optimized for single core vectorization & cache utilization
- Great performance with minimal effort!



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

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

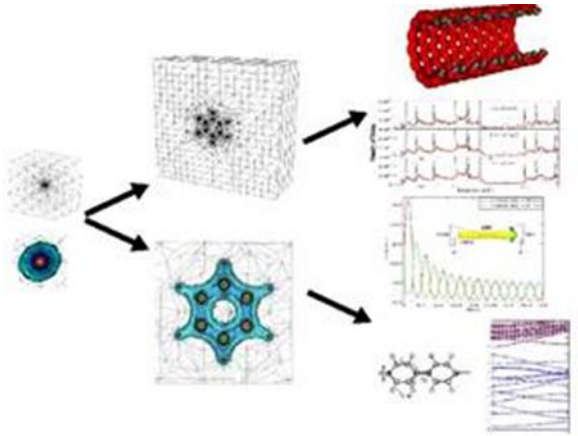
- Splines
- Interpolation
- Trust Region
- Fast Poisson Solver

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

What's New for Intel® Math Kernel Library 2019?

Just-In-Time Fast Small Matrix Multiplication

- Improved speed of S/DGEMM for Intel® Advanced Vector Extensions (Intel® AVX-512, Intel® AVX2) and with JIT capabilities



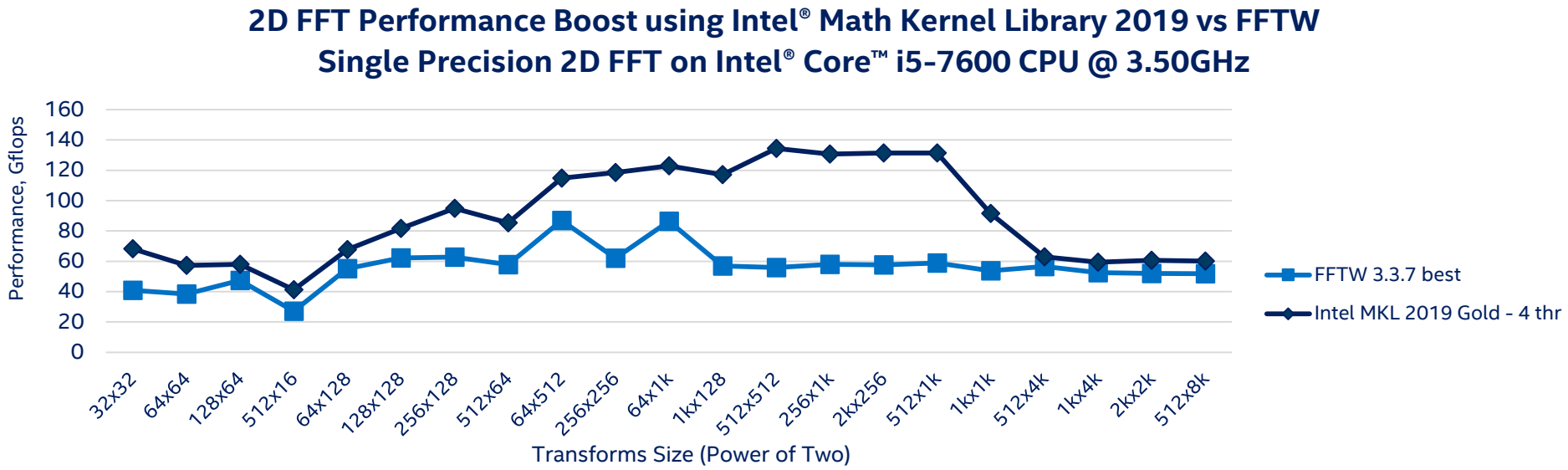
Sparse QR Solvers

- Solve sparse linear systems, sparse linear least squares problems, eigenvalue problems, rank and null-space determination, and others

Generate Random Numbers for Multinomial Experiments

- Highly-optimized multinomial random number generator for finance, geological and biological applications

FFT Performance Boost on Intel® Core™ Processor Intel® Math Kernel Library



Performance results are based on testing as of July 12, 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, see Performance Benchmark Test Disclosure.
Testing by Intel as of July 12, 2018. **Configuration:** Intel® Core ™ i5-7600, 1x4 cores, 3.50GHz, 6MB CPU Cache, 64GB RAM, OS RHEL 7.2
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.

Speed-up Analytics at the Edge with Intel® Data Analytics Acceleration Library (Intel® DAAL)

Boost Machine Learning & Data Analytics Performance

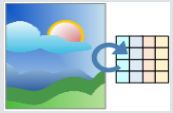
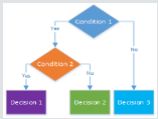
- Helps applications deliver better predictions faster
- Optimizes data ingestion & algorithmic compute together for highest performance
- Supports offline, streaming & distributed usage models to meet a range of application needs
- Split analytics workloads between edge devices and cloud to optimize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

Learn More: software.intel.com/daal

What's New in the 2019 Release

New Algorithms

- **High performance Logistic Regression**, most widely-used classification algorithm
- **Extended Gradient Boosting Functionality** provides inexact split calculations & algorithm-level computation canceling by user-defined callback for greater flexibility
- **User-defined Data Modification Procedure in CSV & IDBC data sources to implement** a wide range of feature extraction & transformation techniques

Pre-processing	Transformation	Analysis	Modeling	Validation	Decision Making
					
Decompression, Filtering, Normalization	Aggregation, Dimension Reduction	Summary Statistics Clustering, etc.	Machine Learning (Training) Parameter Estimation Simulation	Hypothesis Testing Model Errors	Forecasting Decision Trees, etc.

Speed Imaging, Vision, Signal, Security & Storage Apps with Intel® Integrated Performance Primitives (Intel® IPP)

Accelerate Image, Signal, Data Processing & Cryptography Computation Tasks

- Multi-core, multi-OS and multi-platform ready, computationally intensive & highly optimized functions
- Use high performance, easy-to-use, production-ready APIs to quickly improve application performance
- Reduce cost & time-to-market on software development & maintenance

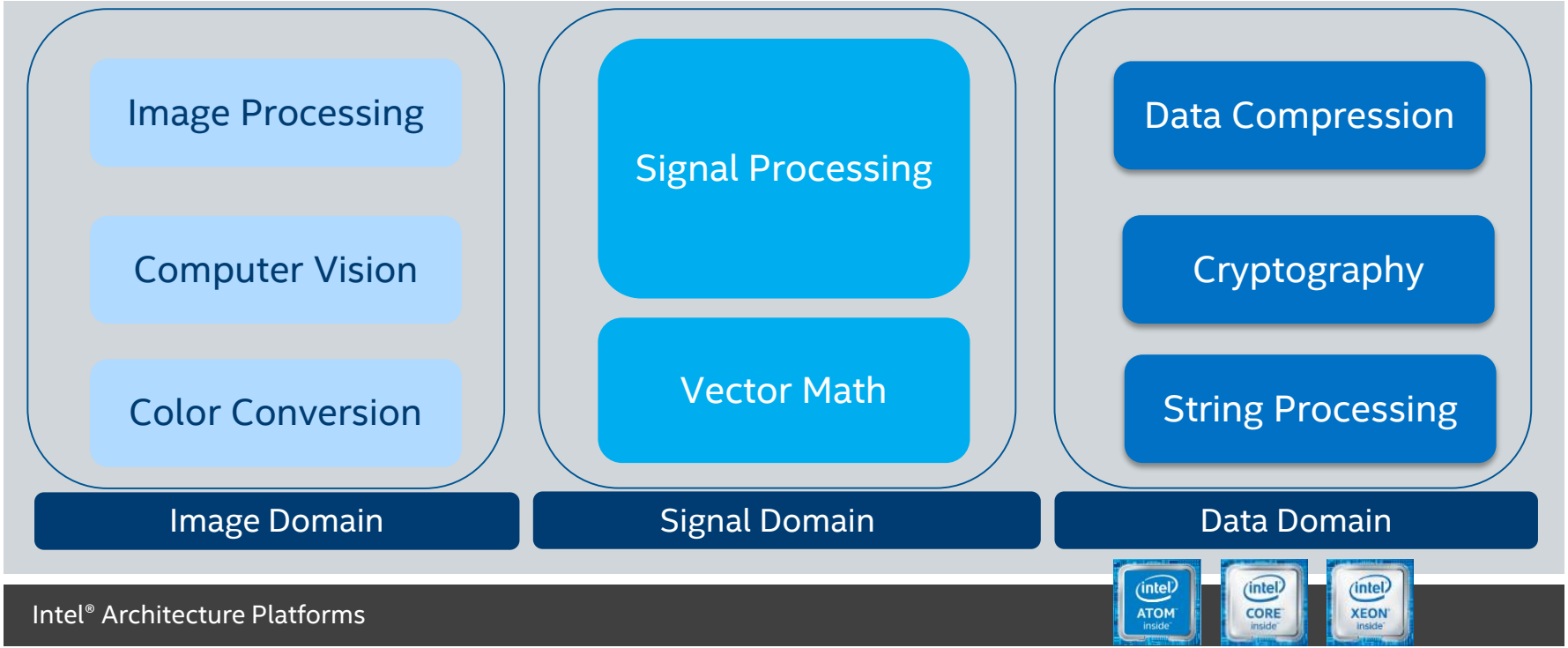
What's New in 2019 Release

- Improved LZ4 compression & decompression performance on high entropy data
- New color conversion functions for convert RBG images to CIE Lab color models, & vice versa
- Open source distribution of Intel® IPP Cryptography Library
- Extended optimization for Intel® Advanced Vector Extensions (Intel®-512 & Intel® AVX2)s
- Added Threading Layer with OpenMP* and Threading Building Blocks support for various image processing functions

Learn More: software.intel.com/intel-ipp

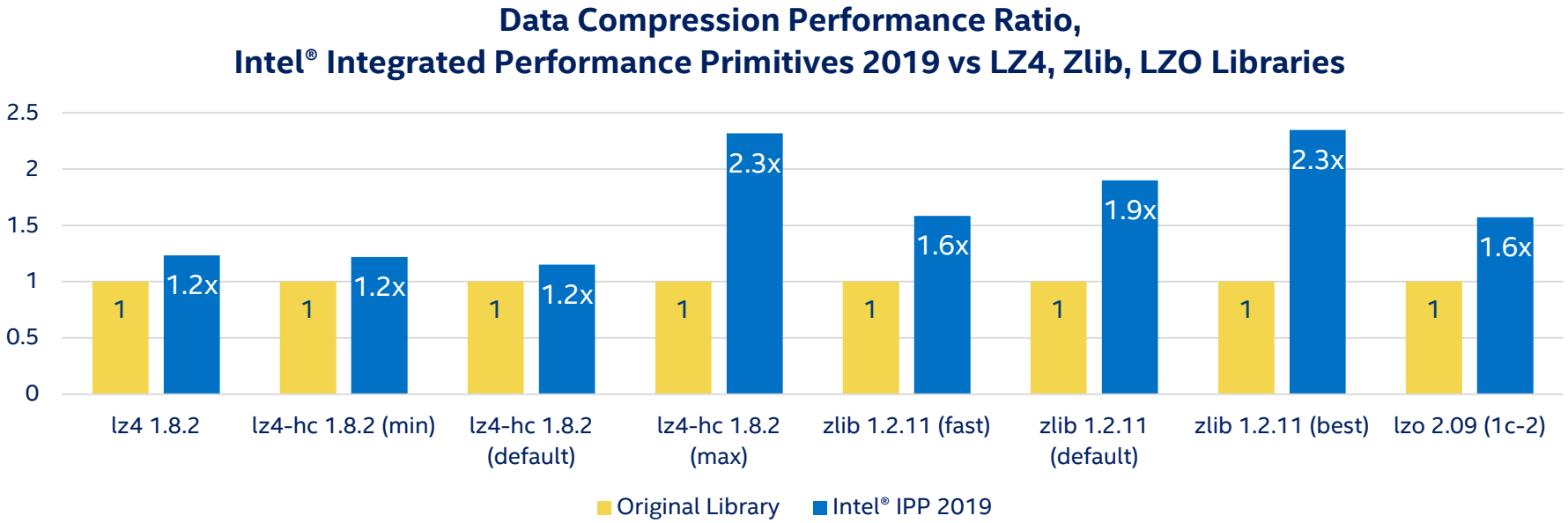
What's Inside Intel® Integrated Performance Primitives

High Performance, Easy-to-Use & Production Ready APIs



Performance Improvement for Data Compression

Intel® Integrated Performance Primitives



Performance results are based on testing as of Aug. 15, 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, see Performance Benchmark Test Disclosure.
Testing by Intel as of August 15, 2018. Configuration: Intel® Core™ i5-7600 CPU @3.50GHz, 4 cores, hyper-threading off; Cache: L1=32KB, L2=256KB, L3=6MB; Memory: 64GB; OS: RH EL Server 7.2
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.

