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

- Speeds computations for scientific, engineering, financial and machine learning applications by providing highly optimized, threaded, and vectorized math functions
- Provides key functionality for dense and sparse linear algebra (BLAS, LAPACK, PARDISO), FFTs, vector math, summary statistics, deep learning, splines and more
- Dispatches optimized code for each processor automatically without the need to branch code
- Optimized for single core vectorization and cache utilization
- Automatic parallelism for multi-core and many-core
- Scales from core to clusters
- Available at no cost and royalty free
- Great performance with minimal effort!

**INTEL® MATH KERNEL LIBRARY OFFERS...**

- Dense &Sparse Linear Algebra
- Fast Fourier Transforms
- Vector Math
- Vector RNGs
- Fast Poisson Solver
- & More!
Automatic Dispatching to Tuned ISA-specific Code Paths

More cores → More Threads → Wider vectors

<table>
<thead>
<tr>
<th></th>
<th>Intel® Xeon® Processor 64-bit</th>
<th>Intel® Xeon® Processor 5100 series</th>
<th>Intel® Xeon® Processor 5500 series</th>
<th>Intel® Xeon® Processor 5600 series</th>
<th>Intel® Xeon® Processor E5-2600 v2 series</th>
<th>Intel® Xeon® Processor E5-2600 v3 series</th>
<th>Intel® Xeon® Scalable Processor™</th>
</tr>
</thead>
<tbody>
<tr>
<td>Up to Core(s)</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>18-22</td>
<td>28</td>
</tr>
<tr>
<td>Up to Threads</td>
<td>2</td>
<td>2</td>
<td>8</td>
<td>12</td>
<td>24</td>
<td>36-44</td>
<td>56</td>
</tr>
<tr>
<td>SIMD Width</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>256</td>
<td>256</td>
<td>512</td>
</tr>
<tr>
<td>Vector ISA</td>
<td>Intel® SSE3</td>
<td>Intel® SSE3</td>
<td>Intel® SSE4- 4.1</td>
<td>Intel® SSE 4.2</td>
<td>Intel® AVX</td>
<td>Intel® AVX2</td>
<td>Intel® AVX-512</td>
</tr>
</tbody>
</table>

1. Product specification for launched and shipped products available on ark.intel.com.
What’s New in Intel® Math Kernel Library 2019?

Just-In-Time Fast Small Matrix Multiplication

- Improved speed of S/DGEMM for Intel® AVX2 and Intel® AVX-512 with JIT capabilities

Sparse QR Solvers

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

Generate Random Numbers for Multinomial Experiments

- Highly optimized multinomial random number generator for finance, geological and biological applications
What’s Inside Intel® Math Kernel Library

<table>
<thead>
<tr>
<th>LINEAR ALGEBRA</th>
<th>FFTS</th>
<th>VECTOR RNGS</th>
<th>SUMMARY STATISTICS</th>
<th>VECTOR MATH</th>
<th>&amp; MORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAS</td>
<td>Multidimensional</td>
<td>Congruential</td>
<td>Kurtosis</td>
<td>Trigonometric</td>
<td>Splines</td>
</tr>
<tr>
<td>LAPACK</td>
<td>FFTW interfaces</td>
<td>Wichmann-Hill</td>
<td>Variation coefficient</td>
<td>Hyperbolic</td>
<td>Interpolation</td>
</tr>
<tr>
<td>ScALAPACK</td>
<td></td>
<td>Mersenne Twister</td>
<td>Order statistics</td>
<td>Exponential</td>
<td>Trust Region</td>
</tr>
<tr>
<td>Sparse BLAS</td>
<td>Cluster FFT</td>
<td>Sobol</td>
<td>Min/max</td>
<td>Log</td>
<td>Power</td>
</tr>
<tr>
<td>Iterative sparse solvers</td>
<td></td>
<td>Neiderreiter</td>
<td>Variance-covariance</td>
<td>Root</td>
<td></td>
</tr>
<tr>
<td>PARDISO*</td>
<td></td>
<td>Non-deterministic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster Sparse Solver</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.

1Available only in Intel® Parallel Studio Composer Edition.
DGEMM, SGEMM Optimized by Intel® Math Kernel Library on Intel® Xeon® Processor

---

**DGEMM on Intel® Xeon® Platinum 8180 Processor**  
2.50GHz

**SGEMM on Intel® Xeon® Platinum 8180 Processor**  
2.50 GHz

Performance results are based on testing as of July 9, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information, see [Performance Benchmark Test Disclosure](#). Testing by Intel as of July 9, 2018.

