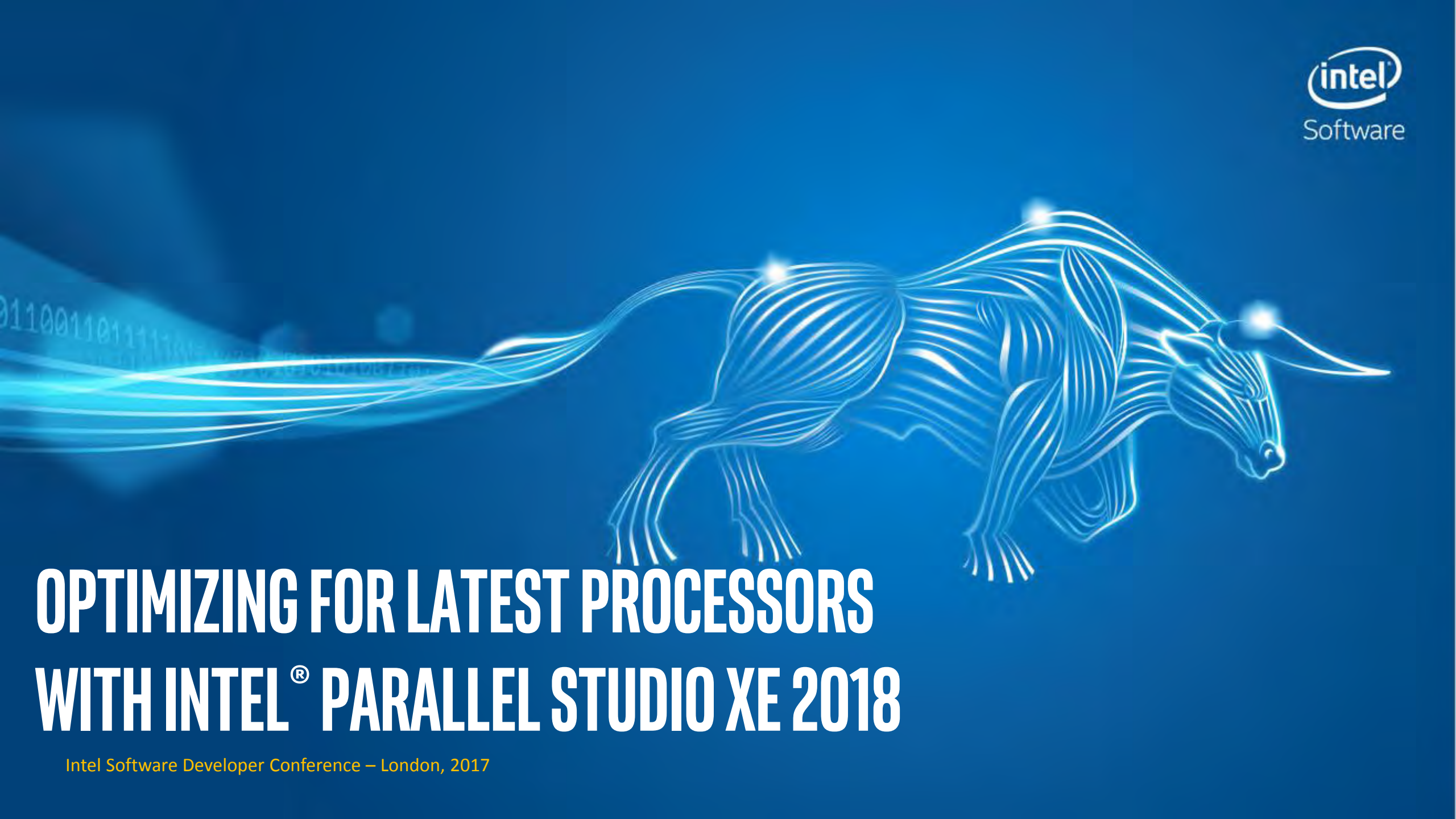


OPTIMIZING FOR LATEST PROCESSORS WITH INTEL® PARALLEL STUDIO XE 2018

Intel Software Developer Conference – London, 2017

Legal Disclaimer and Optimization Notice

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.

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.

OpenCL and the OpenCL logo are trademarks of Apple Inc. used by permission by Khronos.

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

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

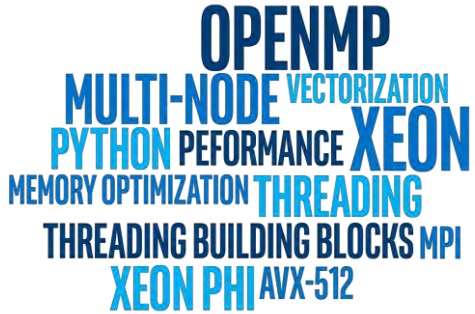
Operating System: Windows*, Linux*, MacOS[1]*

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

What's New in Intel® Parallel Studio XE 2018

Modernize your Code to be Fast, Scalable, Portable, & Parallel

- Speed application performance with **Intel® AVX-512** for the latest Intel® Xeon® Scalable and Intel® Xeon Phi™ processors. Accelerate MPI applications with Intel® Omni-Path Architecture.
- Accelerate HPC with high-performance Python*.
- Find high impact, but under optimized loops using Intel® Advisor's roofline analysis.
- Stay up-to-date with the latest standards and IDEs.
  - Full C++14 and initial C++ 2017 draft
  - Full Fortran 2008 and initial Fortran 2015 draft
  - Python 2.7 and 3.6, initial OpenMP 5.0 draft
  - Microsoft Visual Studio* 2017 integration



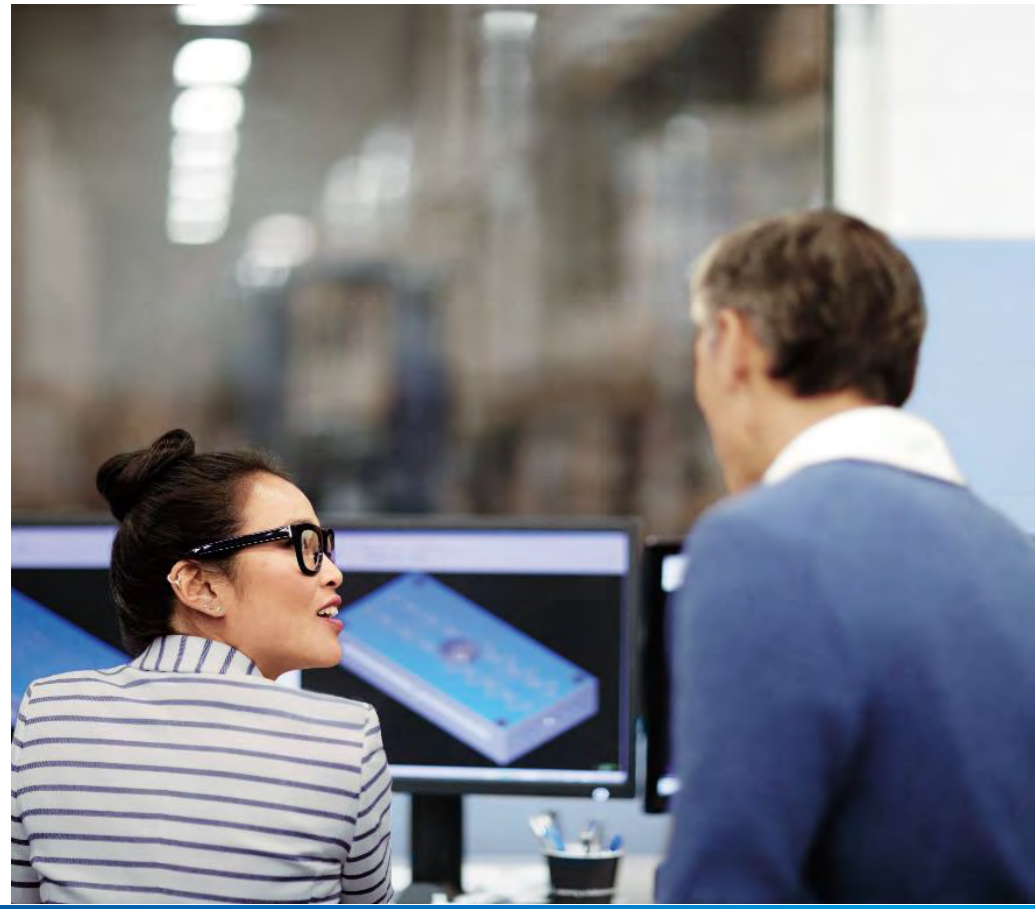
- Flexibility for What You Need
  - Quickly spot high payoff opportunities for faster code using a combined performance snapshot for MPI, CPU, FPU, and memory use. Adds MPICH and Cray support.
  - Easily access the latest Intel® Performance Libraries and Intel® Python* Distribution via APT GET, YUM and Conda.
  - New, broader redistribution rights for Intel® Performance Libraries and Intel® Distribution for Python*.

Take Advantage of Intel Priority Support

Paid licenses of Intel® Software Development Tools include Priority Support for one year from your date of purchase, with options to extend support at a highly discounted rate.

Benefits

- **Direct & private** interaction with Intel engineers. Submit confidential inquiries & code samples via the Online Service Center.
- **Responsive help** with your technical questions & other product needs.
- **Free access** to all new product updates & access to older versions.



Additional Resources

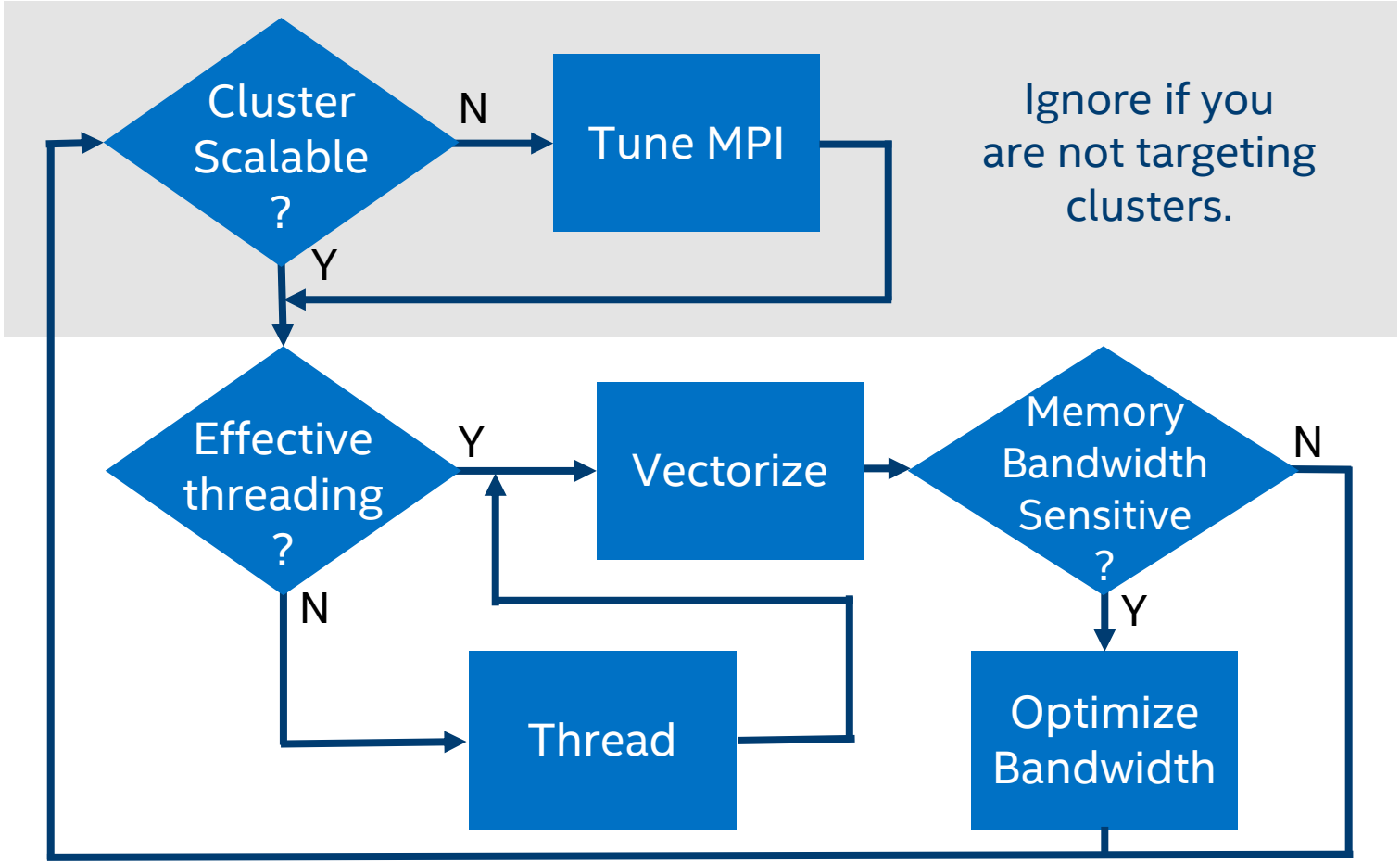
- **Learn from other experts via community product forums**
- **Access to a vast library** of self-help documents that build off decades of experience with creating high performance code.

