CODE THAT PERFORMS
INTEL® PARALLEL STUDIO XE 2018

Accelerate HPC, Enterprise & Cloud Applications
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*.
What’s Inside Intel® Parallel Studio XE
Comprehensive Software Development Tool Suite

**COMPOSER EDITION**

**BUILD**
Compilers & Libraries

- C / C++ Compiler
- Fortran Compiler
- Intel® Threading Building Blocks

**ANALYZE**
Analysis Tools

- Intel® VTune™ Amplifier
- Intel® Inspector
- Intel® Advisor

**SCALE**
Cluster Tools

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

**PROFESSIONAL EDITION**

- Intel® Math Kernel Library
- Intel® Integrated Performance Primitives
- Intel® Data Analytics Acceleration Library

- Intel® VTune™ Amplifier Performance Profiler
- Intel® Inspector Memory & Thread Debugger
- Intel® Advisor Vectorization Optimization & Thread Prototyping

**CLUSTER EDITION**

- Intel® MPI Library Message Passing Interface Library
- Intel® Trace Analyzer & Collector MPI Tuning & Analysis
- Intel® Cluster Checker Cluster Diagnostic Expert System

**Operating System:** Windows*, Linux*, MacOS

**Intel® Architecture Platforms**

- Intel® Distribution for Python*
- High Performance Scripting

---


---

Optimization Notice

Copyright © 2017, Intel Corporation. All rights reserved.

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

---

1Available only in Intel® Parallel Studio XE Composer Edition.
How your Business can Benefit from More Performance

- Use the full power, get more performance from Intel hardware
- Speed applications and workload processing
- Increase efficiency and developer productivity
- Solve business challenges, fuel innovation
- Scale forward, drive compatibility and interoperability

Intel® Parallel Studio XE

- Boosts performance on today’s and future Intel® platforms
- Simplifies creating high performance, scalable, reliable parallel code with less effort

3 Editions are Available – Composer, Professional & Cluster
Download a Free Trial
# 3 Editions to Meet Your Needs

**Intel® Parallel Studio XE**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BUILD</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intel® C++ Compiler</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Intel® Fortran Compiler</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Intel® Distribution for Python*</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Intel® Math Kernel Library – fast math library</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Intel® Integrated Performance Primitives – image, signal &amp; data processing</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Intel® Threading Building Blocks – C++ threading library</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Intel® Data Analytics Acceleration Library – machine learning &amp; analytics</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

| **ANALYZE**     |                   |                      |                 |
| Intel® VTune™ Amplifier XE – performance profiler | ✓ | ✓ | ✓ |
| Intel® Advisor – vectorization optimization & thread prototyping | ✓ | ✓ | ✓ |
| Intel® Inspector – memory & thread debugging | ✓ | ✓ | ✓ |

| **SCALE**       |                   |                      |                 |
| Intel® MPI Library – message passing interface library | ✓ |             |                 |
| Intel® Trace Analyzer and Collector – MPI tuning & analysis |             | ✓ |                 |
| Intel® Cluster Checker – cluster diagnostic expert system |             | ✓ |                 |

| Rogue Wave IMSL* Library – Fortran numerical analysis | Bundle or Add-on | Add-on | Add-on |

See additional license options including floating & academic at: [http://intel.ly/perf-tools](http://intel.ly/perf-tools)

---


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

Optimization Notice

Copyright © 2017, Intel Corporation. All rights reserved.
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.
Programs for Free or Discounted Tools

Intel® Parallel Studio XE

Free Software via Special Programs for those who qualify

- **Students, educators**, classroom use, open source developers, & academic researchers
- [software.intel.com/qualify-for-free-software](http://software.intel.com/qualify-for-free-software)

Intel® Performance Libraries Community Licensing

- Available to all - no royalties or restrictions based on company or project size
- [software.intel.com/nest](http://software.intel.com/nest)