**Configuration:** Intel® Xeon® Platinum 8180 H0 205W 2x28@2.5GHz 192GB DDR4-2666

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

Copyright © 2018, Intel Corporation. All rights reserved. *Other names and brands may be claimed as the property of others.

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

Accelerate Image, Signal, Data Processing & Cryptography Computation Tasks

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

What’s New in 2019 Release

▪ Functions for ZFP floating-point data compression to help tackle large data storage challenges, great for oil/gas applications
▪ Optimization patch files for the bzip2 source 1.0.6
▪ Improved LZ4 compression & decompression performance on high entropy data
▪ New color conversion functions for convert RBG images to CIE Lab color models, & vice versa
▪ Extended optimization for Intel® AVX-512 & Intel® AVX2 instruction set
▪ Open source distribution of Intel® IPP Cryptography Library

Learn More: software.intel.com/intel-ipp
Intel® IPP Your Building Blocks for Image, Signal & Data Processing Applications

**What is Intel® IPP?**
Intel IPP provides developers with ready-to-use, processor-optimized functions to accelerate *Image & Signal processing, Data Compression & Cryptography computation tasks*.

**Why should you use Intel® IPP?**
- High Performance
- Easy to use API’s
- Faster Time To Market (TTM)
- Production Ready
- Cross-platform API

**How to get Intel® IPP?**
- Intel Parallel Studio XE
- Intel System Studio
- Free Tools Program

**Optimized for**
- Core i3
- Core i5
- Atom
- Xeon Phi
- Xeon

**Supports**
- Windows
- Linux
- Android
- MacOS

**Addresses**
- Data Center
- Internet of Things
- Embedded Systems
- Cloud Computing

**Image Processing Uses**
- Medical Imaging
- Computer Vision
- Digital Surveillance
- ADAS
- Automated Sorting
- Biometric Identification
- Visual Search

**Signal Processing Uses**
- Games (sophisticated audio content or effects)
- Echo cancellation
- Telecommunications
- Energy

**Data Compression & Cryptography Uses**
- Data centers
- Enterprise data management
- ID verification
- Smart Cards/wallets
- Electronic Signature
- Information security/cybersecurity

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Find out more at: [http://software.intel.com/intel-ipp](http://software.intel.com/intel-ipp)
What’s Inside Intel® Integrated Performance Primitives

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

Image Processing

Computer Vision

Color Conversion

Signal Processing

Vector Math

Data Compression

Cryptography

String Processing

Image Domain

Signal Domain

Data Domain

Operating Systems: Windows*, Linux*, MacOS†*

Intel® Architecture Platforms

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

†Available only in Intel® Parallel Studio Composer Edition.
Performance Improvement for Data Compression

Data Compression Performance Ratio,
Intel® Integrated Performance Primitives 2019 vs LZ4, Zlib, LZO Libraries

Performance results are based on testing as of Aug. 15, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information, see Performance Benchmark Test Disclosure.

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

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

Data Decompression Performance Ratio, Intel® Integrated Performance Primitives 2019 vs LZ4, Zlib, LZO Libraries

Performance results are based on testing as of Aug. 15, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information, see Performance Benchmark Test Disclosure.

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

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

For more complete information about compiler optimizations, see our Optimization Notice.
Get the Benefits of Advanced Threading with Threading Building Blocks

Use Threading to 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 programming
- Highly portable, composable, affordable, approachable, future-proof scalability

What's New in 2019 Release

- New capabilities in Flow Graph improve concurrency & heterogeneity through improved task analyzer & OpenCL* device selection
- New templates to optimize C++11 multidimensional arrays
- C++17 Parallel STL, OpenCL*, & Python* Conda language support
- Expanded Windows*, Linux*, Android*, MacOS* support

Learn More: software.intel.com/intel-tbb
What’s Inside 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
Heterogeneous Support
Threading Building Blocks (TBB)

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

TBB as a **composability layer** for library implementations
- One threading engine **underneath** all CPU-side work

TBB flow graph as a **coordination layer**
- Be the glue that connects heterogeneous hardware & software together
- Expose parallelism between blocks—simplify integration
Excellent Performance Scalability with Threading Building Blocks on Intel® Xeon® Processor

Performance results are based on testing as of July 31, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information visit www.intel.com/benchmarks.