WHICH TOOL SHOULD I USE?

Optimizing Performance on Parallel Hardware

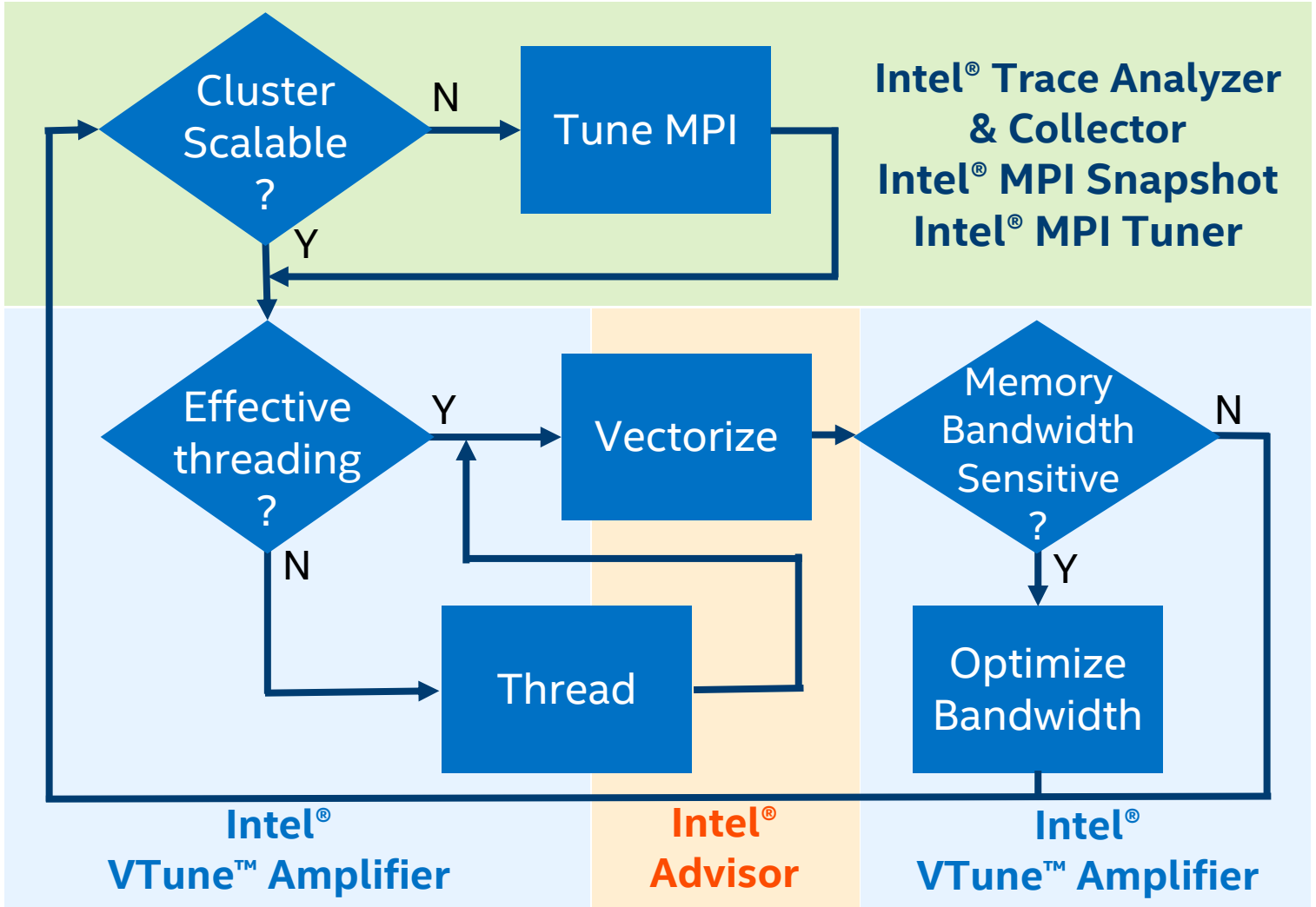
Intel® Parallel Studio XE

It's an Iterative Process…



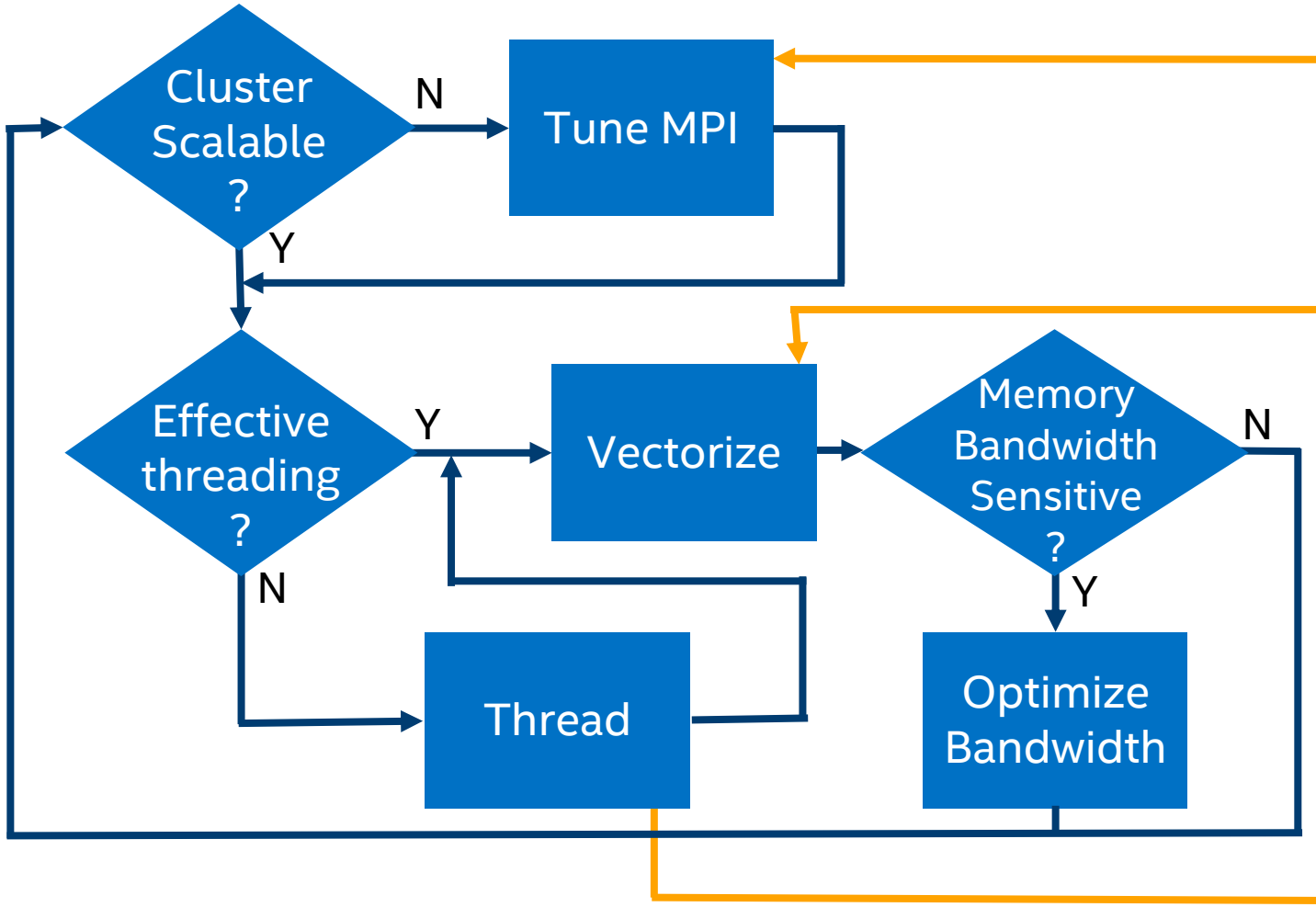
Performance Analysis Tools for Diagnosis

Intel® Parallel Studio XE



Tools for High-Performance Implementation

Intel® Parallel Studio XE



Intel® MPI Library & Benchmarks

Intel® Compiler

Intel® Math Kernel Library
Intel® Integrated Performance Primitives (Media & Data Library)
Intel® Data Analytics Acceleration Library
OpenMP*
Intel® Threading Building Blocks (Threading Library)

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 XE
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

What's New in Intel® Compilers 2018

Updates to All Versions

- **Advance Support for Intel® Architecture** – Use Intel compiler to generate optimized code for Intel Atom® through Intel® Xeon® Scalable and Xeon Phi™ processor families
- **Achieve Superior Parallel Performance** – Vectorize & thread your code (using OpenMP*) to take full advantage of the latest SIMD-enabled hardware, including AVX-512 instructions
- **Develop Smart Code with Confidence** – Access extensive compiler diagnostics to study code generation characteristics, use with Intel® VTune™ Amplifier & Intel® Advisor for further analysis
- **Faster Compile Time** – Memory management improvements reduce application compile time without sacrificing runtime performance
- **Lightweight Hardware-based Profile-guided Optimization** alternative – Experience many benefits of profile information without the overhead of instrumentation[1]

What's New in C++

Initial C++17, OpenMP* 5; full C++ 14 support

- Standards-driven parallelization for C++ developers

What's New in Fortran

Full Fortran 2008 support

- Submodules, BLOCK, superior coarray performance

Initial Fortran 2015 support (draft standard)

- Further C interoperability (ISO/IEC TS 29113:2012)

Full OpenMP* 4.5 support; initial OpenMP 5

- Thread & vectorize your code using standard APIs

[1]Requires Intel® VTune™ Amplifier

Faster Python* with Intel® Distribution for Python*

Advance Performance Closer to Native Code

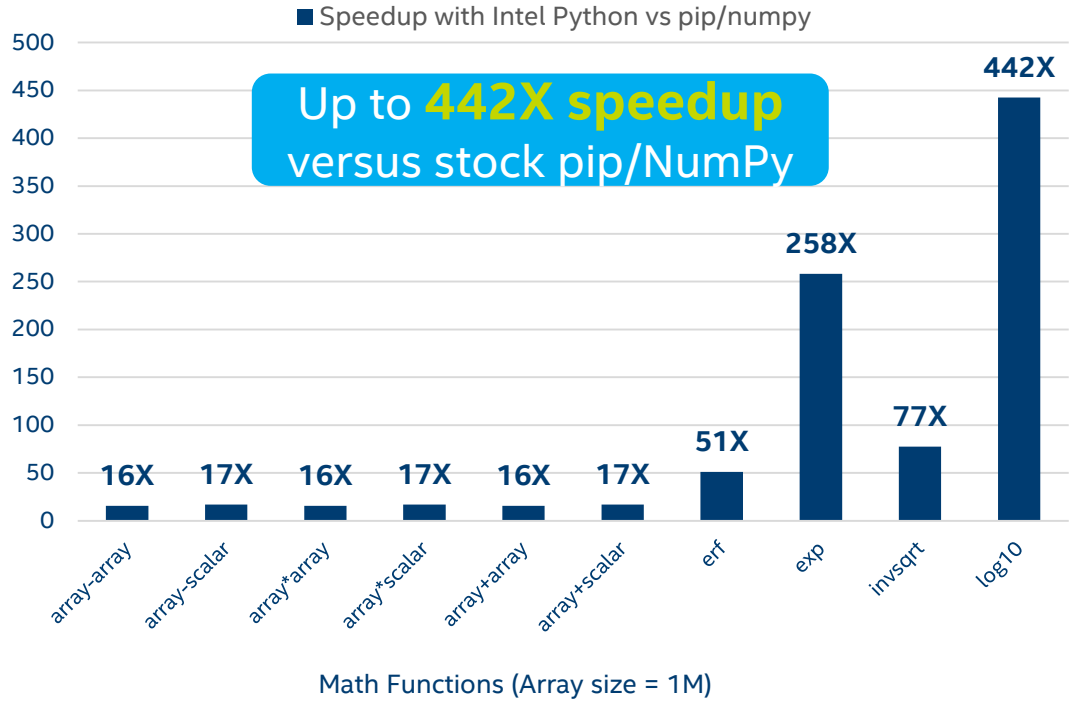
- Accelerated NumPy, SciPy, scikit-learn for scientific computing, machine learning & data analytics
- Drop-in replacement for existing Python - no code changes required
- Highly optimized for the latest Intel processors