Free Tools Programs

Students, Educators, classroom use, Open Source Developers, Academic Researchers (qualification required)

---

1Support for free tools is via Community Forums (most do not qualify for Intel Priority Support)
HPC Software Optimization Success Stories

**SCIENCE & RESEARCH**
Up to **35X** faster application performance
NERSC (National Energy Research Scientific Computing Center) - see case study

**LIFE SCIENCE**
Simulations ran up to **7.6X** faster with **9X** energy efficiency**
LAMMPS code - Sandia National Laboratories

**FINANCE**
Up to **2.7X** improved performance** compared to NVIDIA Tesla K80*
Monte Carlo European Options Benchmark*

**VISUALIZATION**
Up to **5.17X** performance improvement** compared to NVIDIA Titan X*
Intel Embree v2.9.0

---

**Optimization Notice**
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations & functions. Any change to any of those factors may cause the results to vary. You should consult other information & performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more information go to http://www.intel.com/performance. Optimization Notice: 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.
More Resources & Advanced Training

Intel® Parallel Studio XE

- Overview, features, support
- Dev tools training materials
- Reviews
- More Intel® Software Development Products

Intel Code Modernization Program

- Overview
- Live training
- Remote Access

software.seek.intel.com/fall-webinar-series

intel.com/content/www/us/en/events/hpcdevcon/overview.html
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
Deliver Industry-leading C/C++ & Fortran Code Performance, Unleash the Power of the latest Intel® Processors

- Develop optimized and vectorized code for various Intel® architectures, including Intel® Xeon® and Xeon Phi™ processors
- Leverage latest language and OpenMP* standards, and compatibility with leading compilers & IDEs
- SIMD Data Layout Template (SDLT) library
  - Vectorize your standard C++ array-of-structure code
- Virtual function vectorization
  - Boost performance of your object-oriented C++ code

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

---

**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

---

¹Requires Intel® VTune™ Amplifier
# Intel® Compilers: Huge Value-Add for Developers

<table>
<thead>
<tr>
<th>As a software developer, I care about</th>
<th>...my challenges are</th>
<th>Intel® Compilers offer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance</strong> – I develop applications that need to execute FAST</td>
<td>Taking advantage of the latest hardware innovations</td>
<td>Access to full power of the latest x86-compatible processors &amp; instruction sets</td>
</tr>
<tr>
<td><strong>Productivity</strong> – I need productivity &amp; ease of use offered by compilers</td>
<td>Finding support for leading languages &amp; programming models</td>
<td>Support for the latest Fortran, C/C++ &amp; OpenMP* standards; compatibility with leading compilers &amp; IDEs</td>
</tr>
<tr>
<td><strong>Scalability</strong> – I develop &amp; debug my application locally, &amp; deploy my application globally</td>
<td>Maintaining my code as core counts &amp; vector widths increase at a fast pace</td>
<td>Scalable performance without changing code as newer generation processors are introduced</td>
</tr>
</tbody>
</table>
New options and features

Function splitting

- useful to set the inline depth for too large function
- \(-[f|Q]\text{fnsplit} [=n]\), where \(0 \leq n \leq 100\)
- function splitting for functions blocks with execution probability less or equal to \(n\)
- it forces the compiler to do function splitting even if there is no dynamic profiling
- also support GCC’s \(-\text{freorder-blocks-and-partition}\) (requires dynamic profiling)
New options and features

**SVML calls dispatching during compile time**
- direct call to cpu-specific SVML entry is performed by default now
- removes SVML dynamic dispatching overhead for programs built with –x… options
- no dispatching in runtime improves performance
- –[f|Q]imf-force-dynamic-target[[=|]:]funclist] reverts the previous behavior

**FP consistency vector vs scalar code**
- for scalar math LIBM is used which does not guarantee bitwise-same result of vectorized math
  lib SVML
- –[f|Q]imf-use-svml changes scalar LIBM calls to “scalar“ SVML calls
- compiler vectorizes math functions in fp-model precise
New options and features