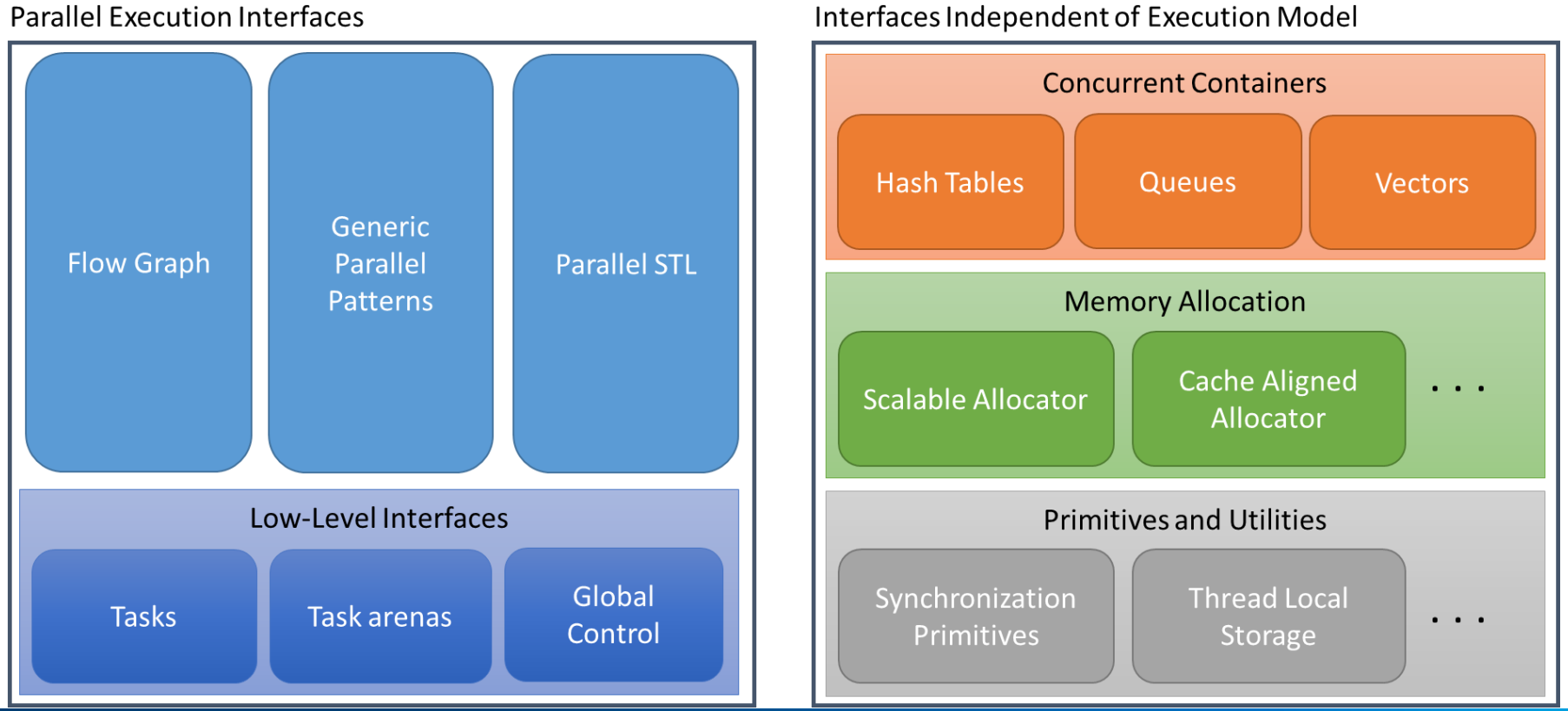
Get the Benefits of Advanced Threading with Threading Building Blocks

Use Threading Techniques to Leverage Multicore Performance & Heterogeneous Computing for C++

- Parallelize computationally intensive work across CPUs & GPUs—deliver higher-level & simpler solutions using C++
- Most feature-rich & comprehensive solution for parallel programming
- Highly portable, composable, affordable, approachable, future-proof scalability

Learn More: software.intel.com/intel-tbb

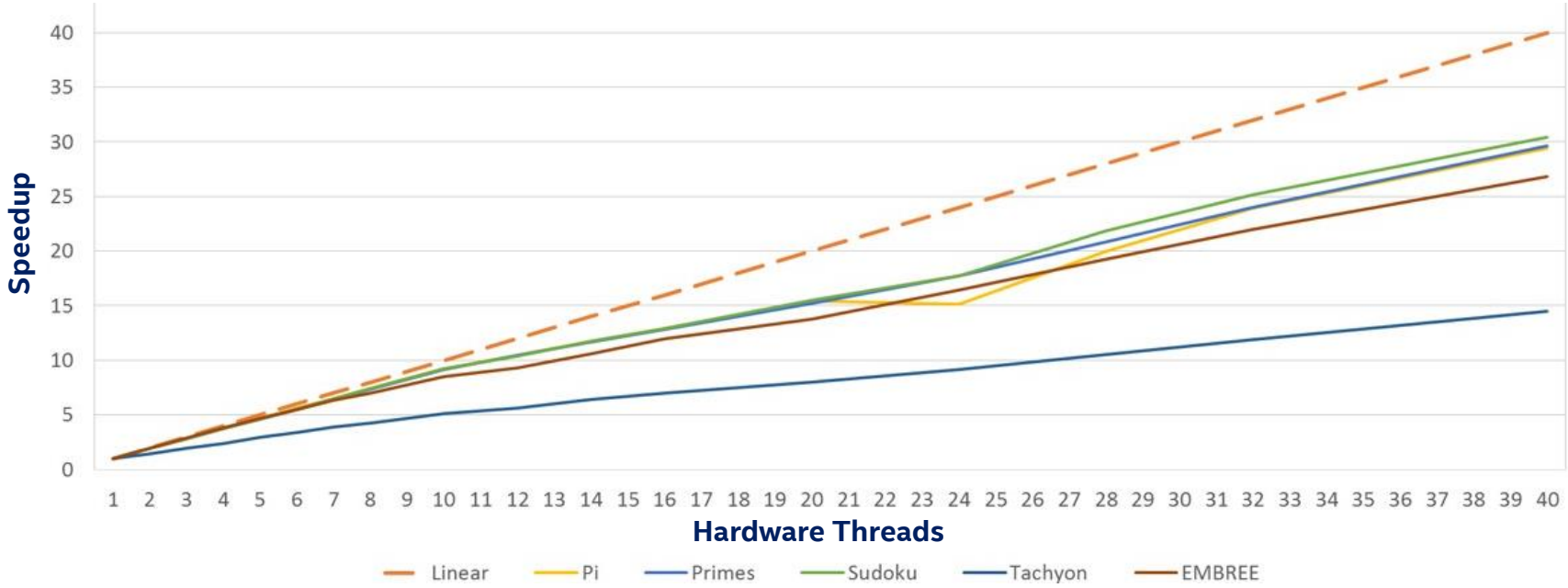
What's Inside Threading Building Blocks



Advantages of Using Threading Building Blocks over other Threading Models

- **Specify tasks** instead of manipulating threads. Threading Building Blocks (TBB) maps your logical tasks onto threads with full support for nested parallelism
- TBB uses **proven , efficient parallel patterns**.
- TBB uses work stealing to support the load balance of unknown execution time for tasks. This has the advantage of **low-overhead polymorphism**.
- **Flow graph feature** in TBB allows developers to easily express dependency and data flow graphs.
- Has **high level parallel algorithms**, concurrent containers, and low level building blocks like scalable memory allocator, locks and atomic operations.

Excellent Performance Scalability with Threading Building Blocks on Intel® Xeon® Processor



Performance results are based on testing as of July 31, 2018 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. 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 July 31, 2018. Software versions: Intel® C++ Intel® 64 Compiler, Version 18.0, Threading Building Blocks (TBB) 2019; Hardware: 2x Intel® Xeon® Gold 6152 CPU @ 2.10GHz, 192GB Main Memory; Operating System: CentOS Linux* release 7.4 1708 (Core), kernel 3.10.0-693.e17.x86_64; Note: sudoku, primes and tachyon are included with TBB.

INTEL® PARALLEL STUDIO XE COMPONENT TOOLS

BUILD

Intel® C++ Compiler
Intel® Fortran Compiler
Intel® Distribution for Python*
Intel® Math Kernel Library
Intel® Integrated Performance Primitives
Intel® Threading Building Blocks
Intel® Data Analytics Acceleration Library
Included in Composer Edition

ANALYZE

Intel® VTune™ Amplifier
Intel® Advisor
Intel® Inspector

Part of the Professional Edition

SCALE

Intel® MPI Library
Intel® Trace Analyzer & Collector
Intel® Cluster Checker

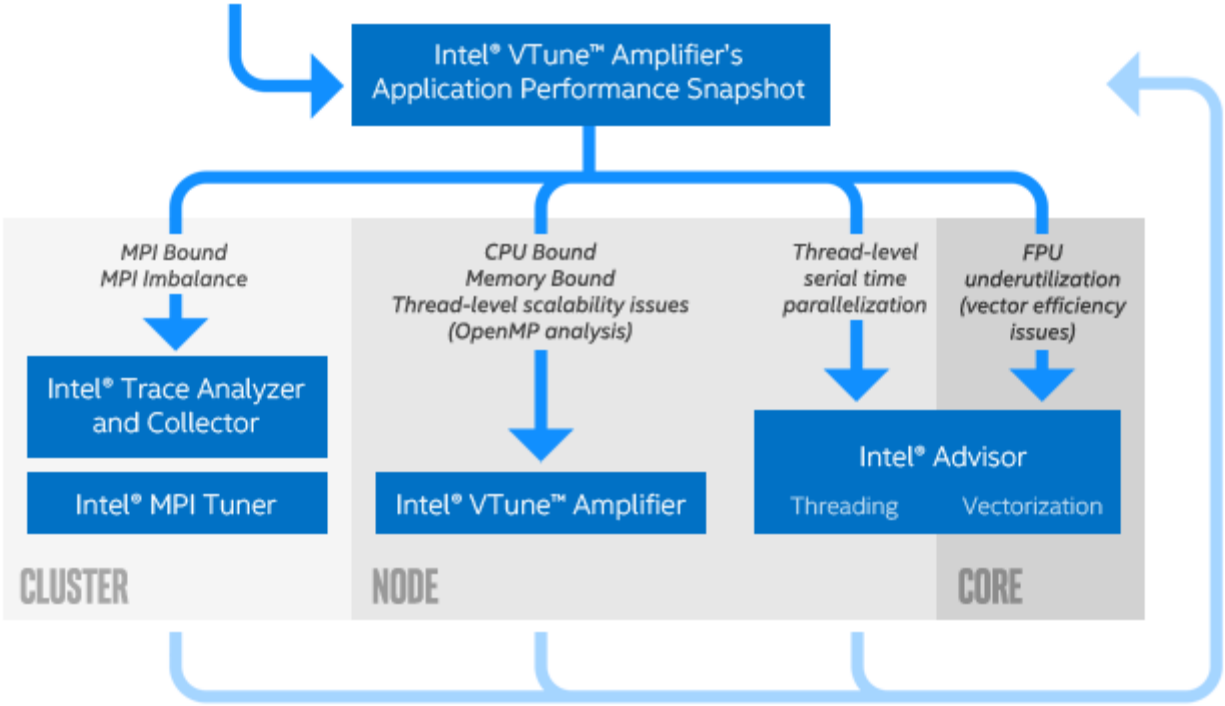
Part of the Cluster Edition

Before diving Into a particular tool …

- How to assess that I have **potential in performance** tuning?
- **Which tool** should I use first?
- What to use on **large scale** avoiding being overwhelmed with huge trace size, post processing time and collection overhead?
- How to **quickly** evaluate environment settings or incremental code changes?
- **Answer:**

  Use VTune Amplifier's Application Performance Snapshot

Performance Optimization Workflow based on APS



Better, Faster Application Performance Snapshot

Intel® VTune™ Amplifier

Better Answers

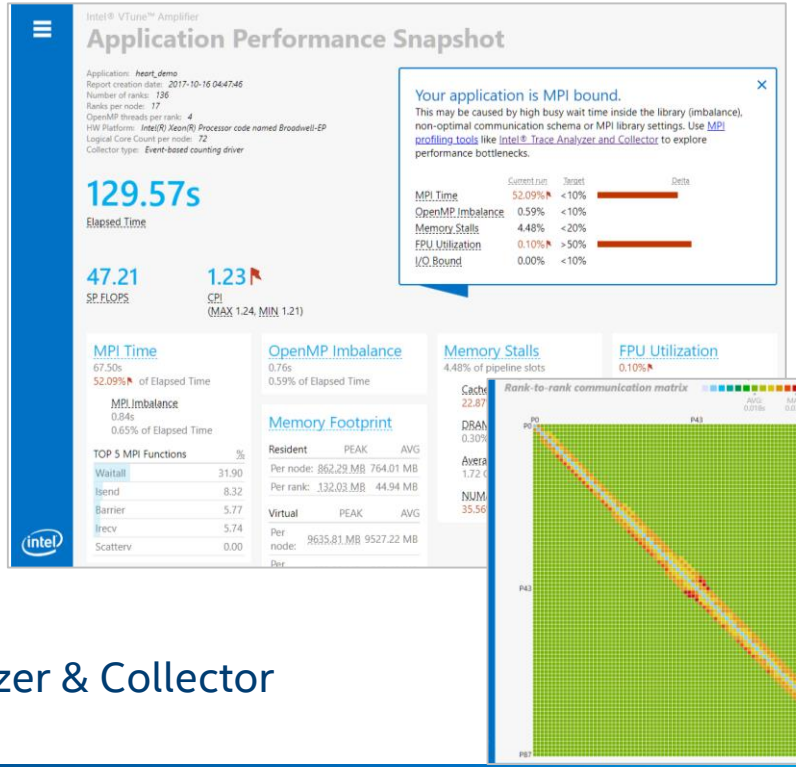
- CPU utilization analysis of physical cores

Less Overhead

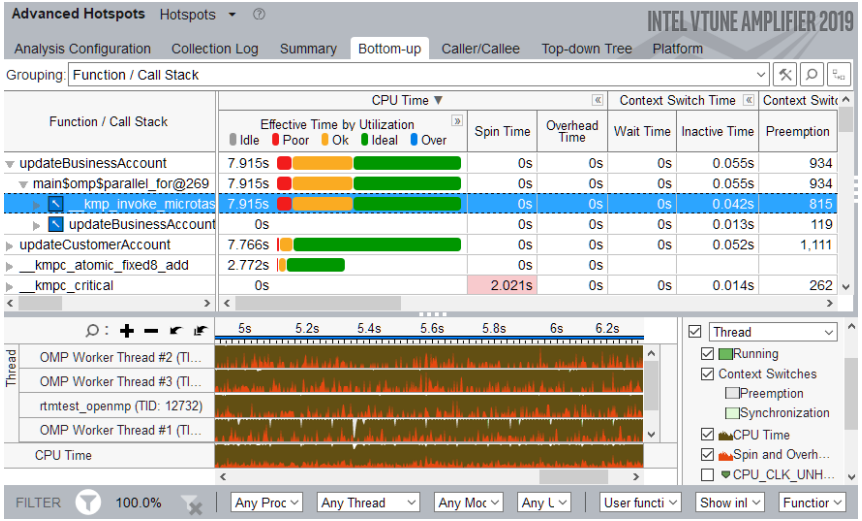
- Lower MPI trace overhead & faster result processing
- New data selection & pause/resume let you focus on useful data

Easier to Use

- Visualize rank-to-rank & node-to-node MPI communications
- Easily configure profiling for Intel® Trace Analyzer & Collector



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



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

Intel® VTune™ Amplifier—What's New in 2019 Details

Performance Profiler

Easier to Use

- New workflow provides easier-to-learn tuning workflow and a simplified setup
- New visualization simplifies general exploration complex counter information

I/O Analysis—Tune SPDK storage and DPDK network performance

- Measure "empty" polling cycles where no real work is done

Platform Profiler

- Longer data collection finds hardware configuration issues and poorly tuned applications

Tune CPU/GPU Rendering—GPU Rendering Analysis (Linux*, Android* only)

- Detects performance-critical API calls of OpenGL-ES applications
- Finds virtual Xen* hypervisor domains that bottleneck the system

Balance CPU Loading—CPU Analysis

- Balance CPU loading for better performance
- Assess memory transfer time, CPU balance, CPU context switches, FPU utilization, workload wait times, and more