What's New in the 2018 edition

- Updated to support Python 3.6
- Optimized scikit-learn for machine learning speedups
- Conda build recipes for custom infrastructure

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

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



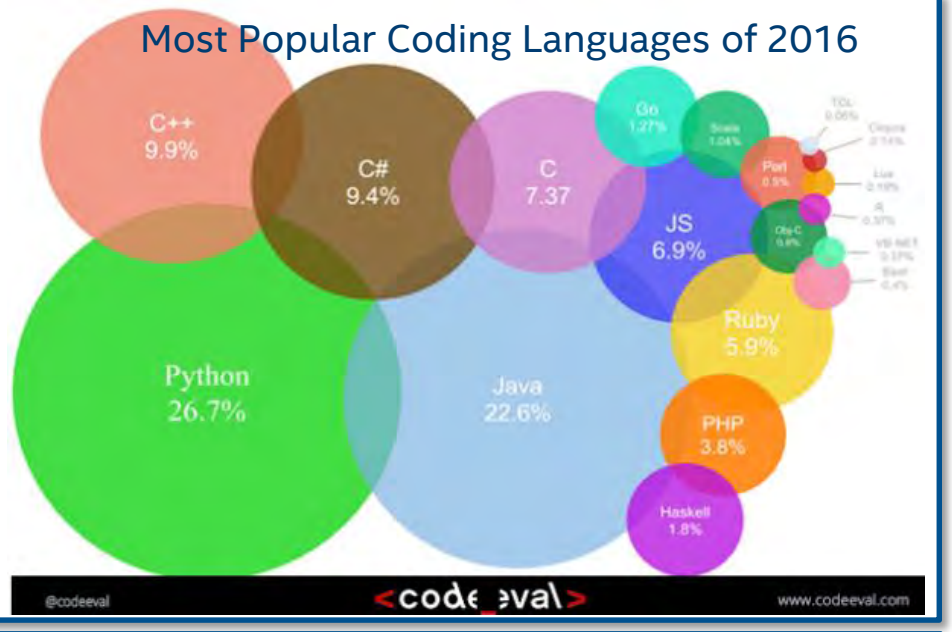
Configuration: Hardware: Intel® Xeon® CPU E5-2699 v4 @ 2.20GHz (2 sockets, 22 cores per socket, 1 thread per core – HT is off), 256GB DDR4 @ 2400MHz. Software: Stock: CentOS Linux* release 7.3.1611 (Core), python 3.6.2, pip 9.0.1, numpy 1.13.1, scipy 0.19.1, scikit-learn 0.19.0. Intel® Distribution for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

Python* Landscape

Intel® Distribution for Python*

Adoption of Python continues to grow among domain experts & developers for its productivity benefits



Challenge#1

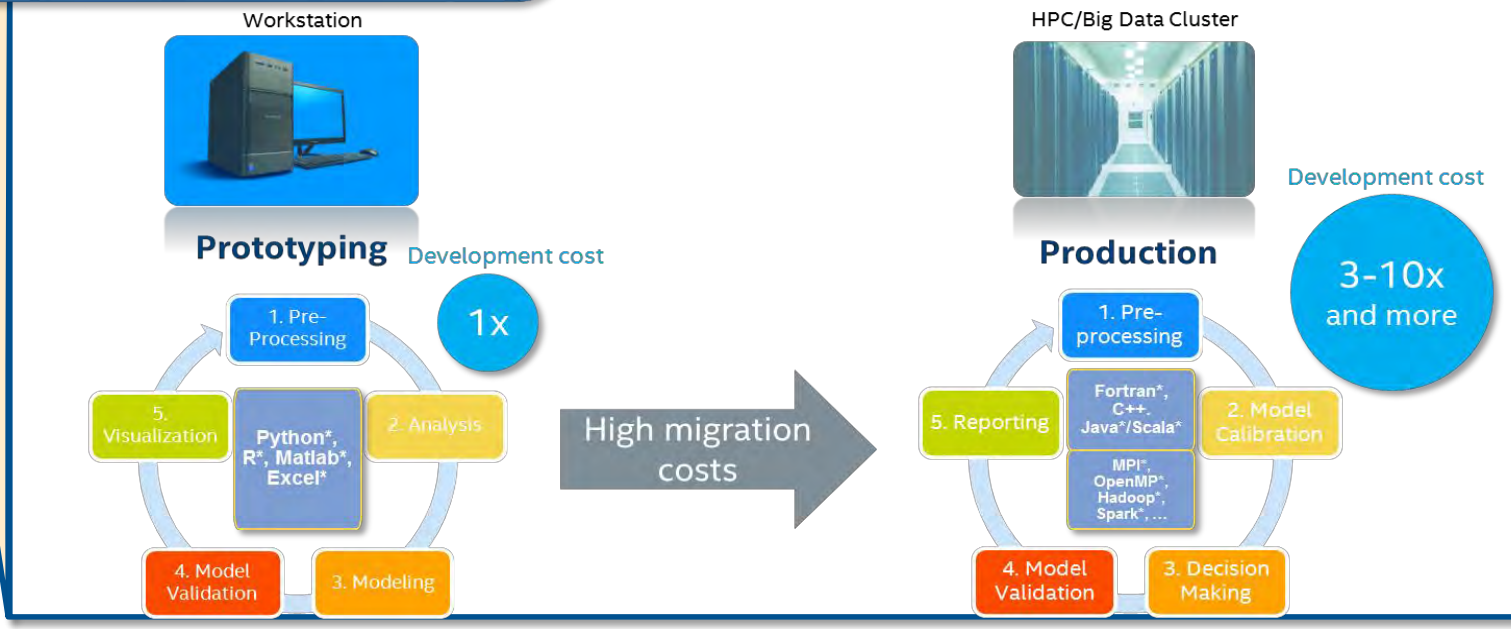
Domain experts are not professional software programmers

Challenge#2

Python performance limits migration to production systems

Intel's Python Tools

- Accelerate Python performance
- Enable easy access
- Empower the community



What's Inside Intel® Distribution for Python

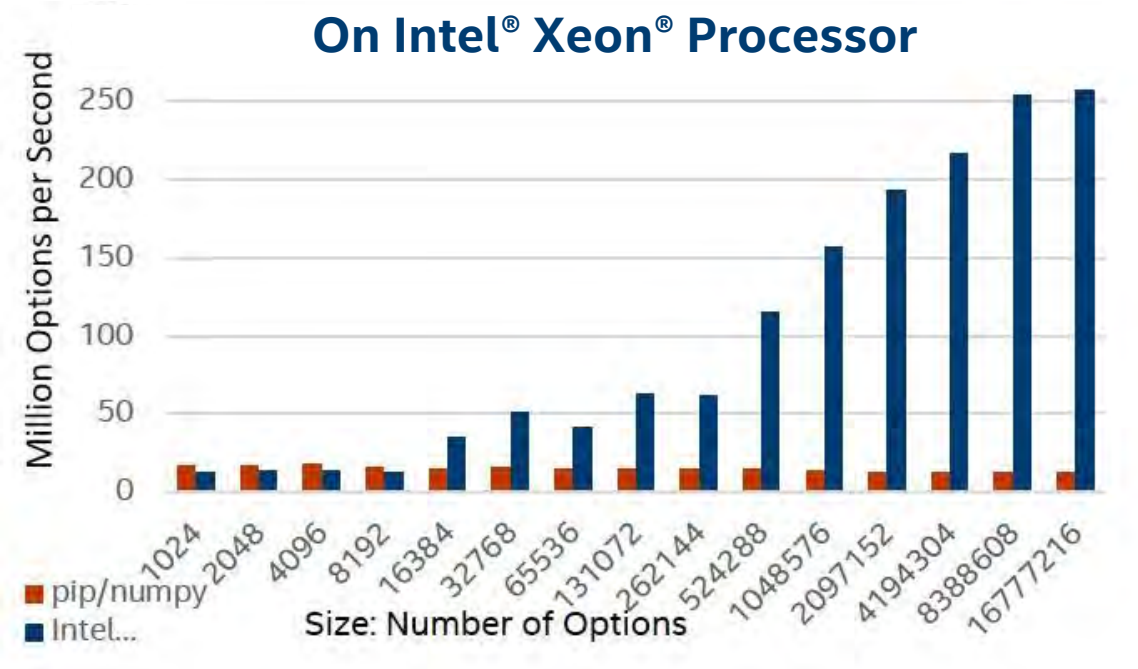
High Performance Python* for Scientific Computing, Data Analytics, Machine & Deep Learning

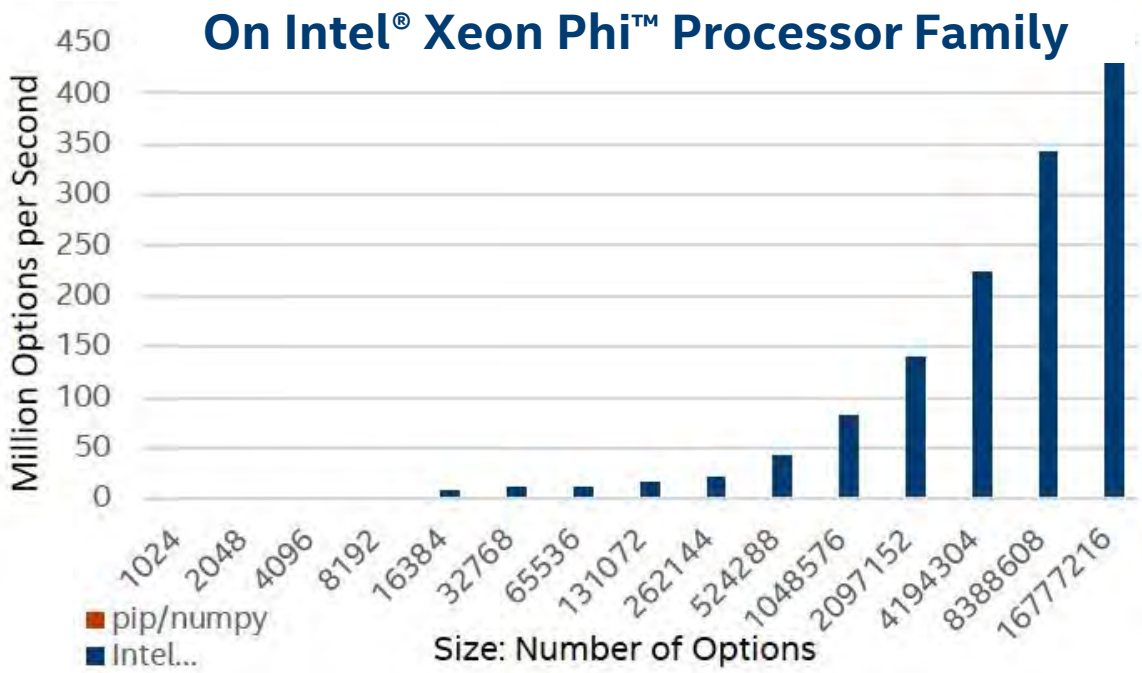
FASTER PERFORMANCE	GREATER PRODUCTIVITY	ECOSYSTEM COMPATIBILITY
Performance Libraries, Parallelism, Multithreading, Language Extensions	**Prebuilt & Accelerated Packages**	**Supports Conda & PIP**
Accelerated NumPy/SciPy/scikit-learn with Intel® MKL[1] & Intel® DAAL[2] Data analytics, machine learning & deep learning with scikit-learn, pyDAAL, Caffe*, Theano* Scale with Numba* & Cython* Includes optimized mpi4py, works with Dask* & PySpark* Optimized for latest Intel® architecture	Prebuilt & optimized packages for numerical computing, machine/deep learning, HPC, & data analytics Drop in replacement for existing Python - No code changes required Jupyter* notebooks, Matplotlib included Free download & free for all uses including commercial deployment	Compatible & powered by Anaconda*, supports conda & pip Distribution & individual optimized packages also available at conda & Anaconda.org, YUM/APT, Docker image on DockerHub Optimizations upstreamed to main Python trunk Priority Support through Intel® Parallel Studio XE
Intel® Architecture Platforms		
Operating System: Windows*, Linux*, MacOS[1]*		