The Intel® Xeon® Processor Scalable Family is based on the server microarchitecture codenamed Skylake

- Compile with processor-specific option `-xCORE-AVX512`
- By default it will not optimize for more restrained ZMM register usage which works best for certain applications


```c
void foo(double *a, double *b, int size) {
    #pragma omp simd
    for(int i=0; i<size; i++) {
        b[i]=exp(a[i]);
    }
}
```

```bash
icpc -c -xCORE-AVX512 -qopenmp -qopt-report:5 foo.cpp
```

remark #15305: vectorization support: vector length 4
remark #15321: Compiler has chosen to target XMM/YMM vector. Try using `-qopt-zmm-usage=high` to override
remark #15478: estimated potential speedup: 5.260
New options and features

The Intel® Xeon® Processor Scalable Family is based on the server microarchitecture codenamed Skylake

- Compile with processor-specific option `[-/Q]xCORE-AVX512`
- By default it will not optimize for more restrained ZMM register usage which works best for certain applications


```c
void foo(double *a, double *b, int size) {
    #pragma omp simd
    for(int i=0; i<size; i++) {
        b[i]=exp(a[i]);
    }
}
```

```bash
icpc -c -xCORE-AVX512 -qopt-zmm-usage=high -qopenmp
-qopt-report:5 foo.cpp
```

remark #15305: vectorization support: vector length 8
remark #15478: estimated potential speedup: **10.110**
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

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.

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.

---

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

Up to 442X speedup versus stock pip/NumPy

Math Functions (Array size = 1M)

<table>
<thead>
<tr>
<th>Function</th>
<th>Speedup Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td>array-array</td>
<td>442X</td>
</tr>
<tr>
<td>array-scalar</td>
<td>258X</td>
</tr>
<tr>
<td>array*array</td>
<td>16X</td>
</tr>
<tr>
<td>array*scalar</td>
<td>17X</td>
</tr>
<tr>
<td>array+array</td>
<td>17X</td>
</tr>
<tr>
<td>array+scalar</td>
<td>17X</td>
</tr>
<tr>
<td>erf</td>
<td>51X</td>
</tr>
<tr>
<td>exp</td>
<td>77X</td>
</tr>
<tr>
<td>invsqrt</td>
<td></td>
</tr>
<tr>
<td>log10</td>
<td></td>
</tr>
</tbody>
</table>

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 intel_7, scipy 0.19.1 np113py36_intel_11, scikit-learn 0.18.2 np113py36_intel_3

Learn More: software.intel.com/distribution-for-python
Python* Landscape
Intel® Distribution for Python*

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

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

Most Popular Coding Languages of 2016

Optimization Notice
Copyright © 2017, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.
## What’s Inside Intel® Distribution for Python

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

### FASTER PERFORMANCE

<table>
<thead>
<tr>
<th>Performance Libraries, Parallelism, Multithreading, Language Extensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerated NumPy/SciPy/scikit-learn with Intel® MKL(^1) &amp; Intel® DAAL(^2)</td>
</tr>
<tr>
<td>Data analytics, machine learning &amp; deep learning with scikit-learn, pyDAAL, Caffe*, Theano*</td>
</tr>
<tr>
<td>Scale with Numba* &amp; Cython*</td>
</tr>
<tr>
<td>Includes optimized mpi4py, works with Dask* &amp; PySpark*</td>
</tr>
<tr>
<td>Optimized for latest Intel® architecture</td>
</tr>
</tbody>
</table>

### GREATER PRODUCTIVITY

<table>
<thead>
<tr>
<th>Prebuilt &amp; Accelerated Packages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prebuilt &amp; optimized packages for numerical computing, machine/deep learning, HPC, &amp; data analytics</td>
</tr>
<tr>
<td>Drop in replacement for existing Python - No code changes required</td>
</tr>
<tr>
<td>Jupyter* notebooks, Matplotlib included</td>
</tr>
<tr>
<td>Free download &amp; free for all uses including commercial deployment</td>
</tr>
</tbody>
</table>