Tune Workloads & System Configuration

Intel® VTune Amplifier

Finds

- Configuration issues
- Poorly tuned software

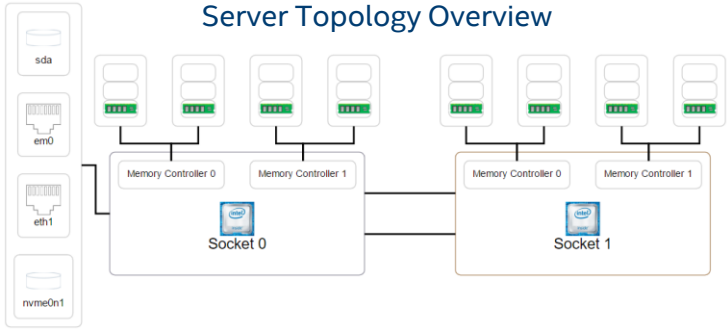
Target Users

- Infrastructure Architects
- Software Architects & QA

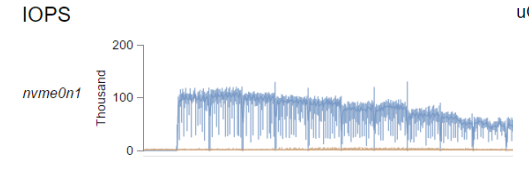
Performance Metrics

- Extended capture (minutes to hours)
- Low overhead – coarse grain metrics
- Sampling OS & hardware performance counters
- RESTful API for easy analysis by scripts

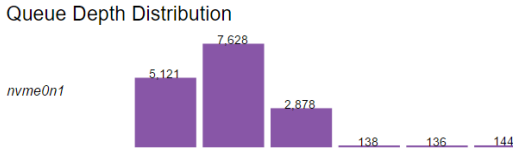
Server Topology Overview



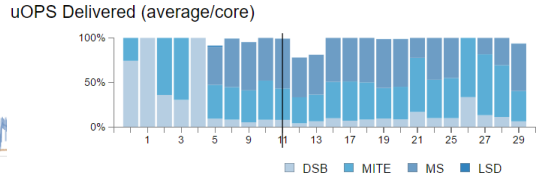
Timelines & Histograms

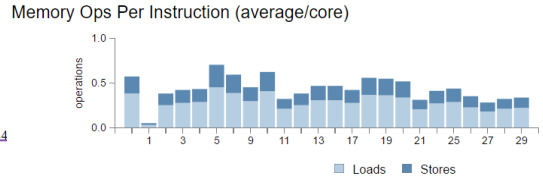


Queue Depth Distribution



Core to Core Comparisons

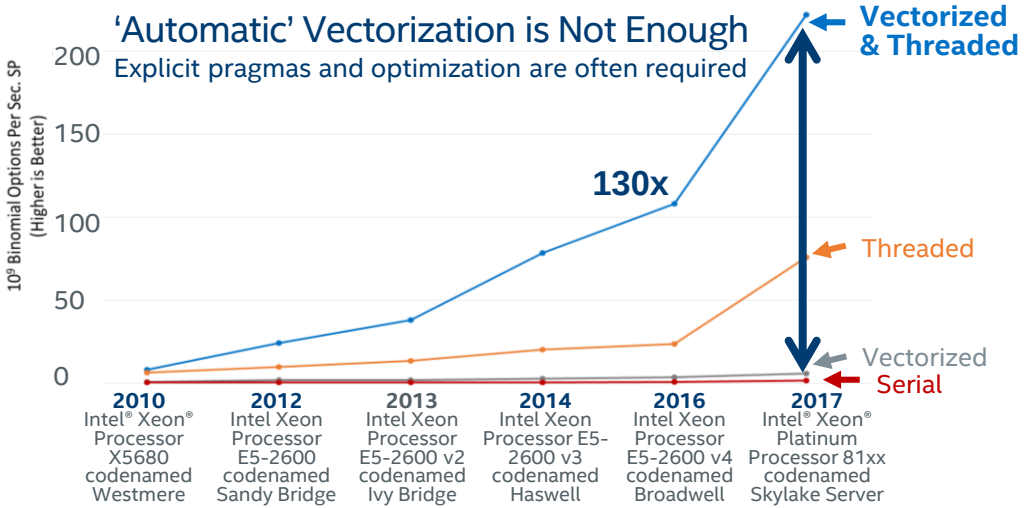




New Analysis Tool!

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

Get Breakthrough Vectorization Performance

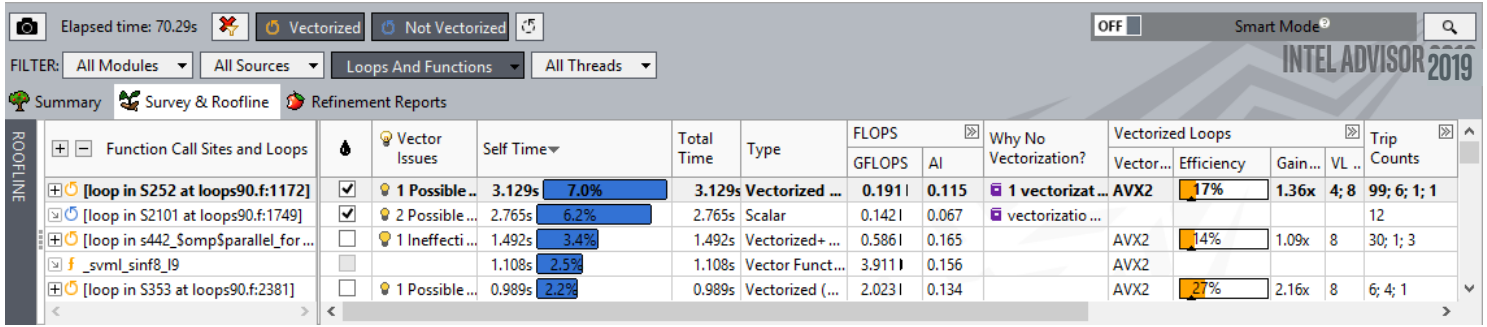
Intel® Advisor—Vectorization Advisor

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



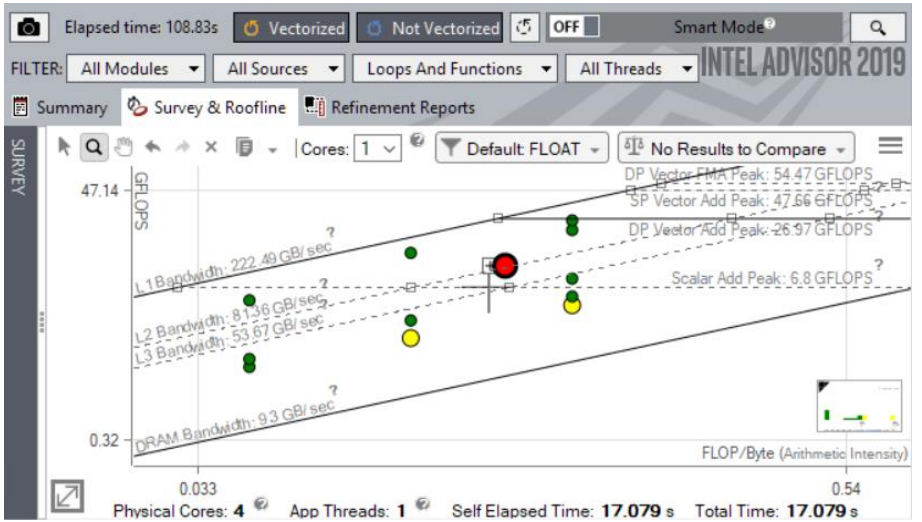
Optimize for Intel® Advanced Vector Extensions 512 (Intel® AVX-512) with or without access to Intel AVX-512 hardware

Find Effective Optimization Strategies

Intel® Advisor—Cache-aware Roofline Analysis

Roofline Performance Insights

- Highlights poor performing loops
- Shows performance 'headroom' for each loop
  - Which can be improved
  - Which are worth improving
- Shows likely causes of bottlenecks
- Suggests next optimization steps



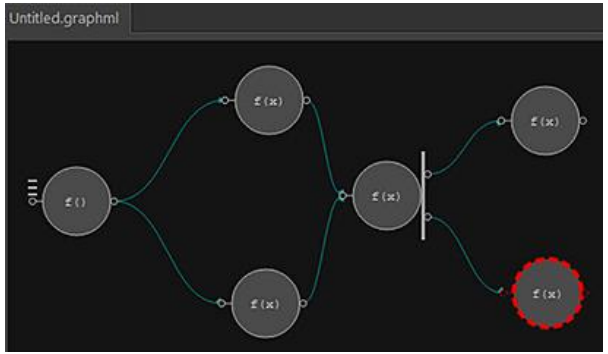
"I am enthusiastic about the new "integrated roofline" in Intel® Advisor. It is now possible to proceed with a step-by-step approach with the difficult question of memory transfers optimization & vectorization which is of major importance."

Nicolas Alferez, Software Architect
Onera – The French Aerospace Lab

Visualize Parallelism

Intel® Advisor—Flow Graph Analyzer (FGA)

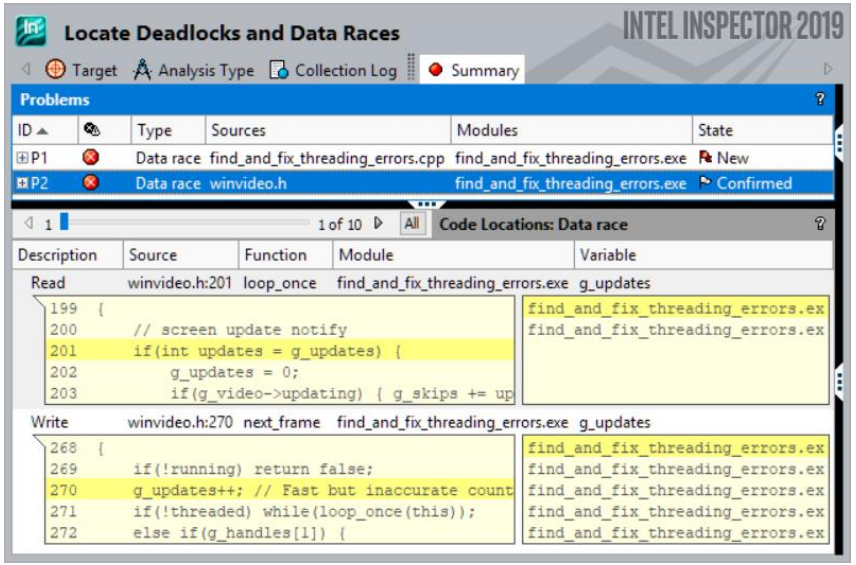
- Interactively build, validate, and analyze algorithms using the Flow Graph Analyzer module
  - Visually generate code stubs, use Threading Building Blocks (TBB) Flow Graph to get started with generating parallel C++ programs
  - Click and zoom through your algorithm's nodes and edges to understand parallel data and program flow
  - Use FGA dashboard to analyze your algorithm's load balancing, concurrency, and other parallel attributes to fine tune your program



- Enjoy full support for TBB Flow Graph; initial support for OpenMP* 5 (draft) OMPT API

Debug Memory & Threading with Intel® Inspector

Find & Debug Memory Leaks, Corruption, Data Races, Deadlocks



Correctness Tools Increase ROI by 12%-21%[1]

- Errors found earlier are less expensive to fix
- Races & deadlocks not easily reproduced
- Memory errors are hard to find without a tool

Debugger Integration Speeds Diagnosis

- Breakpoint set just before the problem
- Examine variables and threads with the debugger

What's New in 2019 Release

Find Persistent Memory Errors

- Missing / redundant cache flushes
- Missing store fences
- Out-of-order persistent memory stores
- PMDK* transaction redo logging errors

New Resource: How to Use Intel® Inspector—Persistence Inspector

Learn More: intel.ly/inspector-xe

[1]**Cost Factors – Square Project Analysis** - *CERT: U.S. Computer Emergency Readiness Team, and Carnegie Mellon CyLab NIST: National Institute of Standards & Technology: Square Project Results*

Race Conditions are Difficult to Diagnose

They only Occur Occasionally & are Difficult to Reproduce—Intel® Inspector

Correct Answer

Thread 1	Thread 2		Shared Counter
			0
Read count		←	0
Increment			0
Write count		→	1
	Read count	←	1
	Increment		1
	Write count	→	2

Incorrect Answer

Thread 1	Thread 2		Shared Counter
			0
Read count		←	0
	Read count	←	0
Increment			0
	Increment		0
Write count		→	1
	Write count	→	1

Boost Distributed Application Performance with Intel® MPI Library

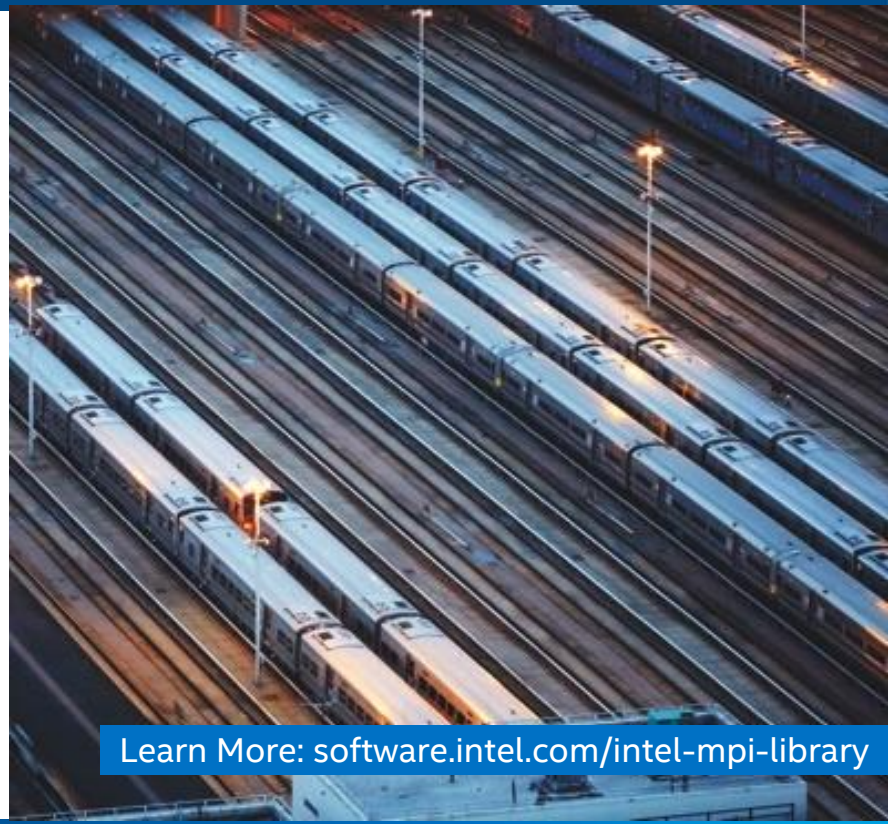
Performance, Scalability & Fabric Flexibility