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

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

Performance Speedups for Intel® Distribution for Python* for Black Scholes* Formula (Higher is Better)

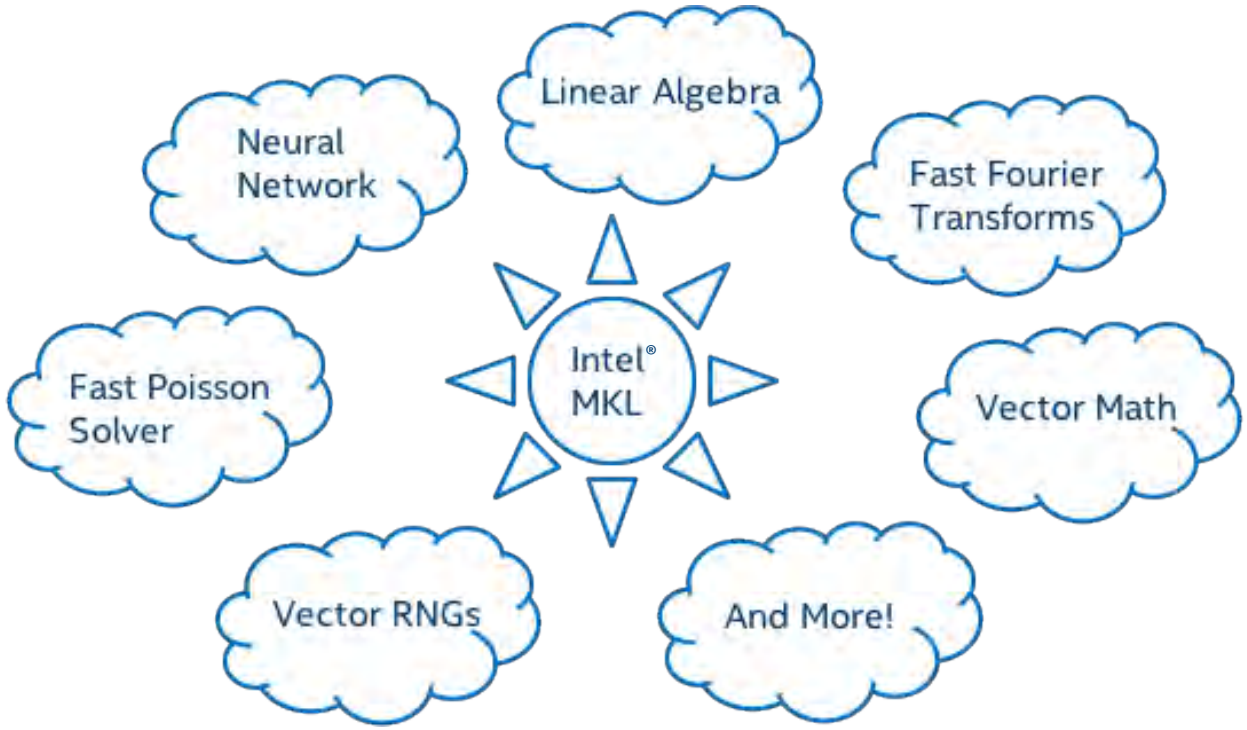




Configuration: Hardware: Intel® Xeon® CPU E5-2699 v4 @ 2.20GHz (2 sockets, 22 cores per socket, 1 thread per core – HT is off), 256GB DDR4 @ 2400MHz. Software: Stock: CentOS Linux* release 7.3.1611 (Core), python 3.6.2, pip 9.0.1, numpy 1.13.1, scipy 0.19.1, scikit-learn 0.19.0. Intel® Distribution for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

Configuration: Hardware: Intel® Xeon Phi™ CPU 7250 @ 1.40GHz (1 socket, 68 cores per socket, 4 threads per core), 192GB DDR4 @ 1200MHz, 16GB MCDRAM @ 7200MHz in cache mode. 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

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



- Highly optimized, threaded, & vectorized math functions that maximize performance on each processor family
- Utilizes industry-standard C and Fortran APIs for compatibility with popular BLAS, LAPACK, and FFTW functions—no code changes required
- Dispatches optimized code for each processor automatically without the need to branch code

What's New in the 2018 edition

- Improved small matrix multiplication performance in GEMM & LAPACK
- Improved ScaLAPACK performance for distributed computation
- 24 new vector math functions
- Simplified license for easier adoption & redistribution
- Additional distributions via YUM, APT-GET, & Conda

Learn More: software.intel.com/mkl

What's Inside Intel® Math Kernel Library

Accelerate HPC, Enterprise, Cloud & IoT Applications

Linear Algebra
- BLAS
- LAPACK
- ScaLAPACK
- Sparse BLAS
- Iterative sparse solvers
- PARDISO*
- Cluster Sparse Solver

FFTs
- Multidimensional
- FFTW interfaces
- Cluster FFT

Neural Networks
- Convolution
- Pooling
- Normalization
- ReLU
- Inner Product

Vector RNGs
- Congruential
- Wichmann-Hill
- Mersenne Twister
- Sobol
- Neiderreiter
- 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

Intel® Architecture Platforms

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

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

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

Accelerate Image, Signal, Data Processing & Cryptography computation tasks

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

What's New in 2018 edition

- Optimized functions for LZ4 data compression/decompression, a fast compression algorithm suitable for applications where speed is key - especially in communication channels
- Optimized functions for GraphicsMagick, a popular image processing toolbox, so customers using this function can achieve improved performance
- Added Platform aware APIs, which automatically detects whether image vectors and length are 32-bit or 64-bit and abstracts this away from the users

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

What's Inside Intel® Integrated Performance Primitives

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

Image Domain
- Image Processing
- Computer Vision
- Color Conversion

Signal Domain
- Signal Processing
- Vector Math

Data Domain
- Data Compression
- Cryptography
- String Processing

Intel® Architecture Platforms

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

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

Get the Benefits of Advanced Threading with Intel® Threading Building Blocks

Use Threading Techniques to fully Leverage Multicore Performance & Heterogeneous Computing

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

What's New in 2018 edition

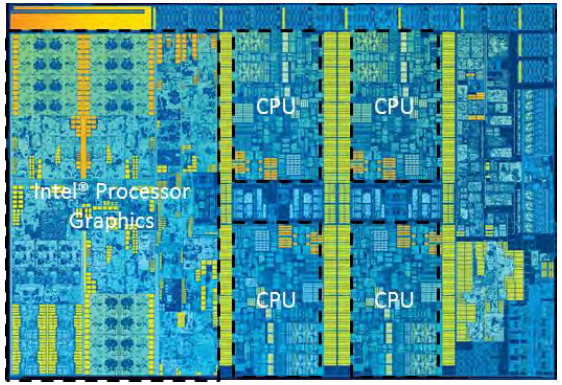
- New capabilities in Flow Graph improve concurrency and heterogeneity
- Improves insight into parallelism inefficiencies for Intel® VTune Amplifier 2018
- Support for Cmake file

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

Heterogeneous Support

Intel® Threading Building Blocks (Intel® TBB)

Intel® TBB flow graph as a coordination layer for heterogeneity—retains optimization opportunities and composes with existing models



CPUs, integrated GPUs, FPGAs, etc.

+

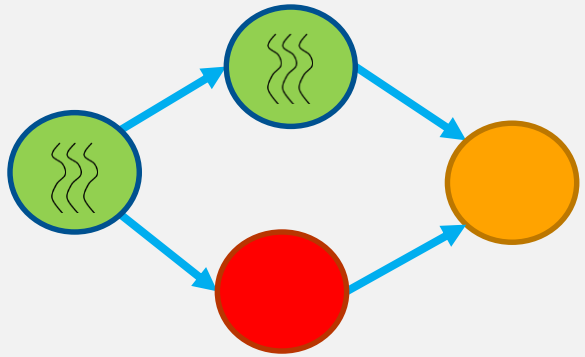
Intel® Threading Building Blocks
OpenVX*
OpenCL*
COI/SCIF
....

Intel® TBB as a **composability layer** for library implementations

- One threading engine ***underneath*** all CPU-side work

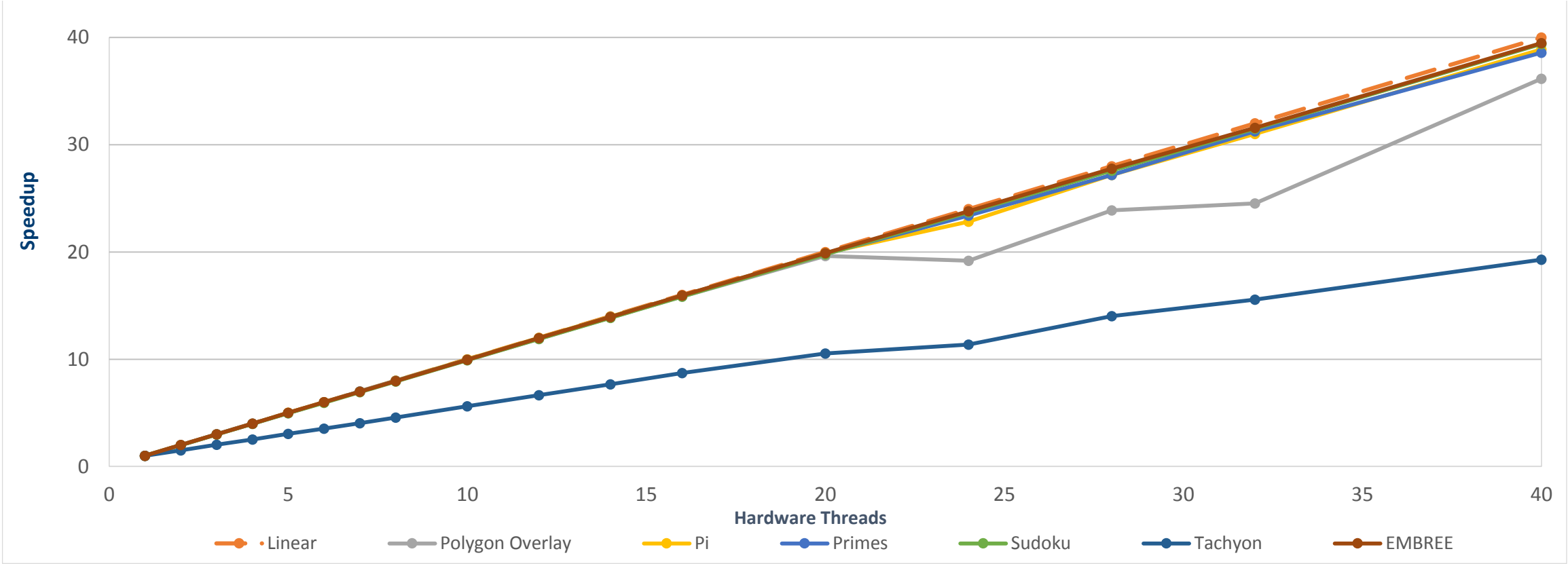
Intel TBB flow graph as a **coordination layer**

- Be the glue that connects heterogenous hardware and software together
- Expose parallelism between blocks—simplify integration