### ECOSYSTEM COMPATIBILITY

<table>
<thead>
<tr>
<th>Supports Conda &amp; PIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compatible &amp; powered by Anaconda*, supports conda &amp; pip</td>
</tr>
<tr>
<td>Distribution &amp; individual optimized packages also available at conda &amp; Anaconda.org, YUM/APT, Docker image on DockerHub</td>
</tr>
<tr>
<td>Optimizations upstreamed to main Python trunk</td>
</tr>
<tr>
<td>Priority Support through Intel® Parallel Studio XE</td>
</tr>
</tbody>
</table>

### Intel® Architecture Platforms

- Operating System: Windows*, Linux*, MacOS\(^3\)*

---

\(^1\) Intel® Math Kernel Library

\(^2\) Intel® Data Analytics Acceleration Library

\(^3\) Available only in Intel® Parallel Studio XE Composer Edition
Performance Speedups for Intel® Distribution for Python* for Black Scholes* Formula (Higher is Better)

On Intel® Xeon® Processor

![Graph showing performance speedups on Intel® Xeon® Processor]

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, 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.

On Intel® Xeon Phi™ Processor Family

![Graph showing performance speedups on Intel® Xeon Phi™ Processor Family]

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, 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, 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, function, 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.
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*

---

*Available only in Intel® Parallel Studio XE Composer Edition.

*Other names and brands may be claimed as the property of others.
Automatic Performance Scaling from the Core, to Multicore, to Many Core & Beyond

Intel® Math Kernel Library (Intel® MKL)

Extract more performance from available computing resources

- Core: **vectorization**, prefetching, cache utilization
- Multi-Many core (processor/socket) level **parallelization**
- Multi-socket (node) level **parallelization**
- Clusters **scaling**
DGEMM, SGEMM Optimized by Intel® Math Kernel Library for Intel® Xeon® Platinum Processor (formerly codenamed Skylake Server)

**Optimization Notice**

Performance (GFlop/s) vs Problem Size (M=N=K) for 16 threads, 28 threads, and 56 threads in SGEMM and DGEMM.

*Configuration*: Intel® Xeon® Platinum 8180, 2x28 cores, 2.5GHz, 38.5MB L3 cache, 376GB RAM, OS Ubuntu 16.04 LTS; Intel® Math Kernel Library (Intel® MKL) 2018. 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](http://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 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.
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**

<table>
<thead>
<tr>
<th>Image Domain</th>
<th>Signal Domain</th>
<th>Data Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Processing</td>
<td>Signal Processing</td>
<td>Data Compression</td>
</tr>
<tr>
<td>Computer Vision</td>
<td>Vector Math</td>
<td>Cryptography</td>
</tr>
<tr>
<td>Color Conversion</td>
<td></td>
<td>String Processing</td>
</tr>
</tbody>
</table>

#### Operating System:
- Windows*, Linux*, MacOS®

---

2. Other names and brands may be claimed as the property of others.
Take Advantage of Intel® AVX-512 with Intel® IPP 2018

Intel® Integrated Performance Primitives (Intel® IPP)

Intel® IPP Data Compression Performance Boost vs Original LZ4 and LZO Libraries

<table>
<thead>
<tr>
<th>Original Library</th>
<th>Intel® IPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LZ4 v.1.7.5 Fast Compression</td>
<td>1.21x</td>
</tr>
<tr>
<td>LZ4 v.1.7.5 High Compression</td>
<td>1.39x</td>
</tr>
<tr>
<td>LZ0 v.2.0.8</td>
<td>1.62x</td>
</tr>
</tbody>
</table>

Intel® IPP Data Decompression Performance Boost vs Original LZ4 and LZO Libraries