Standards Based Optimized MPI Library for Distributed Computing

- Built on open source MPICH Implementation
- Tuned for low latency, high bandwidth & scalability
- Multi-fabric support for flexibility in deployment

What's New in 2019 Release

- New MPI code base- MPI-CH4 (on the path to Exascale & beyond)
- Greater scalability & shortened CPU paths
- Superior MPI Multi-threaded performance
- Supports the latest Intel® Xeon® Scalable processor



INTEL® PARALLEL STUDIO XE COMPONENT TOOLS

BUILD

Intel® C++ Compiler
Intel® Fortran Compiler
Intel® Distribution for Python*
Intel® Math Kernel Library
Intel® Integrated Performance Primitives
Intel® Threading Building Blocks
Intel® Data Analytics Acceleration Library
Included in Composer Edition

ANALYZE

Intel® VTune™ Amplifier
Intel® Advisor
Intel® Inspector

Part of the Professional Edition

SCALE

Intel® MPI Library
Intel® Trace Analyzer & Collector
Intel® Cluster Checker

Part of the Cluster Edition

Intel® MPI Library Features

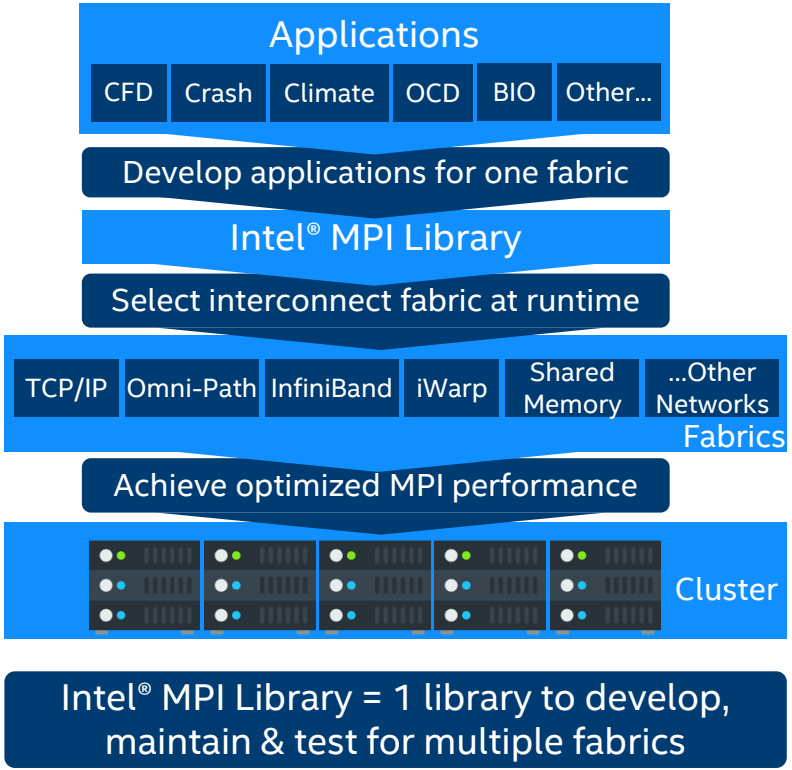
Optimized MPI Application Performance
- Application-specific tuning
- Automatic tuning
- Support for Intel® Omni-Path Architecture Fabric

Multi-vendor Interoperability & Lower Latency
- Industry leading latency
- Performance optimized support for the fabric capabilities through OpenFabrics* (OFI)

Faster MPI Communication
- Optimized collectives

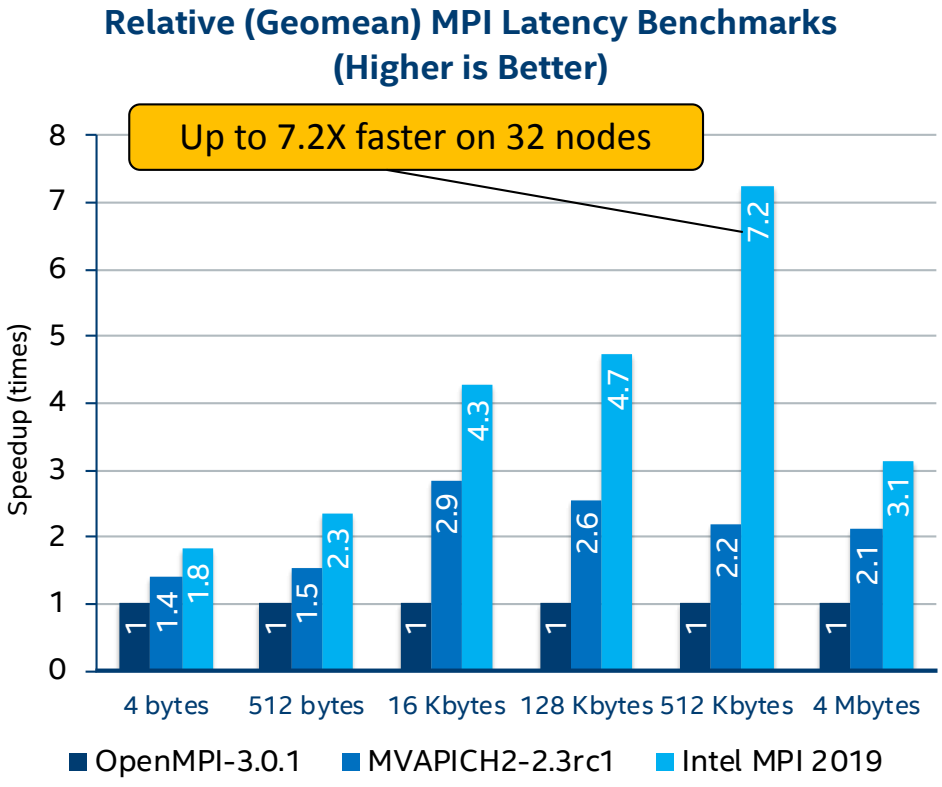
Sustainable Scalability
- Native InfiniBand* interface support allows for lower latencies, higher bandwidth, and reduced memory requirements

More Robust MPI Applications
- Seamless interoperability with Intel® Trace Analyzer & Collector



Superior MPI Performance with Intel® MPI Library on Linux* 64

1,280 Processes, 32 Xeon nodes (Intel® Omni-Path) Linux* 64



Performance results are based on testing as of Sept. 5, 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, see Performance Benchmark Test Disclosure.

Configuration: Testing by Intel as of Sept. 5, 2018.Hardware: Intel® Xeon®Gold 6148 CPU @ 2.40GHz; 192 GB RAM. Interconnect: Intel® Omni-Path Host Fabric InterfaceSoftware: RHEL* 7.4; IFS 10.7.0.0.145; Libfabric internal; Intel® MPI Library 2019; Intel® MPI Benchmarks 2019 (built with Intel® C++ Compiler XE 18.0.2.199 for Linux*);

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.

For more complete information about compiler optimizations, see our Optimization Notice.

Profile & Analyze High Performance MPI Applications

Intel® Trace Analyzer & Collector

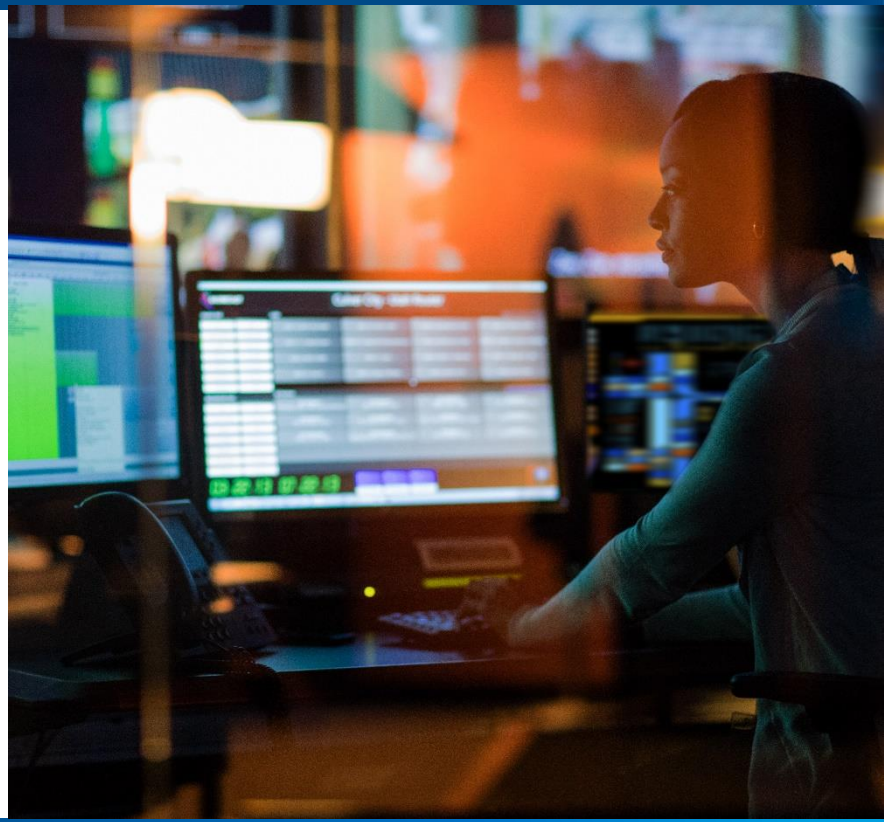
Powerful Profiler, Analysis & Visualization Tool for MPI Applications

- Low overhead for accurate profiling, analysis & correctness checking
- Easily visualize process interactions, hotspots & load balancing for tuning & optimization
- Workflow flexibility: Compile, Link or Run

What's New in 2019 Release

- Minor updates & enhancements
- Supports the latest Intel® Xeon® Scalable processors

Learn More: software.intel.com/intel-trace-analyzer

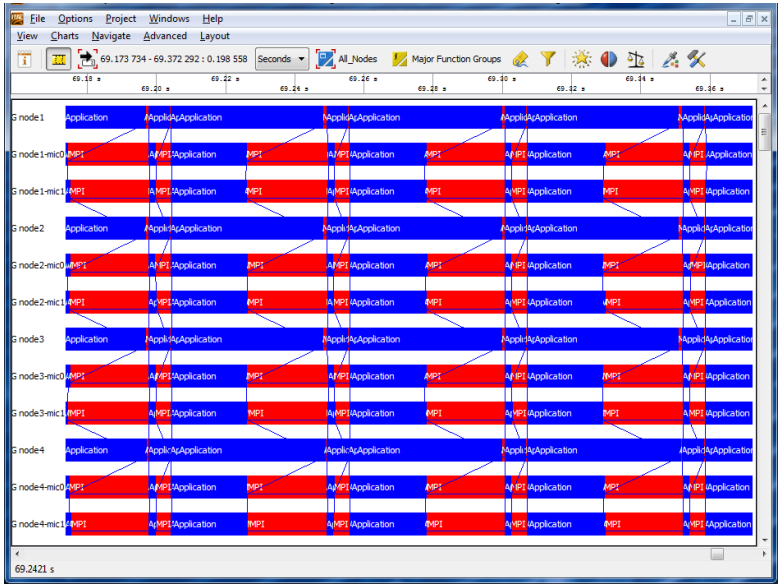


Efficiently Profile MPI Applications

Intel® Trace Analyzer & Collector

Helps Developers

- Visualize & understand parallel application behavior
- Evaluate profiling statistics & load balancing
- Identify communication hotspots



Features

- Event-based approach
- Low overhead
- Excellent scalability
- Powerful aggregation & filtering functions
- Idealizer
- Scalable

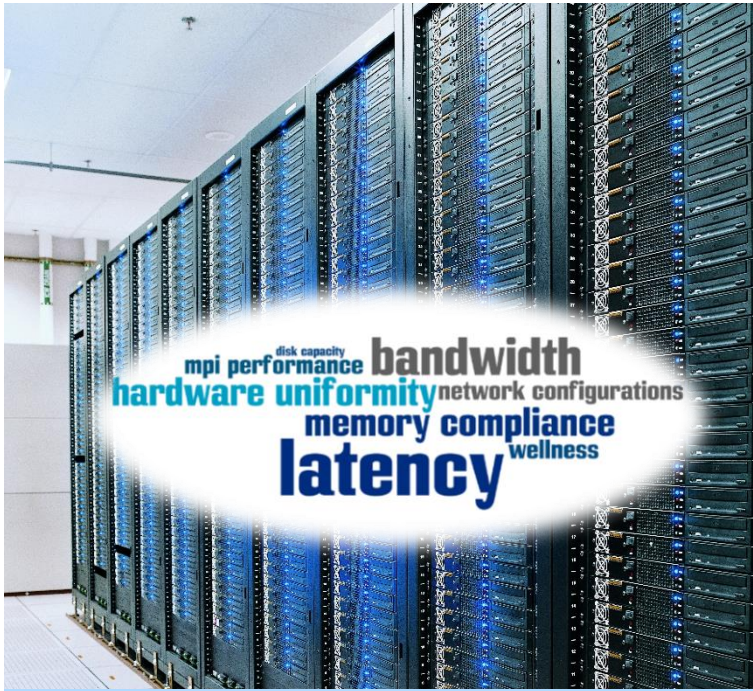
Use an Extensive Diagnostic Toolset for High Performance Compute Clusters—Intel® Cluster Checker (for Linux*)

Ensure Cluster Systems Health

- Expert system approach providing cluster systems expertise - verifies system health: find issues, offers suggested actions
- Provides extensible framework, API for integrated support
- Check 100+ characteristics that may affect operation & performance – improve uptime & productivity

New in 2019 Release: Output & Features Improve Usability & Capabilities

- Simplified execution with a **single command**
- **New output** format with overall summary
  - Simplified issue assessment for 'CRITICAL', 'WARNING', or 'INFORMATION'
  - Extended output to logfile with details on issue, diagnoses, observations
- Added **auto-node discovery** when using Slurm*
- Cluster State **2 snapshot comparison** identifies changes
- And more…



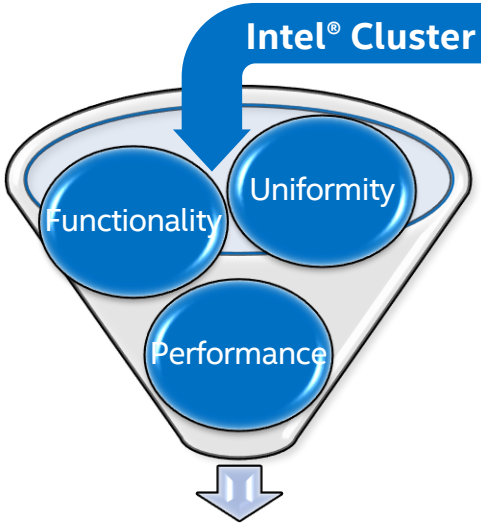
For application developers, cluster architects & users, & system administrators