Excellent Performance Scalability with Intel® TBB on Intel® Xeon® Processor

Intel® Threading Building Blocks 2018



Configuration: Software versions: Intel® C++ Intel® 64 Compiler, Version 17.4, Intel® Threading Building Blocks 2018 (Intel® TBB); Hardware: 2x Intel® Xeon® CPU E5-2699 v4@ 2.20GHz 44/T), 128GB Main Memory; Operating System: Red Hat Enterprise Linux Server* 7.2 (Maipo), kernel 3.10.0-327.4.5.el7.x86_64; Note: sudoku, primes and tachyon are included with Intel® TBB. 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. Benchmarks Source: Intel Corporation - performance measured in Intel labs by Intel employees. 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

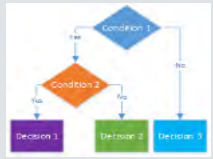
Speedup Analytics & Machine Learning with Intel® Data Analytics Acceleration Library (Intel® DAAL)

- Highly tuned functions for classical machine learning and analytics performance across a spectrum of Intel® architecture devices
- Optimizes data ingestion together with algorithmic computation for highest analytics throughput
- Includes Python*, C++, Java* APIs, and connectors to popular data sources including Spark* and Hadoop*

Learn More: software.intel.com/daal

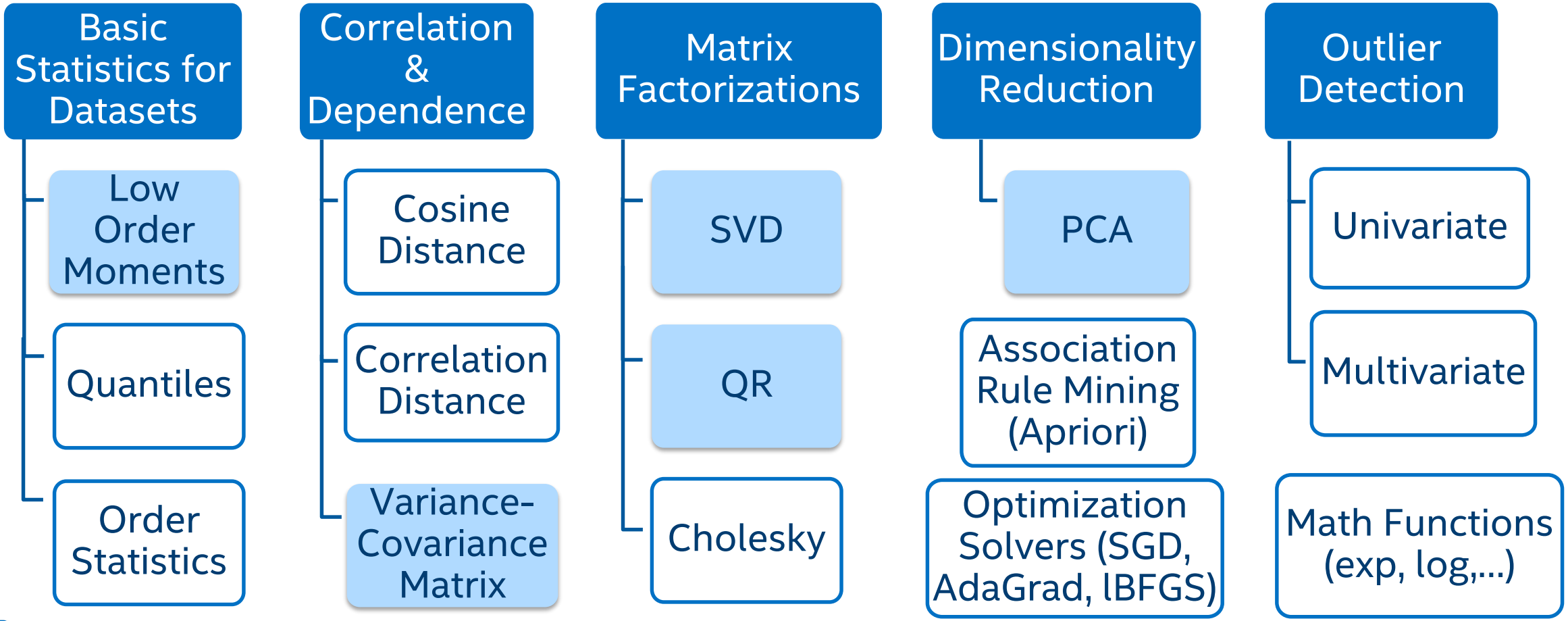
What's New in the 2018 Edition

- New Algorithms
  - Classification & Regression Decision Tree and Forest
  - k-NN
  - Ridge Regression
- Spark* MLlib-compatible API wrappers for easy substitution of faster Intel® DAAL functions
- Improved APIs for ease of use
- Repository distribution via YUM, APT-GET, and Conda

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.

Algorithms, Data Transformation & Analysis

Intel® Data Analytics Acceleration Library

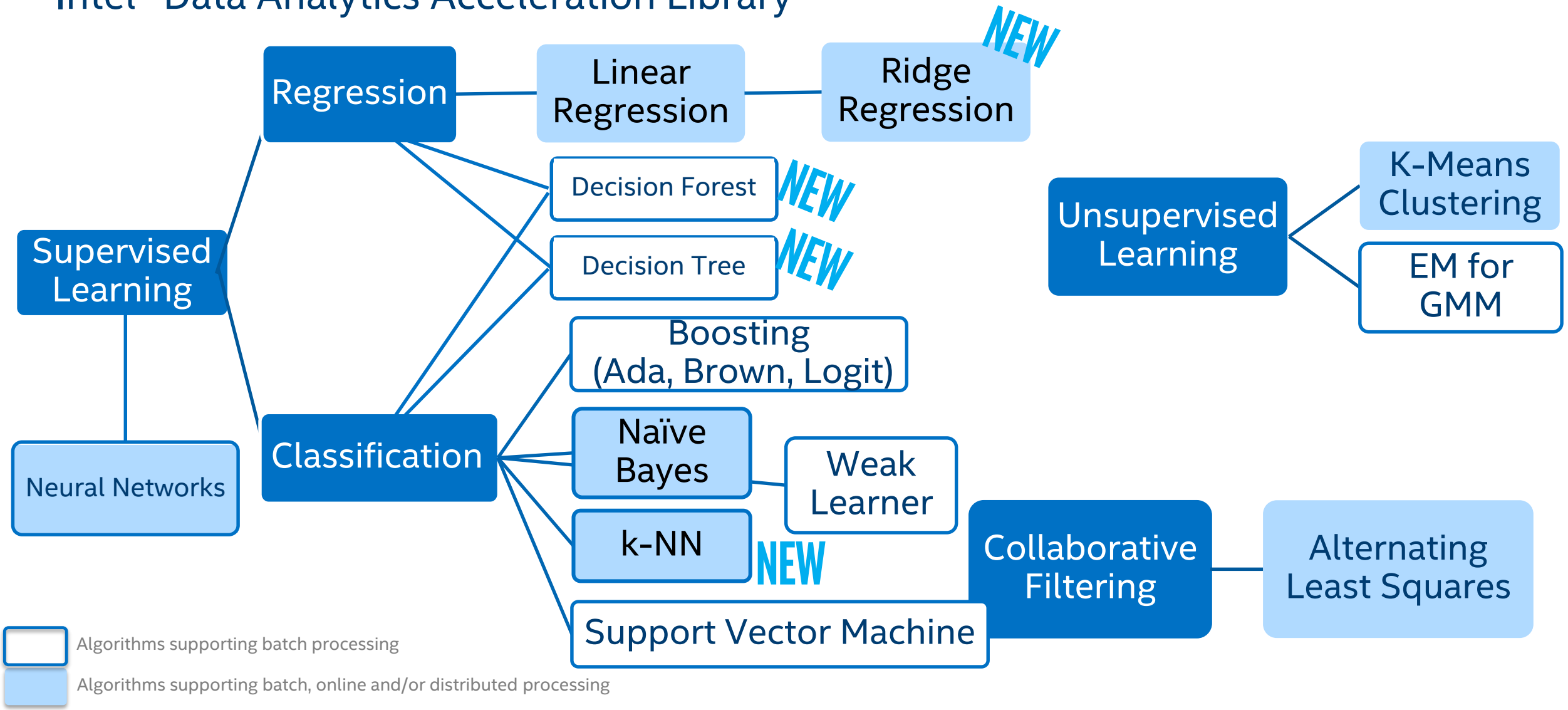


Algorithms supporting batch processing

Algorithms supporting batch, online and/or distributed processing

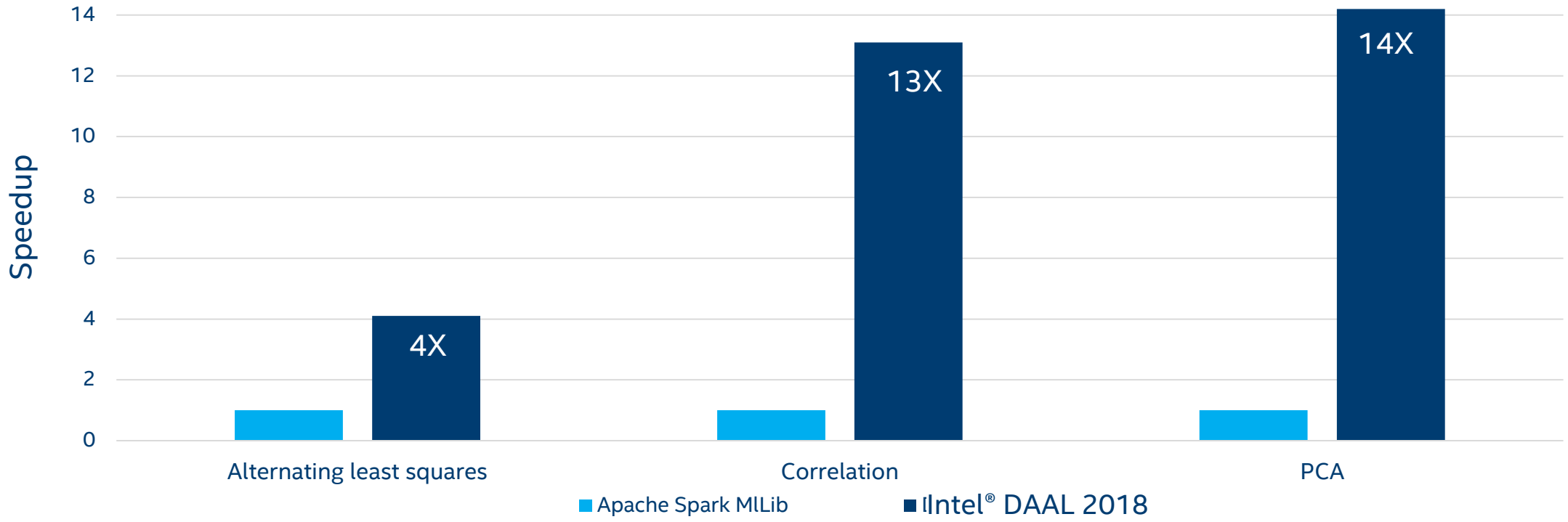
Algorithms & Machine Learning

Intel® Data Analytics Acceleration Library



Intel® DAAL 2018 vs Apache Spark* MlLib Performance

Intel® Data Analytics Acceleration Library (Intel® DAAL)



Configuration: 2x Intel® Xeon® E5-2660 CPU @ 2.60GHz, 128 GB, Intel® DAAL 2018; Alternating Least Squares – Users=1M Products=1M Ratings=10M Factors=100 Iterations=1 MLLib time=165.9 sec DAAL time=40.5 sec Gain=4.1x; Correlation – N=1M P=2000 size=37 GB Mllib time=169.2 sec DAAL=12.9 sec Gain=13.1x; PCA – n=10M p=1000 Partitions=360 Size=75 GB Mllib=246.6 sec DAAL (seq)=17.4 sec Gain=14.2x
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. Source: Intel Corporation - performance measured in Intel labs by Intel employees. Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804.