<table>
<thead>
<tr>
<th>Original Library</th>
<th>Intel® IPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LZ4 v.1.7.5</td>
<td>1.47x</td>
</tr>
<tr>
<td>LZ0 v.2.0.8</td>
<td>1.72x</td>
</tr>
</tbody>
</table>

**Configuration:** Intel® Xeon® Platinum 81xx Processor, Intel® Xeon® Platinum 8168 CPU @ 2.70GHz, L3=33 MB, 2x24 cores + HT, Ubuntu®-64, 109 GB, Intel® Compiler 18, Intel® IPP 2018. 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.

Copyright © 2017, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.
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
What's Inside Intel® Threading Building Blocks

Parallel Execution Interfaces

- Flow Graph
- Generic Parallel Patterns
- Parallel STL

Low-Level Interfaces

- Tasks
- Task arenas
- Global Control

Interfaces Independent of Execution Model

Concurrent Containers

- Hash Tables
- Queues
- Vectors

Memory Allocation

- Scalable Allocator
- Cache Aligned Allocator

Primitives and Utilities

- Synchronization Primitives
- Thread Local Storage

Optimization Notice

Copyright © 2017, Intel Corporation. All rights reserved.

*Other names and brands may be claimed as the property of others.
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® 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 heterogeneous hardware and software together
  - Expose parallelism between blocks—simplify integration

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

Optimization Notice
Copyright © 2017, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.
**Optimization Notice**

Copyright © 2017, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.*
Algorithms, Data Transformation & Analysis

Intel® Data Analytics Acceleration Library

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

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

**Matrix Factorizations**
- SVD
- QR
- Cholesky

**Dimensionality Reduction**
- PCA
- Association Rule Mining (Apriori)
- Optimization Solvers (SGD, AdaGrad, lBFGS)

**Outlier Detection**
- Univariate
- Multivariate
- Math Functions (exp, log,...)

Algorithms supporting batch processing
Algorithms supporting batch, online and/or distributed processing

Optimization Notice
Copyright © 2017, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.
Intel® DAAL 2018 vs Apache Spark* MLlib Performance
Intel® Data Analytics Acceleration Library (Intel® DAAL)

![Chart showing speedup of Alternating least squares, Correlation, PCA]

**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](http://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
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**
  - 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.
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

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.

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

‘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
Flow Graph Analyzer as a technology preview and as a feature of Intel® Parallel Advisor XE

- Flow Graph Designer is a visualization tool that supports the analysis and design of parallel applications that use the Intel® Threading Building Blocks (Intel® TBB) flow graph interface.

- The flow graph interface allows developers to express the dependency, streaming and data flow graphs present in many domains such as media, gaming, finance, high performance computing and healthcare.
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 2018 edition
- Fewer false positives
- C++ 17 std::shared_mutex added
- Windows SRW Locks added

Learn More: intel.ly/inspector-xe

¹Cost 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
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

- 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
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

Intel® MPI Library = 1 library to develop, maintain & test for multiple fabrics
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
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
Pre-packed Cluster Systems Expertise as a Diagnostic Tool—Intel® Cluster Checker
For HPC Experts & those New to HPC

Top Features

Checks cluster functionality, uniformity, & performance
- Standard performance tests (e.g. DGEMM, HPL, Intel® MPI Benchmarks, IOzone, STREAM)
- Hardware & software uniformity
- Consistency for certain kernel & BIOS settings

Ability to embed, extend, & customize the checking capability (API, SDK)

Supports
- Intel® Xeon® & Intel® Xeon Phi™ processor families
- Intel® Omni-Path Fabrics, Intel® True Scale, Ethernet*, InfiniBand*
- Red Hat Enterprise Linux® 6, 7—SUSE Linux Enterprise Server® 11, 12—Ubuntu 16.04, 17.04
- Intel® HPC Orchestrator Advanced, Lustre* filesystem