Functionality, Uniformity, & Performance Tests

Intel® Cluster Checker

Comprehensive pre-packed cluster systems expertise out-of-the-box

- ✓ Suitable for HPC experts & those new to HPC
- ✓ Tests can be executed in selected groups on any subset of nodes



Functionality Tests	*Uniformity Tests*	*Performance Tests*
System-level ▪ Node ▪ Connectivity ▪ Cluster **Validation** ▪ Application platform compliance ▪ Solution compliance	**Hardware** ▪ CPUs ▪ Memory ▪ Interconnect ▪ Disks **Software** ▪ Installed packages & versions ▪ Numerous kernel & BIOS settings	**Benchmarks** ▪ DGEMM ▪ HPCG ▪ HPL ▪ Intel® MPI Benchmarks ▪ IOzone ▪ STREAM
API Available for Integration		

✓ Get compact reports, find problems, validate status

System Qualification with Intel® Cluster Checker

3 Phases of Use from Standing Up to Ongoing Cluster Operations

Installation Qualification	Operational Qualification	Performance Qualification
Evaluate Verify correct installation	**Evaluate** Verify correct operation	**Evaluate** Verify performance
Ensure Correct installation to specs	**Ensure** Correct operation to specs	**Ensure** Correct performance to specs
Establish Baseline for cluster	**Verify** Cluster meets specs & system compliance	**Verify** Cluster meets customers performance targets

Intel Cluster Checker

Developer: Call to Action/More Resources

Download Intel® System Studio

- Free Community License
- Free/Discounted Versions for Students /Academia

Access Developer Resources

- Product site
- Get Started/Documentation
- Code Samples/Training
- Tech.Decoded webinars, how-to videos/articles
- Expert Community Support Forum

Paid Licenses include Priority Support

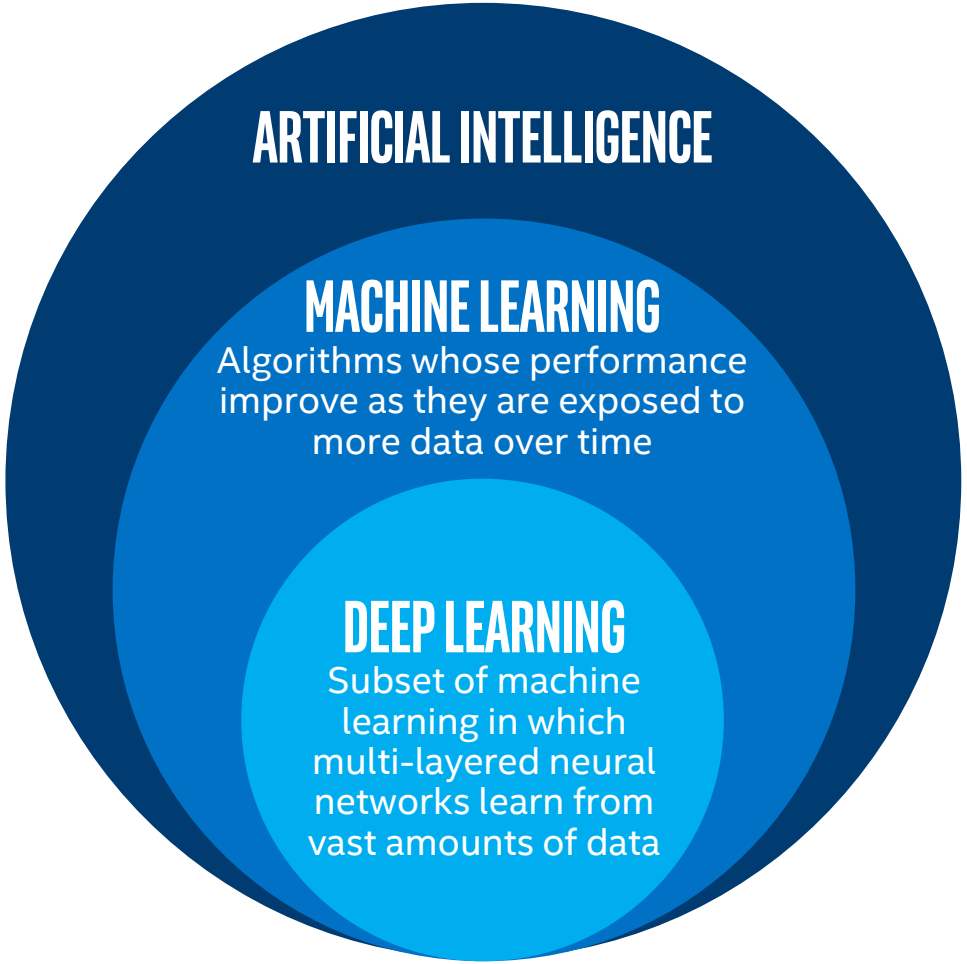
Agenda

- Intel Software Development Tools

- **Intel optimized AI Solutions**

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



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
Jaw line length
...etc.

CLASSIFIER ALGORITHM

SVM
Random Forest
Naïve Bayes
Decision Trees
Logistic Regression
Ensemble methods

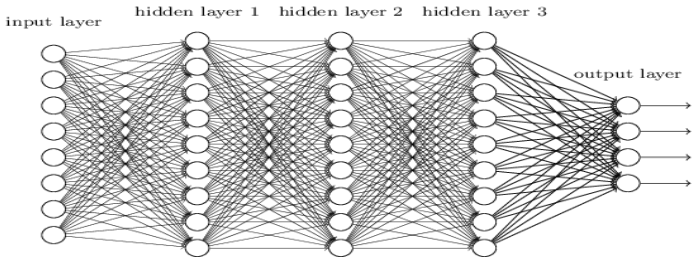
Arjun

DEEP LEARNING

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

$N \times N$

NEURAL NETWORK

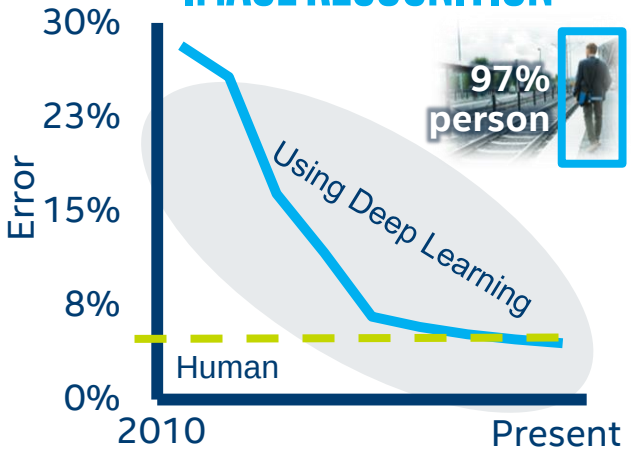


Arjun

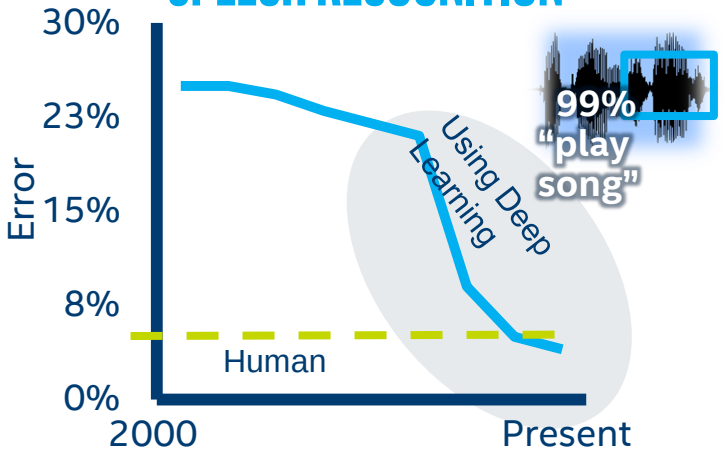
DEEP LEARNING BREAKTHROUGHS

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)

ARTIFICIAL INTELLIGENCE

SOLUTIONS
Solution Architects

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
- Scikit-learn
- Pandas
- NumPy

R
- Cart
- Random Forest
- e1071

Distributed
- MLlib (on Spark)
- Mahout

DEEP LEARNING FRAMEWORKS

COMING SOON!

Now optimized for CPU

TensorFlow* | MXNet* | Caffe* | BigDL/Spark*

Optimizations in progress

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

ATOM | CORE 8th Gen | XEON PLATINUM

Data Center
Edge
Device

DEEP LEARNING ACCELERATORS

NERVANA (COMING 2019) | STRATIX 10 | ARRIA 10 | MOVIDIUS | GNA

NNP L-1000

Inference

† 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.

AI.INTEL.COM

INTEL® MATH KERNEL LIBRARY

Components Comparison : Intel MKL-DNN vs Intel MKL

MKL-DNN (Open Source)

- Convolution
- Pooling
- ReLU
- Inner Product
- Normalization

MKL (Math Kernel Library)

Linear Algebra	Fast Fourier Transforms	Vector Math	Summary Statistics	And More…
• BLAS • LAPACK • ScaLAPACK • Sparse BLAS • Sparse Solvers • Iterative • PARDISO* • Cluster Sparse Solver	• Multidimensional • FFTW interfaces • Cluster FFT	• Trigonometric • Hyperbolic • Exponential • Log • Power • Root • Vector RNGs	• Kurtosis • Variation coefficient • Order statistics • Min/max • Variance-covariance	• Splines • Interpolation • Trust Region • Fast Poisson Solver

INTEL® DATA ANALYTICS ACCELERATION LIBRARY

Speed-up Machine Learning and Analytics with Intel® Data Analytics Acceleration Library (Intel® DAAL)

Boost Machine Learning & Data Analytics Performance

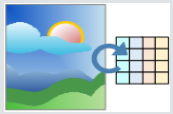
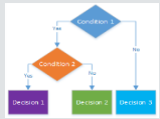
- Helps applications deliver better predictions faster
- Optimizes data ingestion & algorithmic compute together for highest performance
- Supports offline, streaming & distributed usage models to meet a range of application needs
- Split analytics workloads between edge devices and cloud to optimize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

Learn More: software.intel.com/daal

What's New in the 2019 Release

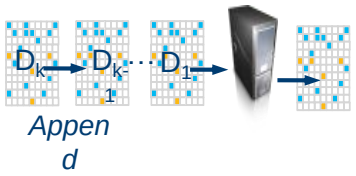
New Algorithms

- **High performance Logistic Regression**, most widely-used classification algorithm
- **Extended Gradient Boosting Functionality** provides inexact split calculations & algorithm-level computation canceling by user-defined callback for greater flexibility
- **User-defined Data Modification Procedure in CSV & IDBC data sources to implement** a wide range of feature extraction & transformation techniques

Pre-processing	Transformation	Analysis	Modeling	Validation	Decision Making
					
Decompression, Filtering, Normalization	Aggregation, Dimension Reduction	Summary Statistics Clustering, etc.	Machine Learning (Training) Parameter Estimation Simulation	Hypothesis Testing Model Errors	Forecasting Decision Trees, etc.