INTEL® 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 XE
Intel® Advisor
Intel® Inspector

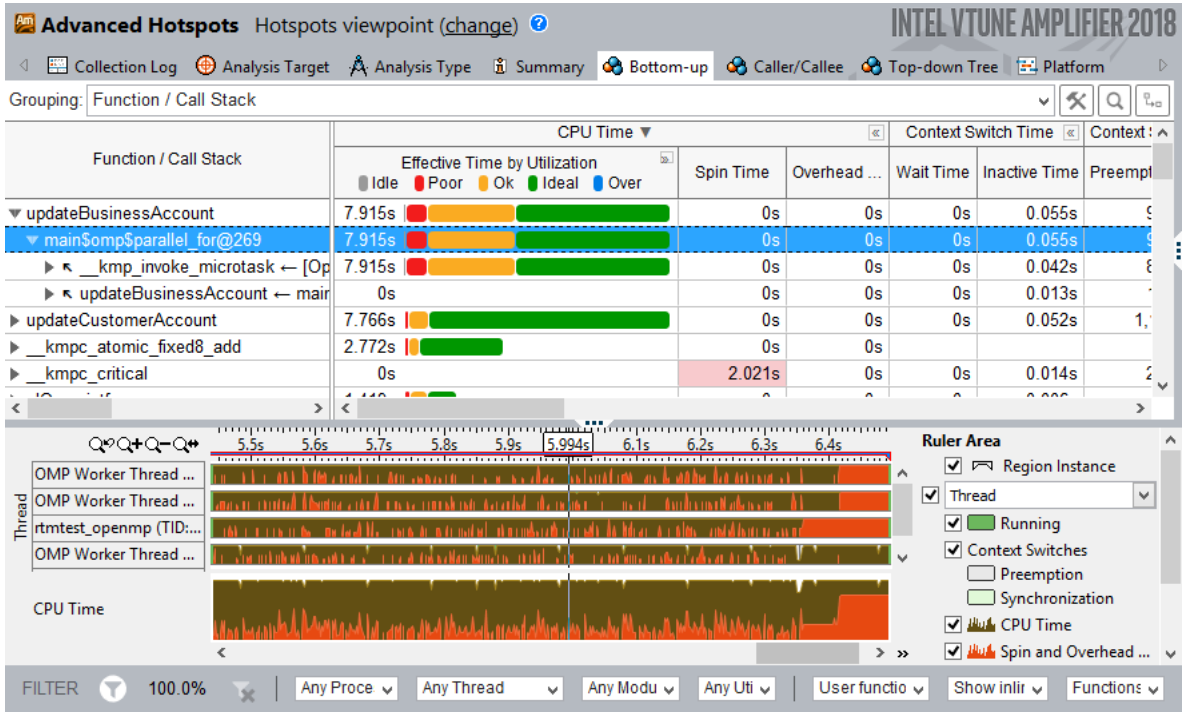
Part of the Professional Edition

SCALE

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

Part of the Cluster Edition

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



Save Time Optimizing Code

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

New for 2018 edition (partial list)

- Quick metrics for shared & distributed memory apps
- Cross-OS analysis – e.g. analyze Linux* from Windows* or macOS*
- Profile inside containers

Learn More: software.intel.com/intel-vtune-amplifier-xe

Rich Set of Profiling Features for Multiple Markets

Intel® VTune™ Amplifier—Performance Profiler

Basic Profiling

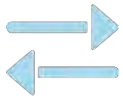
- Hotspots

Threading Analysis

- Concurrency, Locks & Waits
- OpenMP, Intel® Threading Building Blocks

Micro Architecture Analysis

- Cache, branch prediction, …

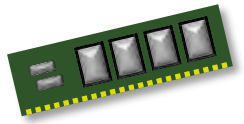


Vectorization + Intel® Advisor

- FLOPS estimates

MPI + Intel® Trace Analyzer & Collector

- Scalability, imbalance, overhead



Use Memory Efficiently

- Tune data structures & NUMA

Optimize for High Speed Storage

- I/O and compute imbalance

Intel® Media SDK Integration

- Meaningful media stack metrics

Low Overhead Java*, Python*, Go*

- Managed + native code

Containers

- Docker*, Mesos*, LXC*

Optimize Private Cloud-Based Applications

Profile Native & Java* Apps in Containers—Intel® VTune™ Amplifier

Profile Enterprise Applications

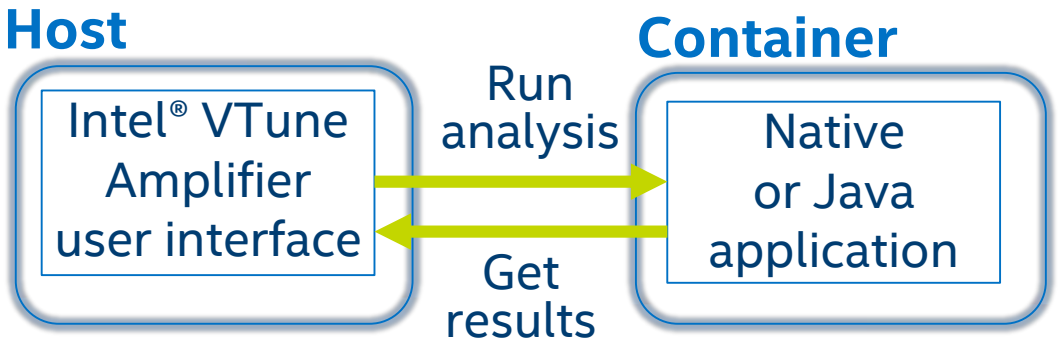
- Native C, C++, Fortran
- Attach to running Java services (e.g., Mail)
- Profile Java daemons without restart

Accurate Low-overhead Data Collection

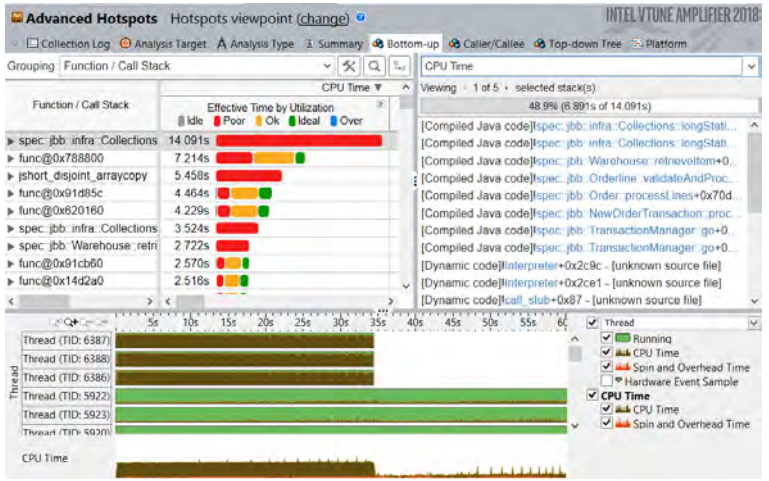
- Advanced hotspots and hardware events
- Memory analysis
- Accurate stack information for Java and HHVM

Popular Containers Supported

- Docker*
- Mesos*
- LXC*



- No container configuration required
- Detection of container is automatic



Software collectors (e.g. Locks & Waits) and Python profiling are not currently available for containers.

Application Performance Snapshot Adds MPI

Data in One Place: MPI+OpenMP+Memory Floating Point—Intel® VTune™ Amplifier

Quick & Easy Performance Overview

- Does the app need performance tuning?

MPI & non-MPI Apps†

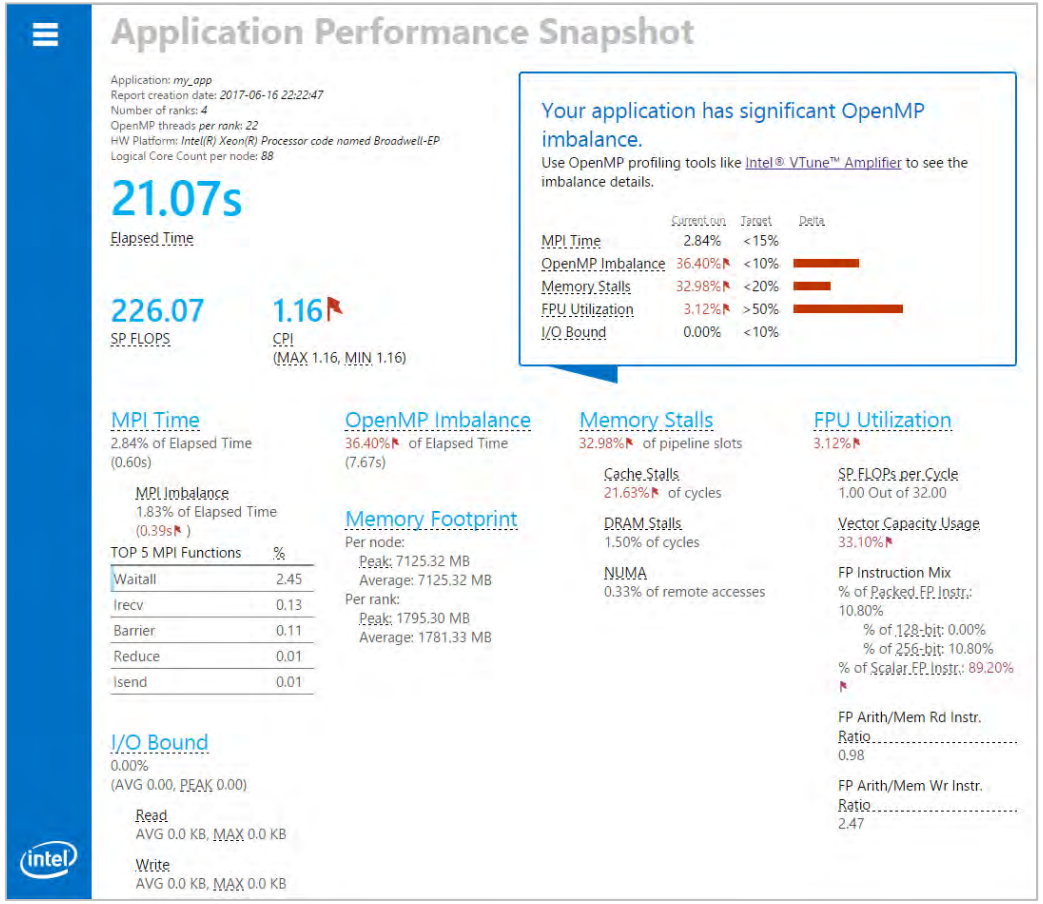
- Distributed MPI with or without threading
- Shared memory applications

Popular MPI Implementations Supported

- Intel® MPI Library
- MPICH & Cray MPI

Richer Metrics on Computation Efficiency

- CPU (processor stalls, memory access)
- FPU (vectorization metrics)

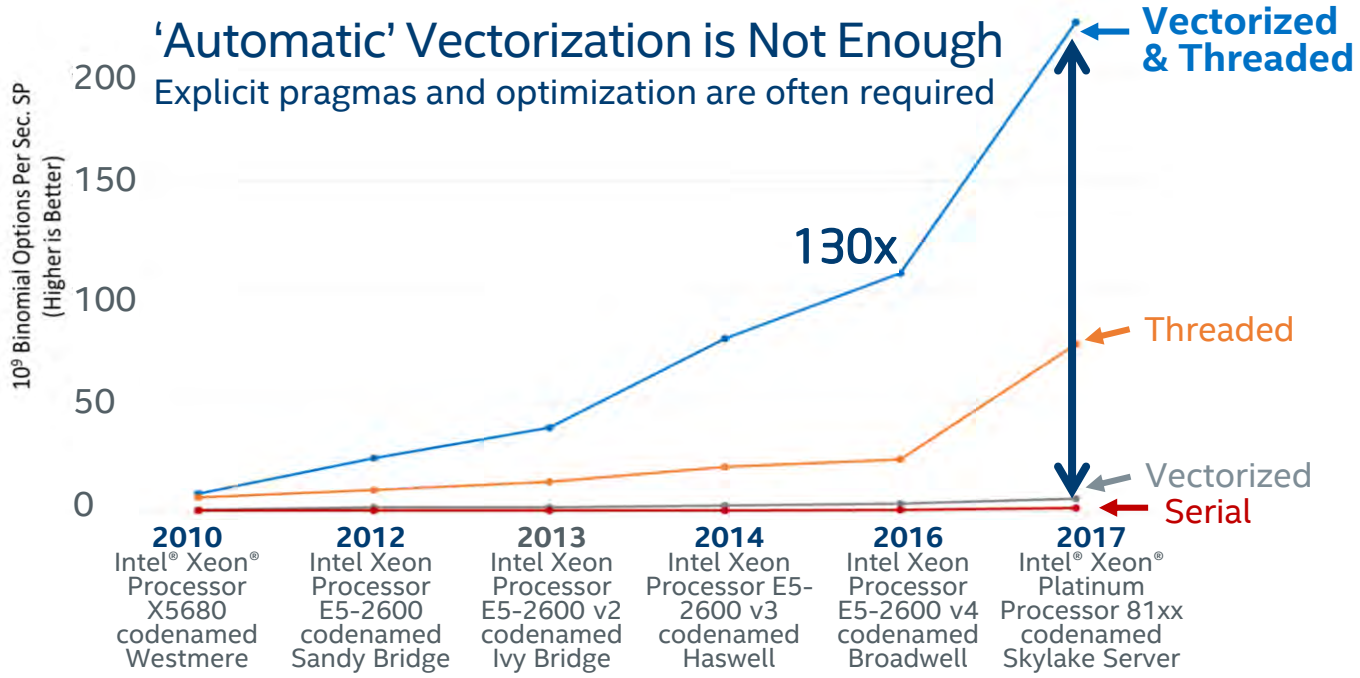


†MPI supported only on Linux*

Modernize Your Code with Intel® Advisor

Optimize Vectorization & Prototype Threading

Performance Increases Scale with Each New Hardware Generation



Modern Performant Code

- Vectorized (uses Intel® AVX-512/AVX)
- Efficient memory access
- Threaded

Intel® Advisor

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

New for 2018 edition (partial list)

- Roofline analysis
- Targeted data collection
- More recommendations

Learn More: http: intel.ly/advisor-xe

See Vectorize & Thread or Performance Dies Configurations for 2010-2017 Benchmarks in Backup. 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. Benchmarks Source: Intel Corporation - performance measured in Intel labs by Intel employees.