Installs with Intel® Parallel Studio XE 2018 Cluster Edition for Linux*
Expert System Based Design
Modeled on Clinical Decision Support Systems

Key Concepts
- **Symptoms** are subjective indications of health
- **Signs** are objective indications of health detected by direct observation
- **Diagnoses** are the identification of the root cause of an issue
- **Remedies** are methods to resolve an issue

<table>
<thead>
<tr>
<th>Concept</th>
<th>Human</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symptom</td>
<td>Difficulty walking; ankle hurts</td>
<td>Job is running slow</td>
</tr>
</tbody>
</table>
| Signs   | - Range of ankle motion limited to 50% of normal  
           - Ankle severely inflamed compared to non-injured leg  
           - X-ray negative for fracture |  
           - DGEMM performance is 50% of peak  
           - Zombie process is using 100% of the processor |
| Diagnosis| Sprained ankle | Zombie process is stealing cycles |
| Remedy  | Ice ankle & keep it elevated, take 500 mg of ibuprofen every 4-6 hours | Kill the zombie process |
WHICH TOOL SHOULD I USE?
Optimizing Performance On Parallel Hardware
It's an iterative process...

Cluster Scalable?

Tune MPI

Ignore if you are not targeting clusters.

Effective threading?

Vectorize

Memory Bandwidth Sensitive?

Optimize Bandwidth

Thread
Performance Analysis Tools for Diagnosis
Intel® Parallel Studio

Optimization Notice
Tools for High Performance Implementation
Intel® Parallel Studio

- Cluster Scalable
  - Effective threading
    - Vectorize
      - Memory Bandwidth Sensitive
        - Optimize Bandwidth
        - Thread
      - Y
      - N
    - Y
    - N
  - Tune MPI
    - N

Intel® MPI Library
Intel® MPI Benchmarks

Intel® Compiler

Intel® Math Kernel Library
Intel® IPP – Media & Data Library
Intel® Data Analytics Library
Intel® OpenMP*

Intel® TBB – Threading Library
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. For more complete information visit www.intel.com/benchmarks.

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

Copyright © 2017, Intel Corporation. All rights reserved. Intel, Pentium, Xeon, Xeon Phi, Core, VTune, Cilk, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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
Stay Current with Support for the Latest Standards, Operating Systems & Processors

Intel® Parallel Studio XE

**Language Standards Support**
- 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

**Development Environment Integration**
- Microsoft Visual Studio* 2017
- Eclipse*
- Xcode*

**Operating Systems**
- Windows* 7 thru 10; Windows Server 2012-2016
- Debian* 8, 9; Fedora* 25; Red Hat Enterprise Linux* 6, 7; SUSE LINUX Enterprise Server* 11,12; Ubuntu* 14.04, 16.04
- macOS* 10.12

**Latest Processors**
- Support and tuning added for Intel® Xeon® Scalable Processors & Intel® AVX-512 instructions

[Complete Specifications]
Modernize Your Code Program
Intel® Parallel Studio XE

- Live webinars – Fall series starts in Sept.
- Expert talks about new features
- Attend live or watch via archives - software.intel.com/events/hpc-webinars
- Online community of BlackBelts, tools, trainings, support
- Intel® HPC Developer Conferences – where devs share proven techniques best practices
- Hands-on training for devs & partners with remote access to Intel® Xeon® & Xeon Phi™ processor-based clusters
- Developer Access Program provides early access to Intel® Xeon Phi™ processors plus 1-yr license for Intel® Parallel Studio XE Cluster Edition software.intel.com/moderncode
Advantages of Using Intel® Threading Building Blocks over other Threading Models

- Specify tasks instead of manipulating threads. Intel® Threading Building Blocks (Intel® TBB) maps your logical tasks onto threads with full support for nested parallelism.
- Intel® TBB uses proven, efficient parallel patterns.
- Intel 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 Intel 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.