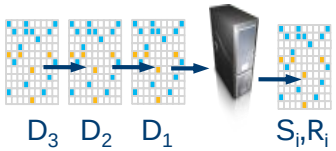
Processing Modes

Batch Processing



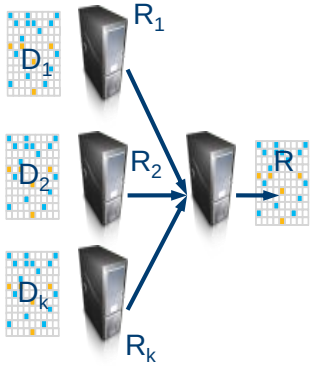
$R = F(D_1,\ldots,D_k)$

Online Processing



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

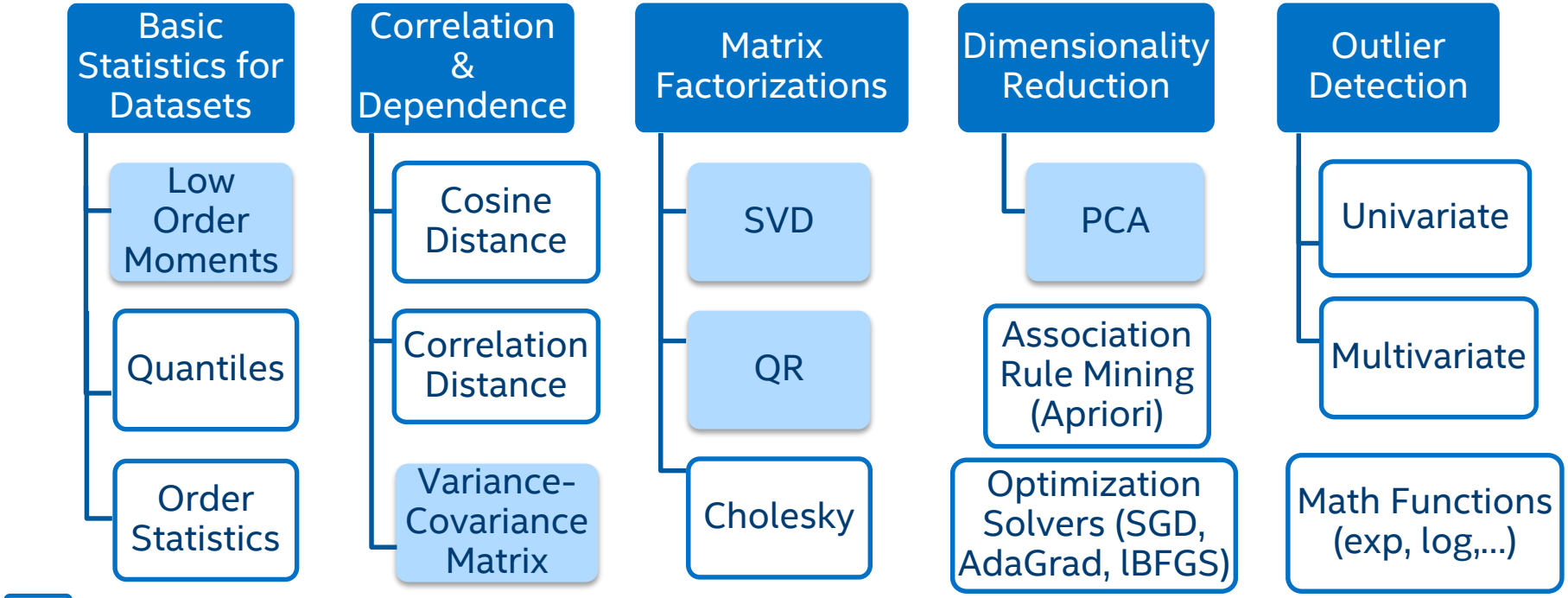
Distributed Processing



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

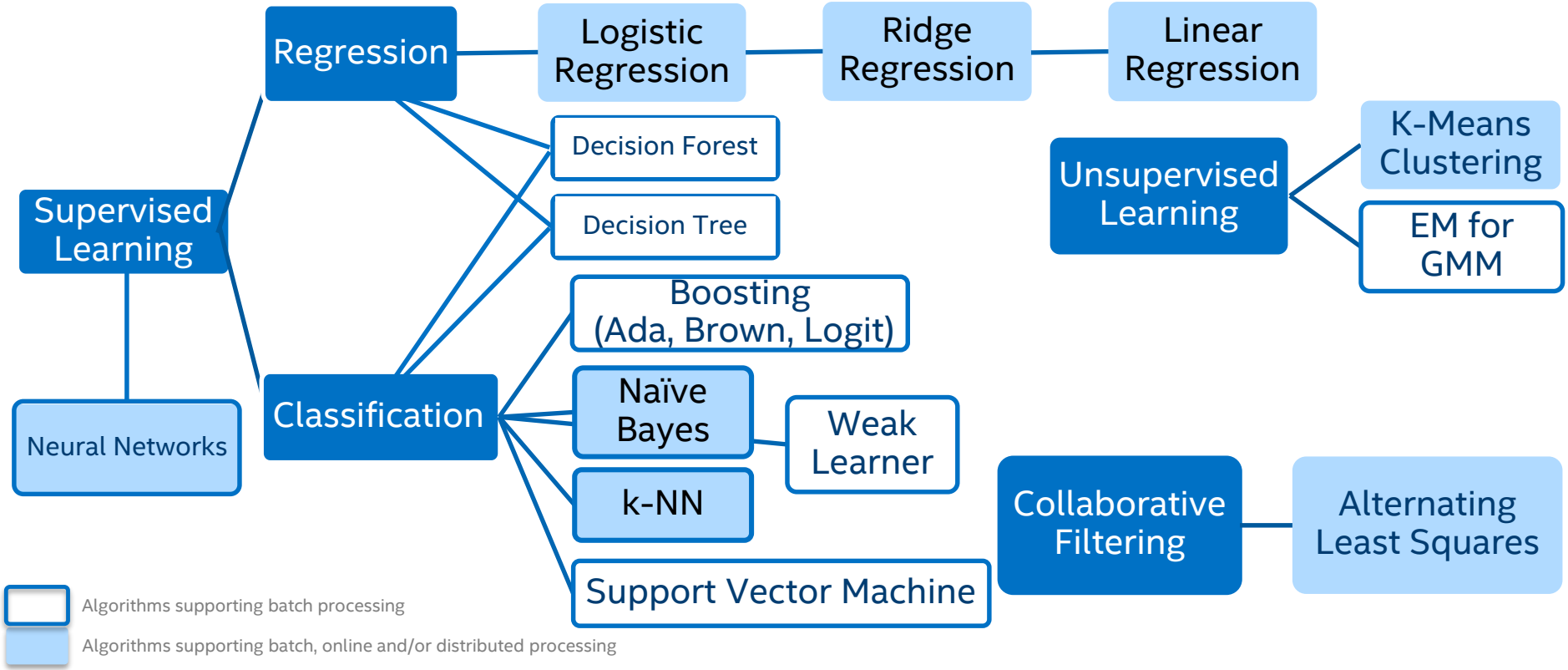
Data Transformation & Analysis Algorithms

Intel® Data Analytics Acceleration Library



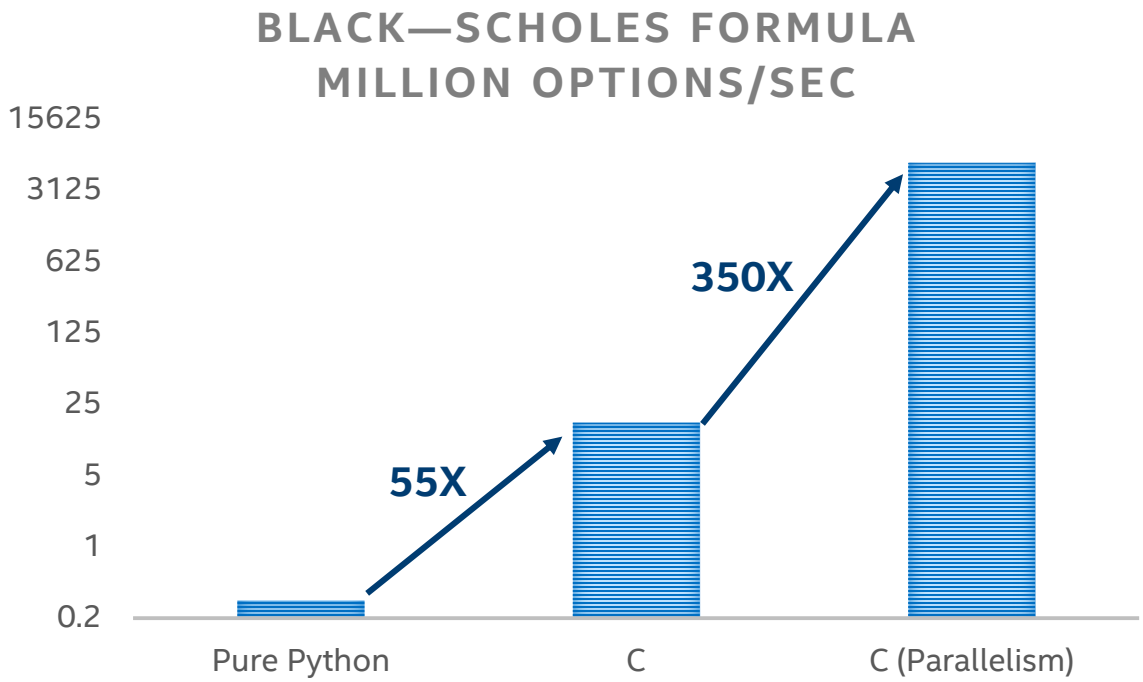
Machine Learning Algorithms

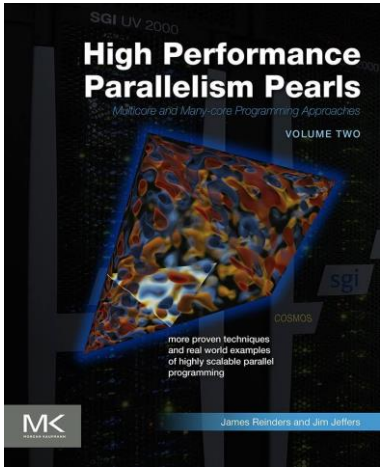
Intel® Data Analytics Acceleration Library



INTEL DISTRIBUTION OF PYTHON

Performance gap between C and Python



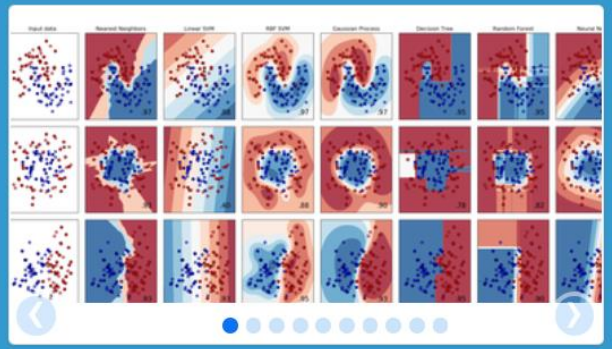


Chapter 19: Performance Optimization of **Black—Scholes** Pricing

$$V_{call} = S_0 \cdot \mathrm{CDF}(d_1) - e^{-rT} \cdot X \cdot \mathrm{CDF}(d_2)$$
$$V_{put} = e^{-rT} \cdot X \cdot \mathrm{CDF}(-d_2) - S_0 \cdot \mathrm{CDF}(-d_1)$$

$$d_1 = \frac{\ln\left(S_0/X\right) + \left(r + \sigma^2/2\right)T}{\sigma\sqrt{T}}$$
$$d_2 = \frac{\ln\left(S_0/X\right) + \left(r - \sigma^2/2\right)T}{\sigma\sqrt{T}}$$

The most popular ML package for Python – Intel Optimized



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, … — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

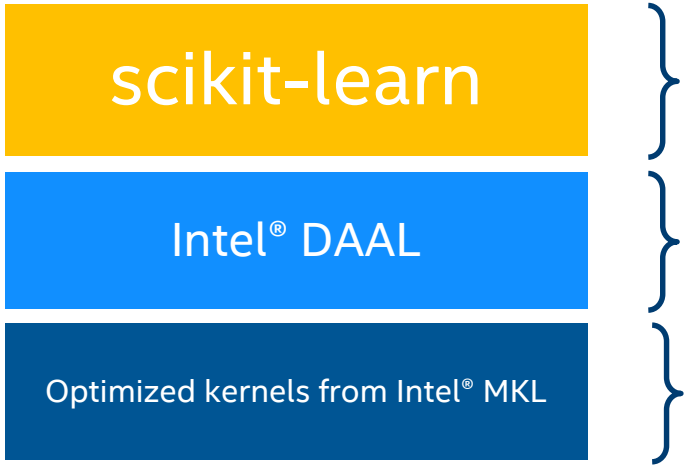
Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, … — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, … — Examples

Optimizing scikit-learn with Intel® DAAL

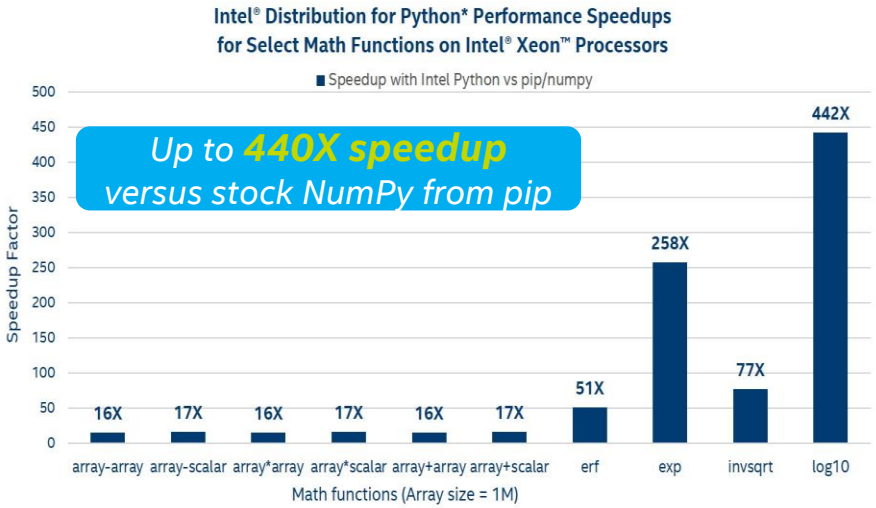


- The most popular package for machine learning
- Hundreds of algorithms with different parameters
- Has a very flexible and easy-to-use interface

High performance of analytical and machine learning algorithms on Intel architecture

High performance basic mathematical routines (BLAS, vector math, RNG, ...)

Faster Python* with Intel® Distribution for Python



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

High Performance Python Distribution

- Accelerated NumPy, SciPy, scikit-learn well suited for scientific computing, machine learning & data analytics
- Drop-in replacement for existing Python. No code changes required
- Highly optimized for latest Intel processors
- Take advantage of Priority Support – connect direct to Intel engineers for technical questions[2]

What's New in 2019 version

- Faster Machine learning with Scikit-learn functions
  - Support Vector Machine (SVM) and K-means prediction, accelerated with Intel® DAAL
- Built-in access to XGBoost library for Machine Learning
- Access to Distributed Gradient Boosting algorithms
- Ease of access installation
- Now integrated into Intel® Parallel Studio XE installer.

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.

[2]Paid versions only.

Installing Intel® Distribution for Python* 2018

Standalone Installer

Download full installer from
https://software.intel.com/en-us/intel-distribution-for-python

Anaconda.org
Anaconda.org/intel channel

```
> conda config --add channels intel
> conda install intelpython3_full
> conda install intelpython3_core
```

PyPI

```
> pip install intel-numpy
> pip install intel-scipy
> pip install mkl_fft
> pip install mkl_random
```

+ Intel library Runtime packages
+ Intel development packages

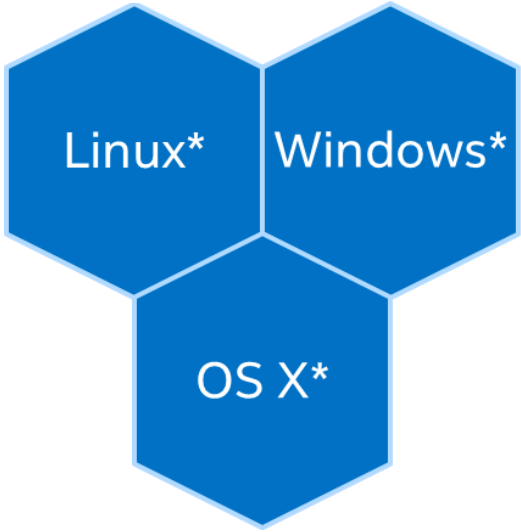
Docker Hub

```
docker pull intelpython/intelpython3_full
```

YUM/APT

Access for yum/apt:
https://software.intel.com/en-us/articles/installing-intel-free-libs-and-python

2.7 & 3.6
(3.7 coming soon)