‘Automatic’ Vectorization is Often Not Enough

A good compiler can still benefit greatly from vectorization optimization—Intel® Advisor

Compiler will not always vectorize

- Check for Loop Carried Dependencies using Intel® Advisor
- All clear? Force vectorization. C++ use: pragma simd, Fortran use: SIMD directive

Not all vectorization is efficient vectorization

- Stride of 1 is more cache efficient than stride of 2 & greater. Analyze with Intel® Advisor
- Consider data layout changes Intel® SIMD Data Layout Templates can help

Benchmarks on prior slides did not all ‘auto vectorize.’ Compiler directives were used to force vectorization & get more performance.

Arrays of structures are great for intuitively organizing data, but are less efficient than structures of arrays. Use Intel® SIMD Data Layout Templates to map data into a more efficient layout for vectorization.

Get Breakthrough Vectorization Performance

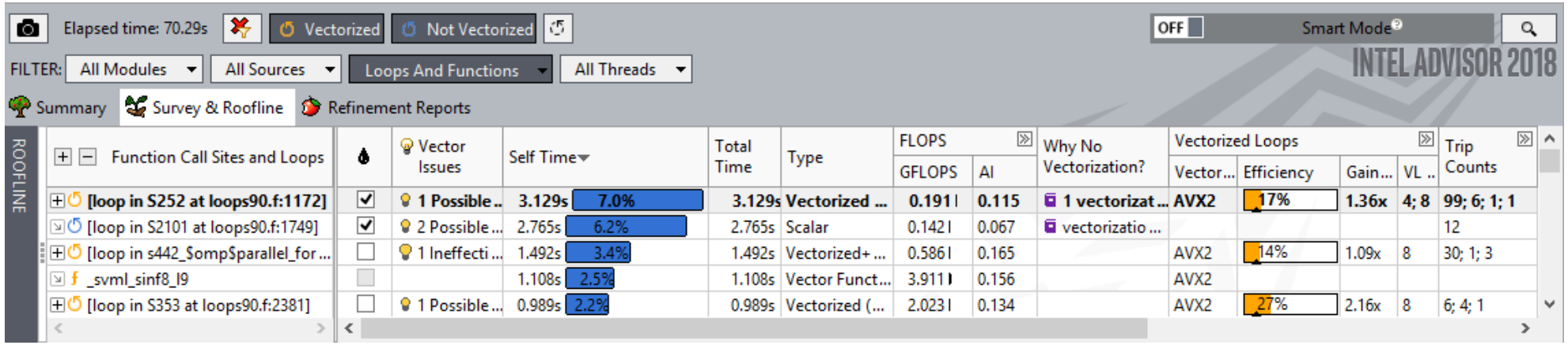
Intel® Advisor—Vectorization Advisor

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



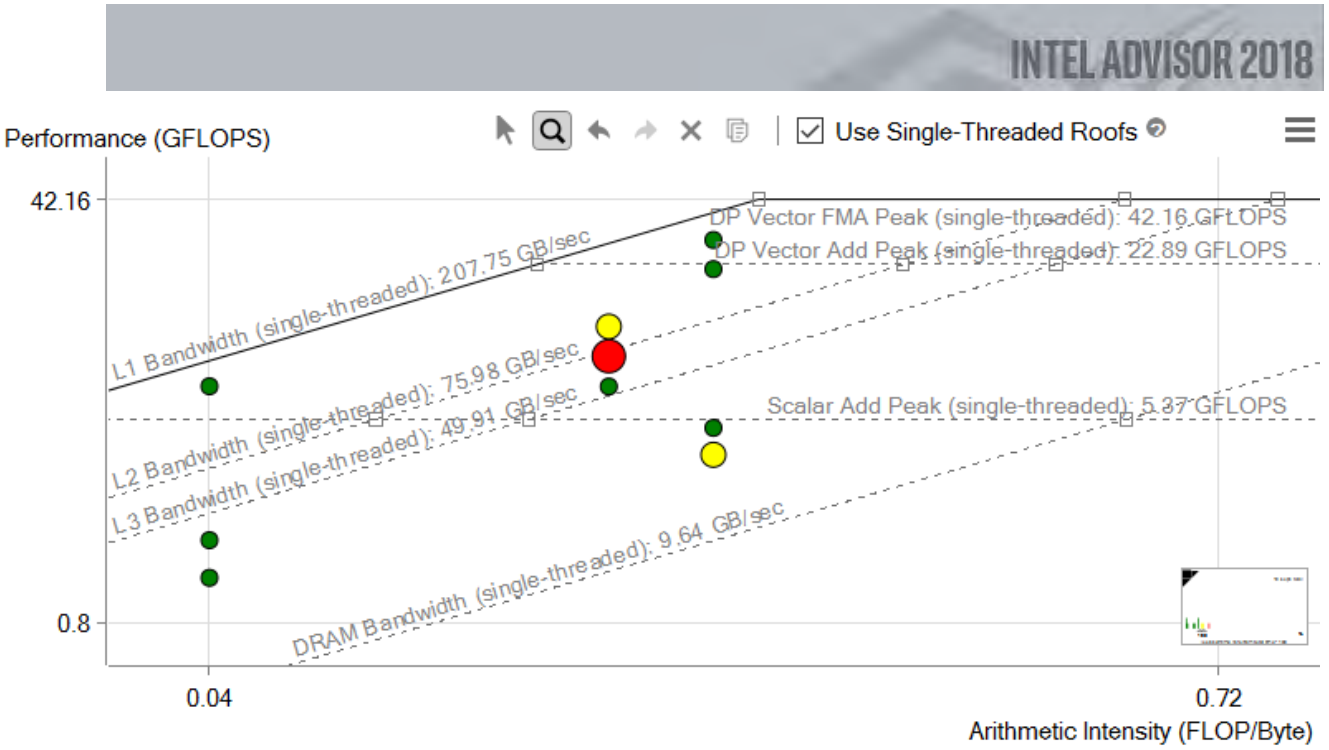
Optimize for Intel® AVX-512 with or without access to AVX-512 hardware

Find Effective Optimization Strategies

Cache-aware Roofline Analysis—Intel® Advisor

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



Design It, Tune, Debug, Then Implement

Design with Disrupting Development—Intel® Advisor Thread Prototyping

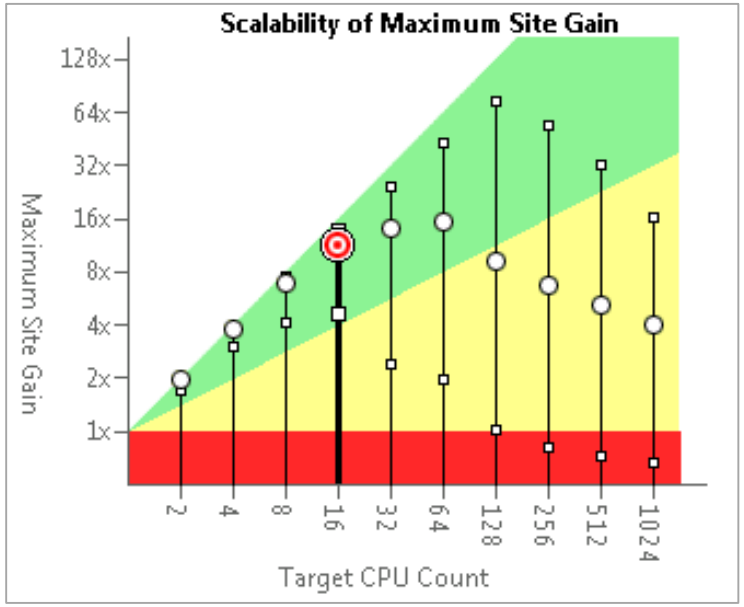
Have You

- Threaded an app, but seen little benefit?
- Hit a “scalability barrier?”
- Delayed release due to synchronization errors?

Data Driven Threading Design

- Quickly prototype multiple options
- Project scaling on larger systems
- Find synchronization errors before implementing threading
- Design without disrupting development

**Add Parallelism with Less Effort,
Less Risk & More Impact**



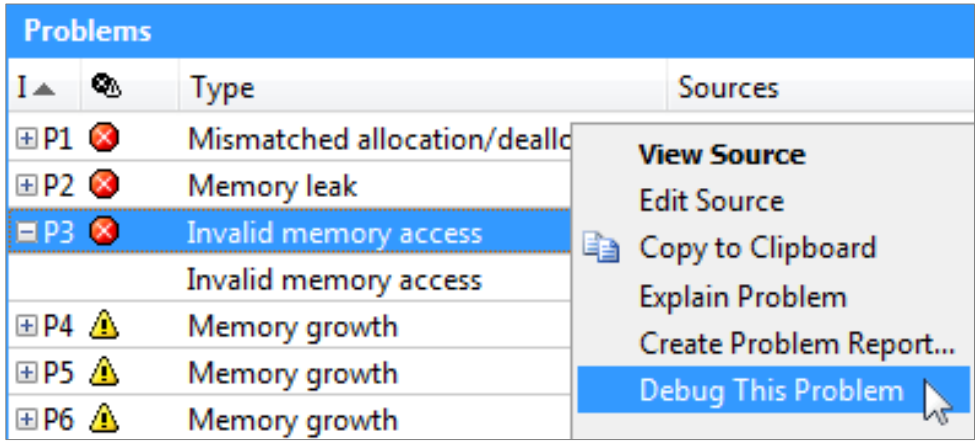
“**Intel® Advisor** allowed us to quickly prototype ideas for parallelism, saving developer time and effort”

Simon Hammond
Senior Technical Staff
Sandia National Laboratories

Debug Memory & Threading with Intel® Inspector

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

Debugger Breakpoints



Diagnose in hours instead of months

Learn More: intel.ly/inspector-xe

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 2018 edition

- Fewer false positives
- C++ 17 std::shared_mutex added
- Windows SRW Locks added

[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*

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 XE
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

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 2018 edition[1]

- Up to **11x** faster in job start-up time
- Up to **25%** reduction in job finalization time
- Supports the latest Intel® Xeon® Scalable processor

Learn More: software.intel.com/intel-mpi-library

[1]See following benchmarks slide for more details

Intel® MPI Library Features

Optimized MPI Application Performance

- Application-specific tuning
- Automatic tuning
- Support for latest Intel® Xeon® & Intel® Xeon Phi™ Processors
- Support for Intel® Omni-Path Architecture Fabric

Multi-vendor Interoperability & Lower Latency

- Performance optimized support for the fabric capabilities through OpenFabrics* (OFI)
- Industry leading latency