Tune Python* + Native Code for Better Performance

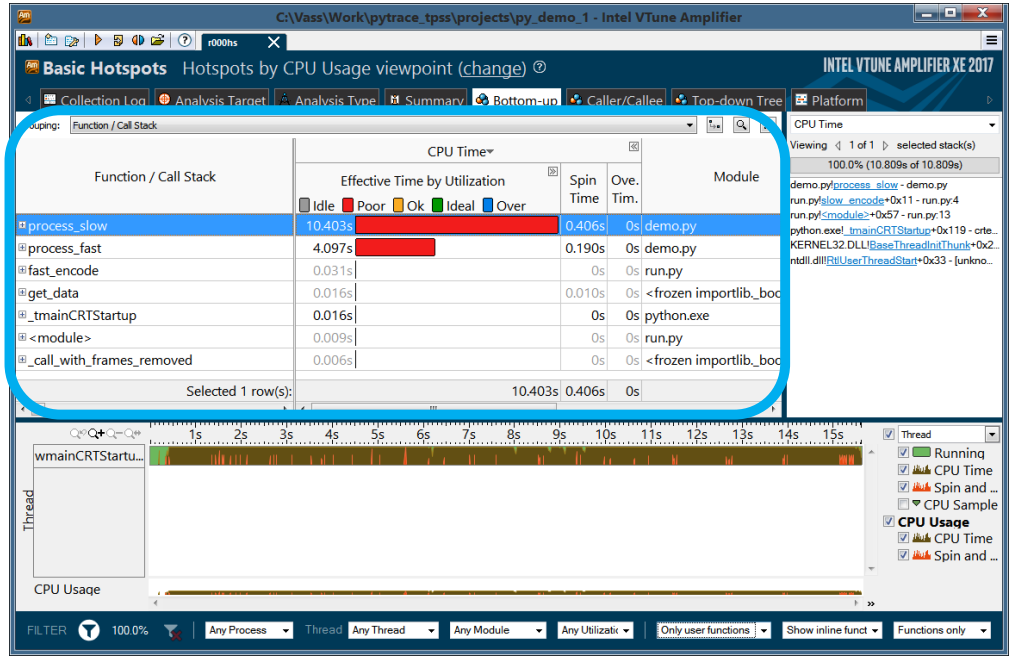
Analyze Performance with Intel® VTune™ Amplifier (available in Intel® Parallel Studio XE)

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



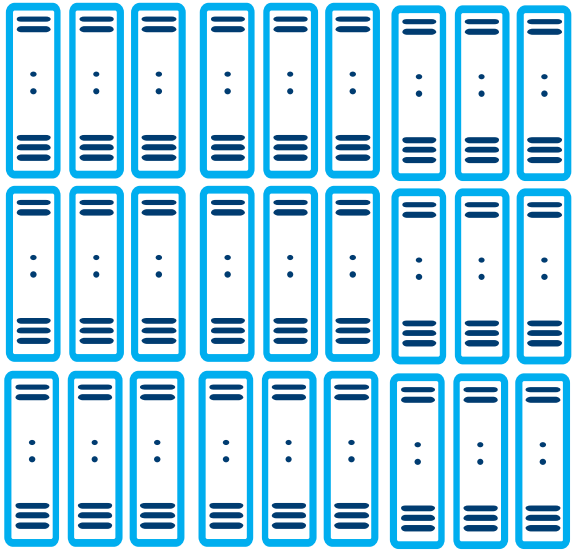
Auto detection & performance analysis of Python & native functions

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

INTEL OPTIMIZED FRAMEWORKS

HPC

AI



INTEL AI (DISTRIBUTED) FRAMEWORKS

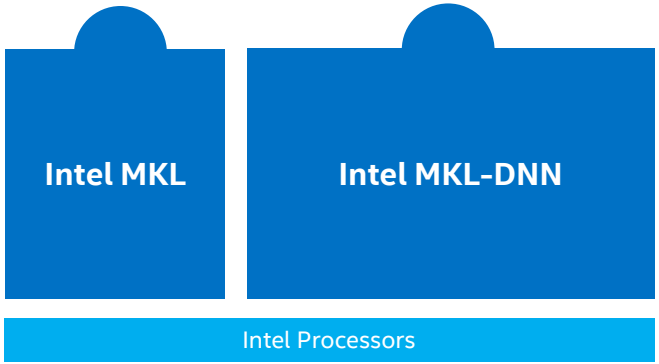
ai.intel.com/framework-optimizations/

Caffe2 PYTORCH* PaddlePaddle*

and more…

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.

Deep Learning Software Stack for Intel processors



Deep learning and AI ecosystem includes edge and datacenter applications.

- Open source frameworks (Tensorflow*, MXNet*, CNTK*, PaddlePaddle*)
- Intel deep learning products (Neon™ framework , BigDL, OpenVINO™ toolkit)
- In-house user applications

Intel MKL and Intel MKL-DNN optimize deep learning applications for Intel processors :

- through the collaboration with framework maintainers to upstream changes (Tensorflow*, MXNet*, PaddlePaddle*, CNTK*)
- through Intel optimized forks (Caffe*, Torch*, Theano*)
- by partnering to enable proprietary solutions

Intel MKL-DNN is an open source performance library for deep learning applications (available at https://github.com/intel/mkl-dnn)

- Fast open source implementations for wide range of DNN functions
- Early access to new and experimental functionality
- Open for community contributions

Intel MKL is a proprietary performance library for wide range of math and science applications
Distribution: Intel Registration Center, package repositories (apt, yum, conda, pip)

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

INTEL® XEON® PROCESSOR PLATFORM PERFORMANCE

Hardware plus optimized software

INFERENCE THROUGHPUT

198x

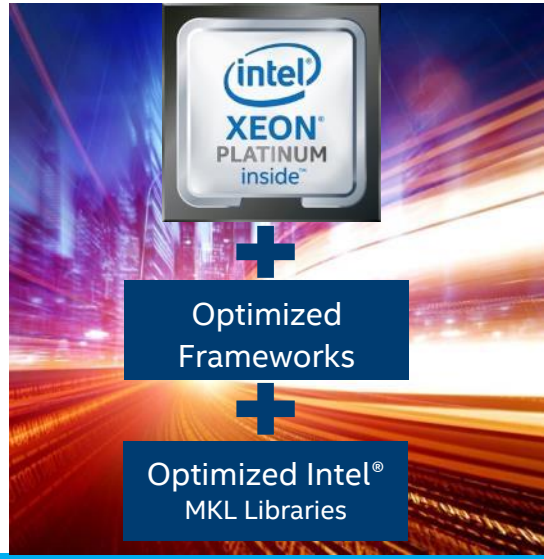
Intel® Xeon® Platinum 8180 Processor
higher Intel optimized Caffe GoogleNet v1 with Intel® MKL
inference throughput compared to
Intel® Xeon® Processor E5-2699 v3 with BVLC-Caffe

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

Components of Apache Spark

An Integrated Framework for Advanced Analytics

Spark SQL

- Seamlessly mix SQL queries
- Unified Data Access (load/query)
- Schema RDD to support structured/ semi structured data

Spark Streaming

- Data continuously streamed in & processed in near real-time
- Support in-memory computation in a fault tolerant manner

GraphX (Graph)

- Distributed graph processing framework for advanced machine learning & data mining

MLlib (Machine Learning)

- Distributed machine learning framework
- Library of distributed machine learning & statistical algorithms

Apache Spark*/Spark Core API

Provides distributed task scheduling, dispatching, memory management, fault recovery, and basic I/O functionalities with a fundamental programming abstraction, RDD for in-memory computation in a fault tolerant manner

R | SQL | Python | Scala | Java

Runs on and access diverse data sources

hadoop | APACHE HBASE | MESOS | cassandra

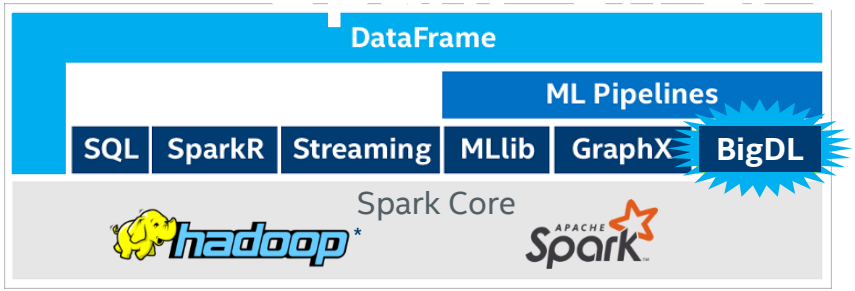
*Other names and brands may be claimed as the property of others

Source: http://spark.apache.org/

INTEL BIGDL - DISTRIBUTED DEEP LEARNING LIBRARY

Performance

Learning for **Apache**



BigDL is an **open-source** distributed deep learning library for Apache Spark* that can run directly on top of existing Spark or Apache Hadoop* clusters

Ideal for DL Models TRAINING and INFERENCE

Designed and Optimized for Intel® Xeon®

No need to deploy costly accelerators, duplicate data, or suffer through scaling headaches!

Feature Parity with TensorFlow*, Caffe* and Torch*

Lower TCO and improved ease of use with existing infrastructure

Deep Learning on Big Data Platform, Enabling Efficient Scale-Out

Powered by Intel® MKL and multi-threaded programming

software.intel.com/bigdl

intel

TensorFlow with Intel MKL/MKL-DNN

Use Intel Distribution for Python*

- Uses Intel MKL for many NumPy operations thus supports MKL_VERBOSE=1
- Available via Conda, or YUM and APT package managers

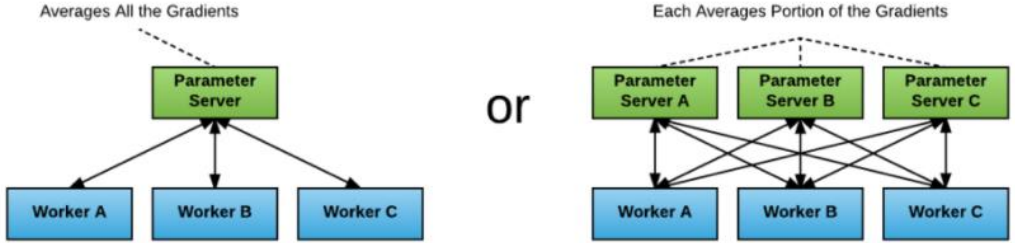
Use pre-built Tensorflow* wheels or build TensorFlow* with `bazel build --config=mkl`

- **Building from source required for integration with Intel Vtune™ Amplifier**
- Follow the CPU optimization advices including setting affinity and # of intra- and inter- ops threads
- More Intel MKL-DNN-related optimizations are slated for the next version: Use the latest TensorFlow* master if possible

Distributed TensorFlow™ Compare

Distributed Tensorflow with Parameter Server

With Parameter Server →

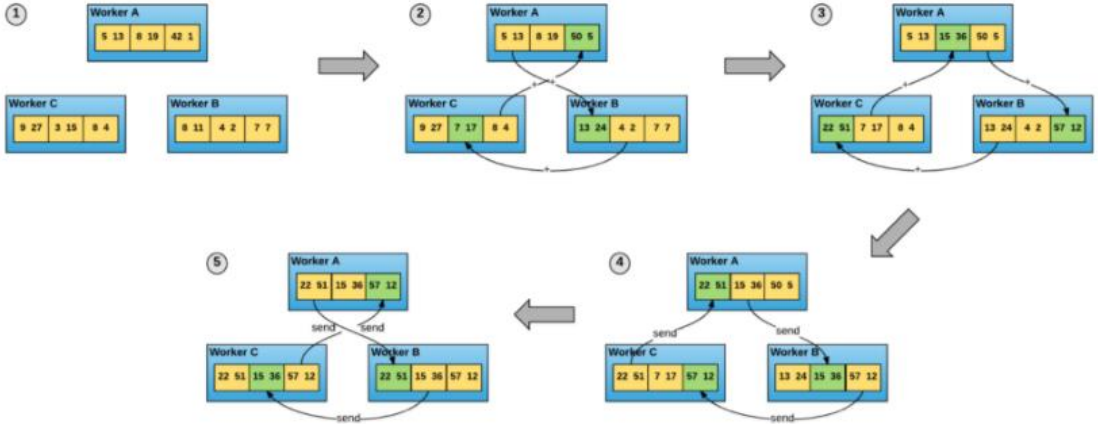


The parameter server model for distributed training jobs can be configured with different ratios of parameter servers to workers, each with different performance profiles.

HOROVOD

Uber's open source Distributed training framework for TensorFlow

No Parameter Server →



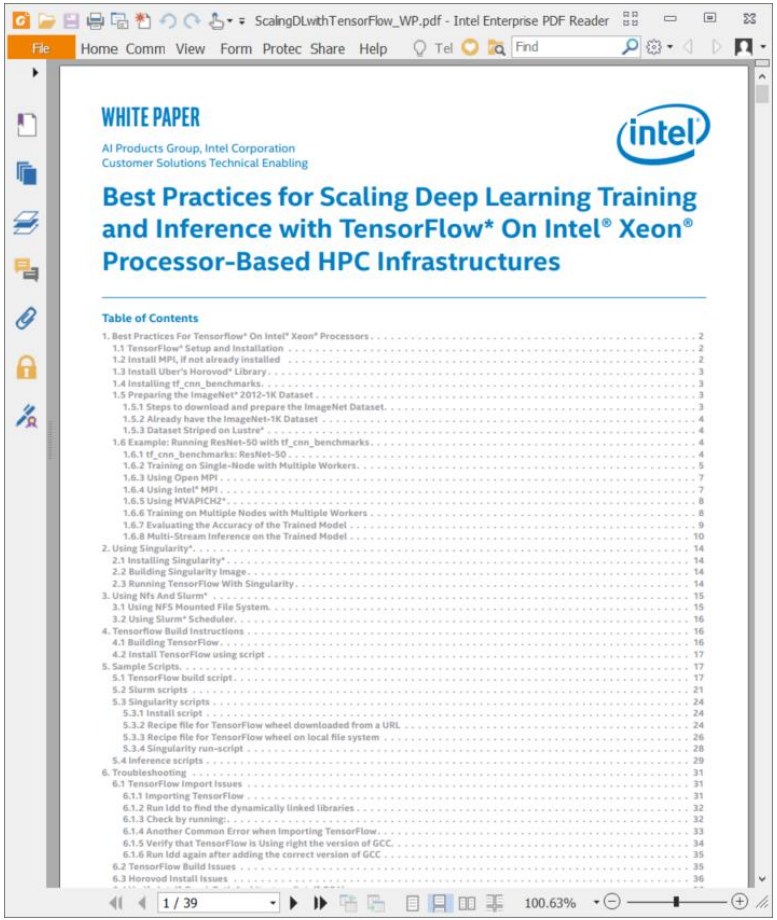
The ring all-reduce algorithm allows worker nodes to average gradients and disperse them to all nodes without the need for a parameter server.

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

Scaling TensorFlow

There is way more to consider when striking for peak performance on distributed deep learning training.:

https://ai.intel.com/white-papers/best-known-methods-for-scaling-deep-learning-with-tensorflow-on-intel-xeon-processor-based-clusters/



Intel distribution of Caffe

A fork of BVLC Caffe* maintained by Intel (Github)

The best-performing CPU framework for CNNs

Supports low-precision inference on Intel Xeon Scalable Processors (formerly known as Skylake)

CROSS PLATFORM INFERENCE

OpenVINO™ - TOOLKIT
Machine Learning / Deep Learning Inference

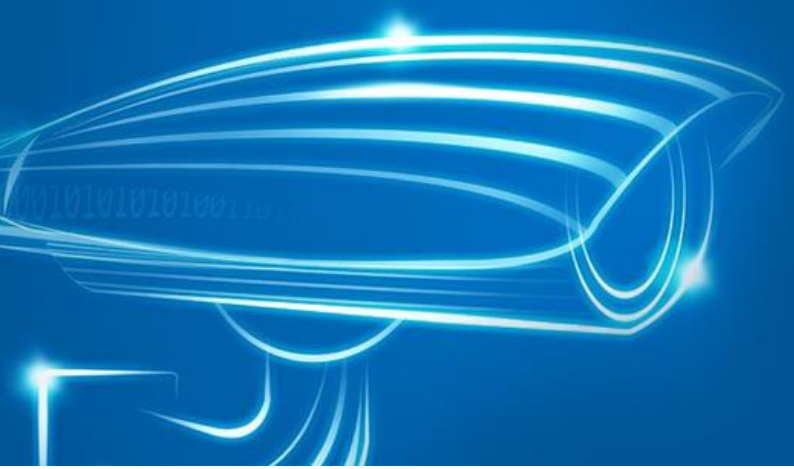
Computer Vision SDK

Deep Learning (DL) Deployment Toolkit

Deep Learning Algorithms

Optimized DL Frameworks

INTEL® DISTRIBUTION OF OPENVINO™ TOOLKIT



Take your computer vision solutions to a new level with deep learning inference intelligence.

What it is

A toolkit to accelerate development of **high performance computer vision** & **deep learning into vision applications** from device to cloud. It enables deep learning on hardware accelerators and easy deployment across multiple types of Intel® platforms.

Who needs this product?

- Computer vision/deep learning software developers
- Data scientists
- OEMs, ISVs, System Integrators

Usages

Security surveillance, robotics, retail, healthcare, AI, office automation, transportation, non-vision use cases (speech, text) & more.

HIGH PERFORMANCE, PERFORM AI AT THE EDGE

STREAMLINED & OPTIMIZED DEEP LEARNING INFERENCE

HETEROGENEOUS, CROSS-PLATFORM FLEXIBILITY

Free Download ▸ software.intel.com/openvino-toolkit

Open Source version ▸ 01.org/openvinotoolkit

Latest version is 2018 R5

What's Inside Intel® Distribution of OpenVINO™ toolkit

Intel® Deep Learning Deployment Toolkit

Model Optimizer
Convert & Optimize

IR

Inference Engine
Optimized Inference

30+ Pre-trained Models

Computer Vision Algorithms

Samples

IR = Intermediate Representation file

Traditional Computer Vision

Optimized Libraries & Code Samples

OpenCV*

OpenVX*

Samples

For Intel® CPU & GPU/Intel® Processor Graphics

Tools & Libraries

Increase Media/Video/Graphics Performance

Intel® Media SDK
Open Source version

OpenCL™ Drivers & Runtimes

For GPU/Intel® Processor Graphics

Optimize Intel® FPGA (Linux* only)

FPGA RunTime Environment
(from Intel® FPGA SDK for OpenCL™)

Bitstreams

OS Support: CentOS* 7.4 (64 bit), Ubuntu* 16.04.3 LTS (64 bit), Microsoft Windows* 10 (64 bit), Yocto Project* version Poky Jethro v2.0.3 (64 bit)

Intel® Architecture-Based Platforms Support

CELERON, ATOM, CORE, XEON, ARRIA 10, MOVIDIUS, IRIS Pro Graphics

Intel® Vision Accelerator Design Products & AI in Production/ Developer Kits

An open source version is available at 01.org/openvinotoolkit (some deep learning functions support Intel CPU/GPU only).

Intel® Deep Learning Deployment Toolkit

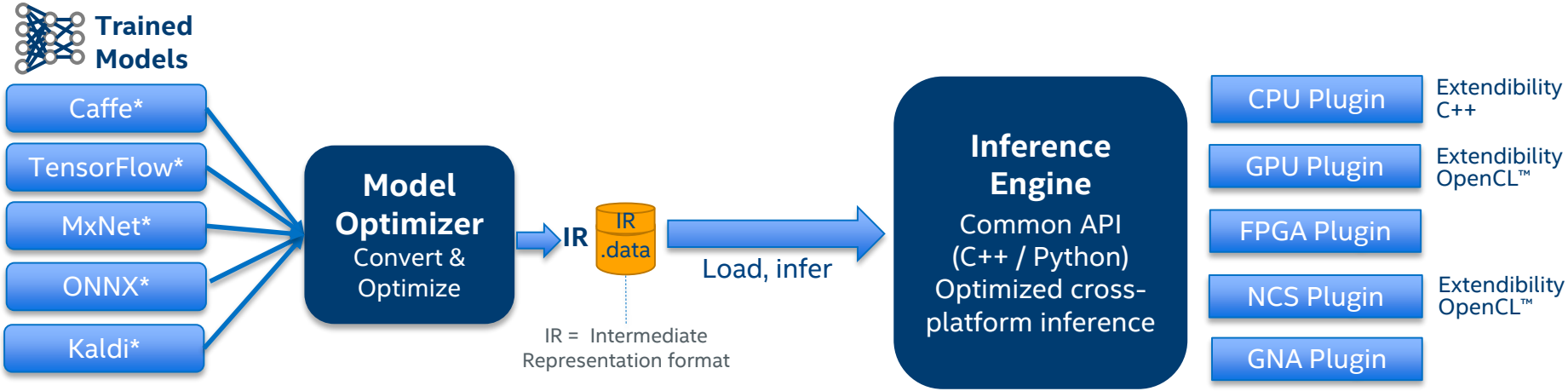
For Deep Learning Inference

Model Optimizer

- **What it is**: A python based tool to import trained models and convert them to Intermediate representation.
- **Why important**: Optimizes for performance/space with conservative topology transformations; biggest boost is from conversion to data types matching hardware.

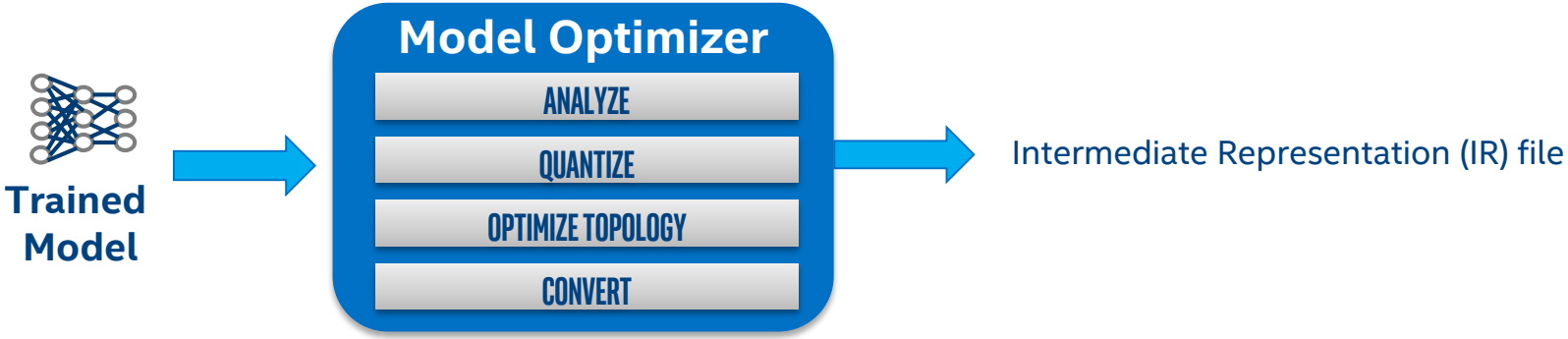
Inference Engine

- **What it is**: High-level inference API
- **Why important**: Interface is implemented as dynamically loaded plugins for each hardware type. Delivers best performance for each type without requiring users to implement and maintain multiple code pathways.



GPU = Intel CPU with integrated graphics processing unit/Intel® Processor Graphics

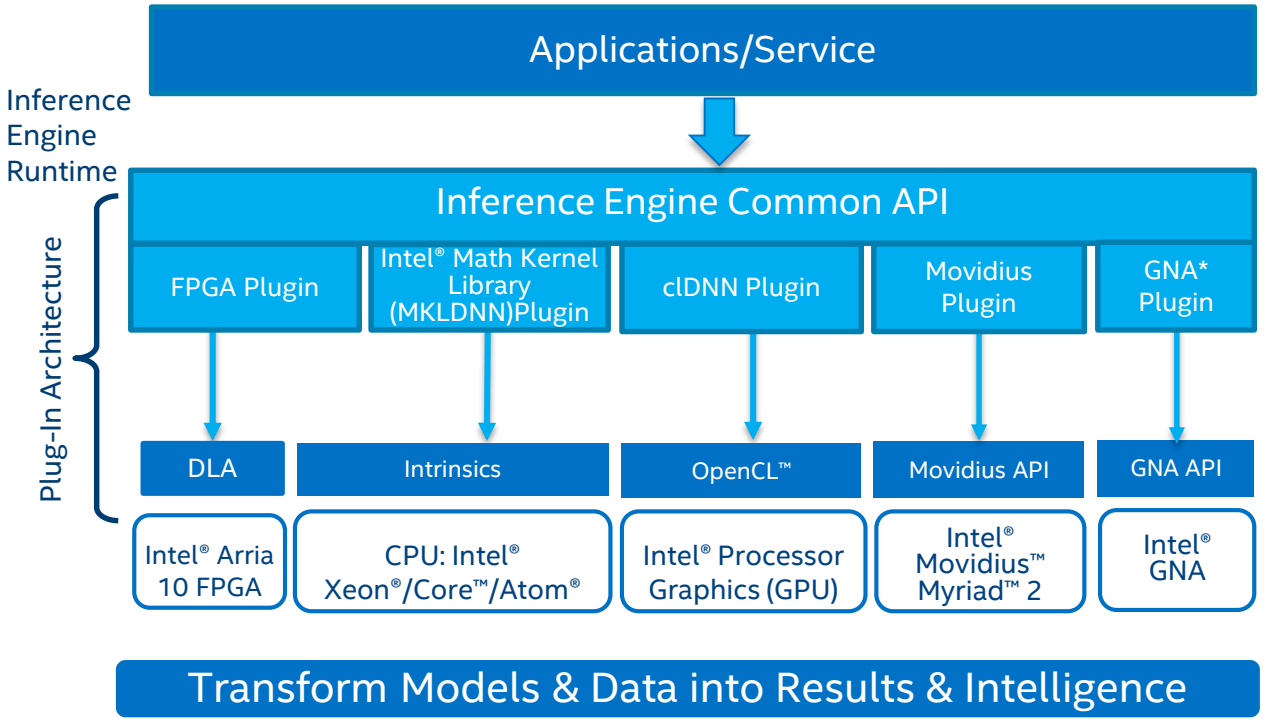
Improve Performance with Model Optimizer



- Easy to use, Python*-based workflow does not require rebuilding frameworks.
- Import Models from various supported frameworks - Caffe*, TensorFlow*, MXNet*, ONNX*, Kaldi*.
- 100+ models for Caffe, MXNet and TensorFlow validated. All public models on ONNX* model zoo supported.
- With support of LSTM and 3D Convolutional based networks and Kaldi framework / Kaldi Nnet2*, the model optimizer extends inferencing for non-vision networks.
- IR files for models using standard layers or user-provided custom layers do not require Caffe.
- Fallback to original framework is possible in cases of unsupported layers, but requires original framework.

Optimal Model Performance Using the Inference Engine

- Simple & Unified API for Inference across all Intel® architecture
- Optimized inference on large IA hardware targets (CPU/GEN/FPGA)
- Heterogeneity support allows execution of layers across hardware types
- Asynchronous execution improves performance
- Futureproof/scale your development for future Intel® processors



GPU = Intel CPU with integrated graphics/Intel® Processor Graphics/GEN
GNA = Gaussian mixture model and Neural Network Accelerator

Speed Deployment with Intel Optimized Pre-trained Models

OpenVINO™ toolkit includes optimized pre-trained models to expedite development and improve deep learning inference on Intel® processors. Use these models for development & production deployment without the need to search for or to train your own models.

Pre-Trained Models

- Age & Gender
- Face Detection – standard & enhanced
- Head Position
- Human Detection – eye-level & high-angle detection
- Detect People, Vehicles & Bikes
- License Plate Detection: small & front facing
- Vehicle Metadata
- Human Pose Estimation
- Text Detection
- Vehicle Detection
- Retail Environment
- Pedestrian Detection
- Pedestrian & Vehicle Detection
- Person Attributes Recognition Crossroad
- Emotion Recognition
- Identify Someone from Different Videos – standard & enhanced
- Facial Landmarks
- Identify Roadside objects
- Advanced Roadside Identification
- Person Detection & Action Recognition
- Person Re-identification – ultra small/ultra fast
- Face Re-identification
- Landmarks Regression
- Smart Classroom Use Cases
- Single image Super Resolution (3 models)

Save Time with Deep Learning Samples & Computer Vision Algorithms

Samples

Use Model Optimizer & Inference Engine for public models & Intel pretrained models.

- Object Detection
- Standard & Pipelined Image Classification
- Security Barrier
- Object Detection for Single Shot Multibox Detector (SSD) using Asynch API
- Object Detection SSD
- Neural Style Transfer
- Hello Infer Classification
- Interactive Face Detection
- Image Segmentation
- Validation Application
- Multi-channel Face Detection

Computer Vision Algorithms

Start quickly with highly-optimized, ready-to-deploy, custom-built algorithms using Intel pretrained models.

- Face Detector
- Age & Gender Recognizer
- Camera Tampering Detector
- Emotions Recognizer
- Person Re-identification
- Crossroad Object Detector
- License Plate Recognition
- Vehicle Attributes Classification
- Pedestrian Attributes Classification

Starting with OpenCV* & OpenVX*

- Intel-optimized functions for faster performance on Intel hardware
- Basic building blocks to speed performance, cut development time & allow customization
- All-in-one package

- Well-established, open source, computer vision library
- Wide variety of algorithms and functions available

- Targeted at real time, low power applications
- Graph-based representation, optimization & execution
- 11 samples included

Legal Disclaimer & Optimization Notice

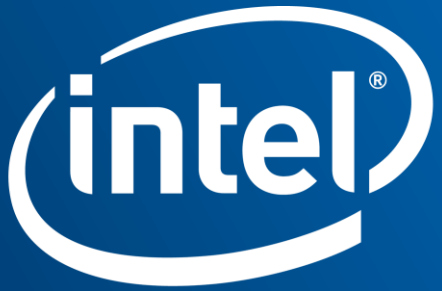
Performance results are based on testing as of August 2017 to September 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.

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



Software