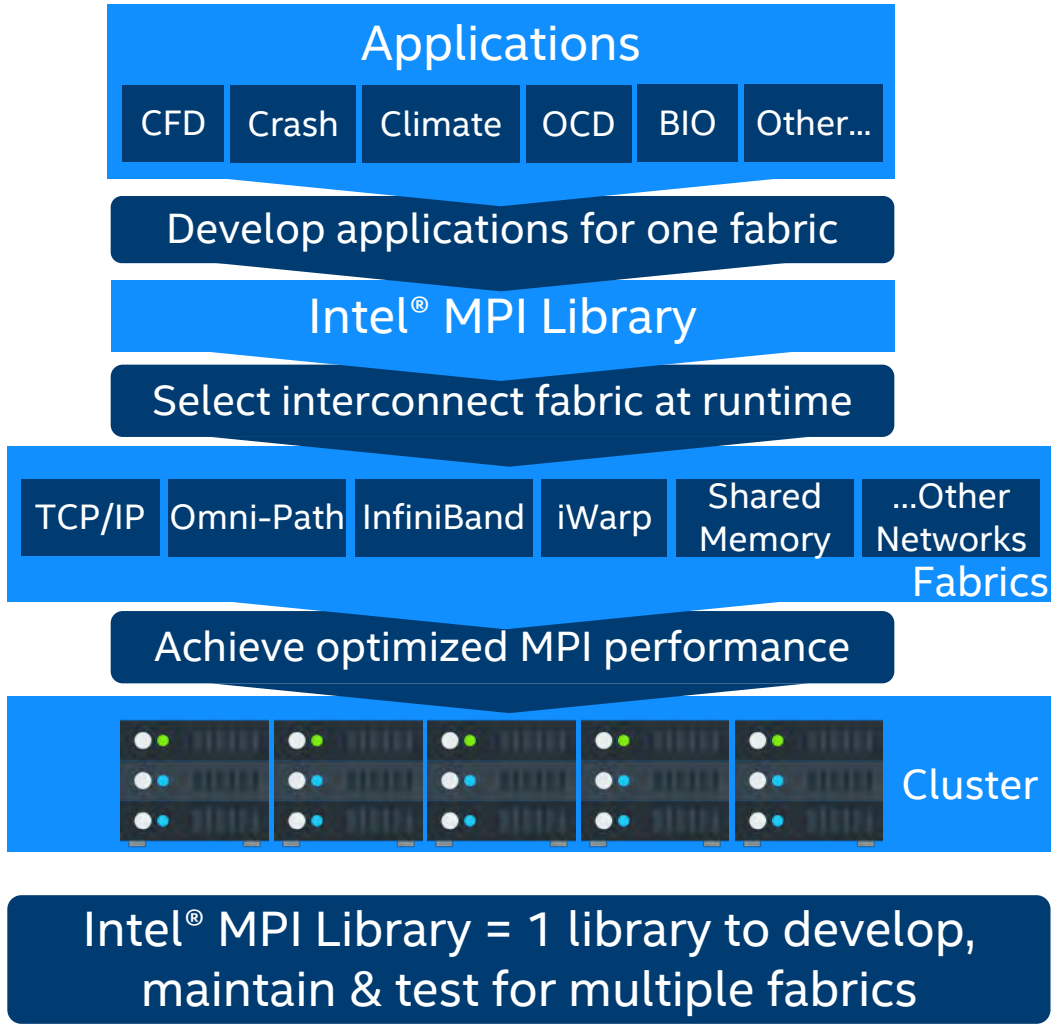
Faster MPI Communication - Optimized collectives

Sustainable Scalability

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

More Robust MPI Applications

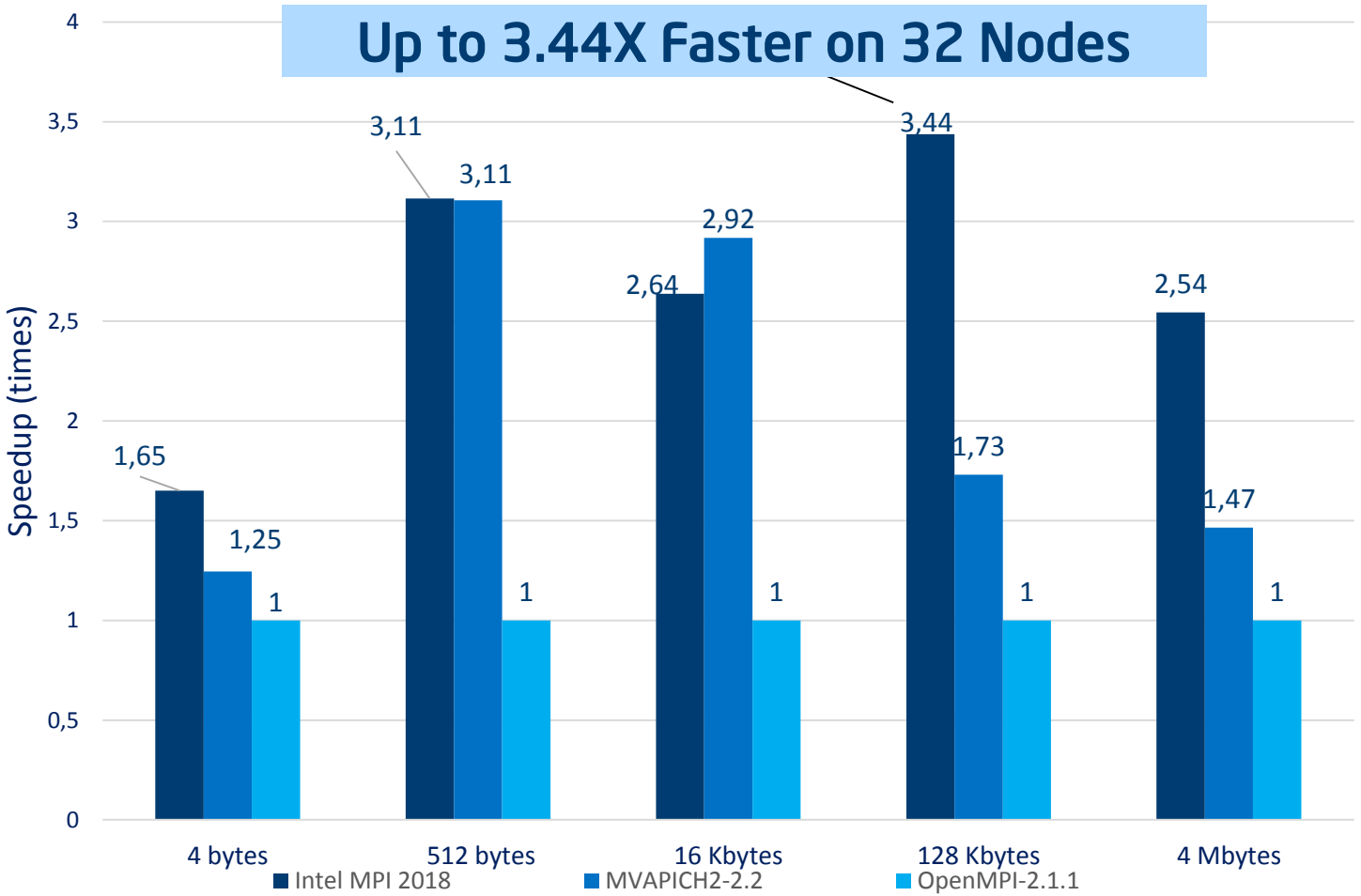
Seamless interoperability with Intel® Trace Analyzer & Collector



Superior MPI Performance with Intel® MPI Library 2018 on Linux* 64

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

MPI Latency Benchmarks (Higher is Better)



Configuration: Hardware: Intel(R) Xeon(R) Gold 6148 CPU @ 2.40GHz; 192 GB RAM. Interconnect: Intel® Corporation Omni-Path HFI Silicon 100 Series [discrete]. Software:RHEL* 7.3; IFS 10.2.0.0.158; Libfabric 1.3.0; Intel® MPI Library 2018 (I_MPI_FABRICS=shm:ofi); Intel® MPI Benchmarks 2018 (built with Intel® C++ Compiler XE 18.0.0 for Linux*). 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. 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 micro-processors. 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 micro-architecture 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

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 2018 edition

- Support of OpenSHMEM* applications
- Supports the latest Intel® Xeon® Scalable and Intel® Xeon Phi™ processors

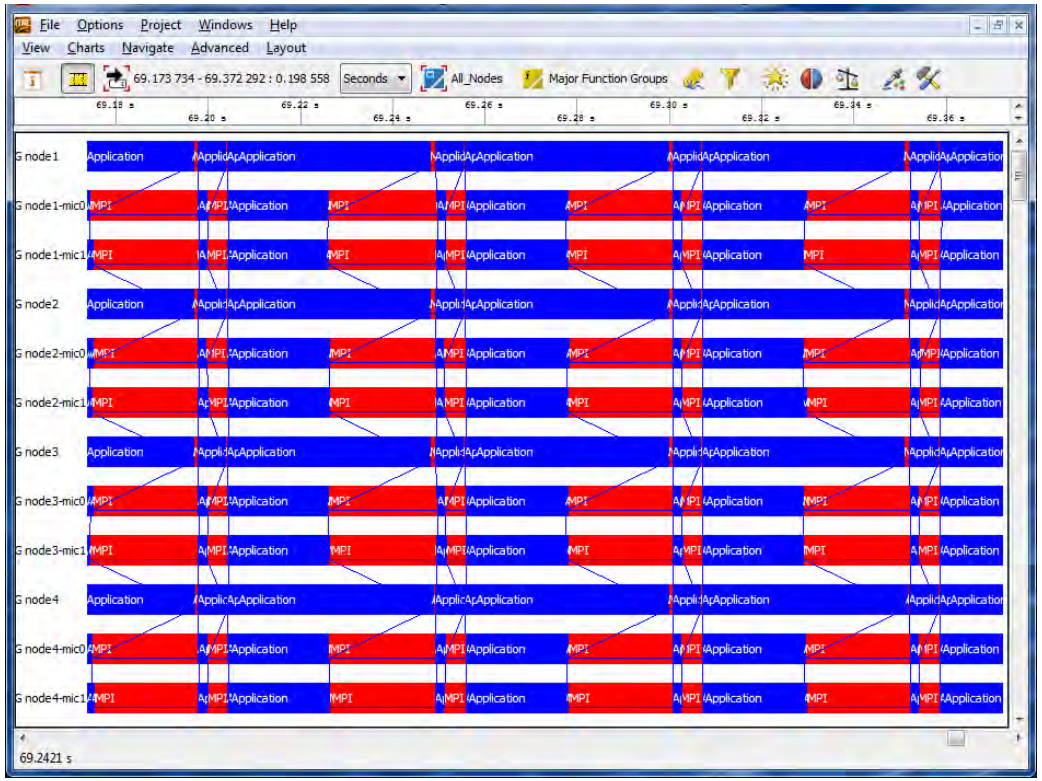
Learn More: software.intel.com/intel-trace-analyzer

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

Intel® Cluster Checker 2018

For Linux* High Performance Compute Clusters

Clusters are Complex Systems!

Challenge is to reduce this complexity barrier for

- Application developers
- Cluster architects
- Cluster users
- System administrators

Intel® Cluster Checker is an expert system approach that provides cluster systems expertise

- Verifies system health
- Offers suggested actions
- Provides extensible framework
- API for integrated support

What's New in Intel® Cluster Checker 2018

Ensure Your HPC Cluster Components Work Together

New Features Improve Usability & Checking Capabilities

- Adds support for new Intel silicon & platform elements (processors, fabric, memory, storage, cluster provisioning, HPC platforms)
- Introduces simplified grouping of checks for extensibility
- Improves diagnostic output
- Validates Intel® Scalable System Framework Classic HPC Cluster Reference Architectures
- Check Intel® Omni-Path in-depth
- Analyze data from multiple database sources

CODE THAT PERFORMS AND OUTPERFORMS

Download a *free*, 30-day trial of
Intel® Parallel Studio XE 2018 today

https://software.intel.com/en-us/intel-parallel-studio-xe/try-buy

AND DON'T FORGET...

To check your inbox for the evaluation survey which will be emailed after this presentation.

P.S.

Everyone who fills out the survey will receive a personalized certificate indicating completion of the training!


intel
Software