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

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

- Highly tuned functions for classical machine learning & analytics performance from datacenter to edge running on Intel® processor-based devices
- Simultaneously ingests data & computes results for highest throughput performance
- Supports batch, streaming & distributed usage models to meet a range of application needs
- Includes Python®, C++, Java® APIs, & connectors to popular data sources including Spark® & Hadoop

What’s New in the 2019 Release

New Algorithms

- **Logistic Regression**, most widely-used classification algorithm
- **Extended Gradient Boosting Functionality** for inexact split calculations & user-defined callback canceling for greater flexibility
- **User-defined Data Modification Procedure** supports a wide range of feature extraction & transformation techniques

Learn More: software.intel.com/daal

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

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
Intel® Data Analytics Acceleration Library 2019 Speedup vs XGBoost*

XGBoost Open Source Project
Performance results are based on testing as of July 9, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products. For more complete information, see Performance Benchmark Test Disclosure.

Testing by Intel as of July 9, 2018. Configuration: Intel® Xeon® Platinum 8180 H0 205W, 2x28@2.50GHz, 192GB, 12x16gb DDR4-2666, Intel® Data Analytics Acceleration Library (Intel® DAAL 2019), RHEL 7.2

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

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.
## Intel® MKL BLAS (Basic Linear Algebra Subprograms)

### De-facto Standard APIs since the 1980s

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>vector vector operations, $O(N)$</td>
<td>matrix vector operations, $O(N^2)$</td>
<td>matrix matrix operations, $O(N^3)$</td>
</tr>
</tbody>
</table>

### Precisions Available

- Real – Single and Double
- Complex - Single and Double

### BLAS-like Extensions

- Direct Call, Batched, Packed and Compact

### Reference Implementation

http://netlib.org/blas/
Intel® MKL LAPACK (Linear Algebra PACKage)

<table>
<thead>
<tr>
<th>De-facto Standard APIs since the 1990s</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000s of Linear Algebra Functions</td>
</tr>
<tr>
<td>Matrix factorizations - LU, Cholesky, QR</td>
</tr>
<tr>
<td>Solving systems of linear equations</td>
</tr>
<tr>
<td>Condition number estimates</td>
</tr>
<tr>
<td>Symmetric and non-symmetric eigenvalue problems</td>
</tr>
<tr>
<td>Singular value decomposition</td>
</tr>
<tr>
<td>and many more ...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Precisions Available</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real – Single and Double,</td>
</tr>
<tr>
<td>Complex – Single and Double</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reference Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://netlib.org/lapack/">http://netlib.org/lapack/</a></td>
</tr>
<tr>
<td><strong>Intel® MKL Fast Fourier Transforms (FFTs)</strong></td>
</tr>
<tr>
<td>------------------------------------------------</td>
</tr>
<tr>
<td><strong>FFTW Interfaces support</strong></td>
</tr>
<tr>
<td><strong>Cluster FFT</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Parallelization</strong></td>
</tr>
<tr>
<td><strong>Storage Formats</strong></td>
</tr>
<tr>
<td><strong>Batch support</strong></td>
</tr>
<tr>
<td><strong>Additional Features</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
# Intel® MKL Vector Math

## Example:

\[ y(i) = e^{x(i)} \text{ for } i = 1 \text{ to } n \]

## Broad Function Support

- Basic Operations – add, sub, mult, div, sqrt
- Trigonometric – sin, cos, tan, asin, acos, atan
- Exponential – exp, pow, log, log10, log2,
- Hyperbolic – sinh, cosh, tanh
- Rounding – ceil, floor, round
  And many more

## Precisions Available

- Real – Single and Double
- Complex - Single and Double

## Accuracy Modes

- High - almost correctly rounded
- Low - last 2 bits in error
- Enhanced Performance - 1/2 the bits correct
## Intel® MKL Sparse Solvers

<table>
<thead>
<tr>
<th>Solver Type</th>
<th>Factor and solve $Ax = b$ using a parallel shared memory $LU$, $LDL$, or $LL^T$ factorization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PARDISO - Parallel</strong></td>
<td>Supports a wide variety of matrix types including real, complex, symmetric, indefinite, ...</td>
</tr>
<tr>
<td><strong>Direct Sparse Solver</strong></td>
<td>Includes out-of-core support for very large matrix sizes</td>
</tr>
<tr>
<td><strong>Parallel Direct Sparse Solver for Clusters</strong></td>
<td>Factor and solve $Ax = b$ using a parallel distributed memory $LU$, $LDL$, or $LL^T$ factorization</td>
</tr>
<tr>
<td></td>
<td>Supports a wide variety of matrix types (real, complex, symmetric, indefinite, ... )</td>
</tr>
<tr>
<td></td>
<td>Supports $A$ stored in 3-array CSR3 or BCSR3 formats</td>
</tr>
<tr>
<td><strong>DSS – Simplified PARDISO Interface</strong></td>
<td>An alternative, simplified interface to PARDISO</td>
</tr>
<tr>
<td><strong>ISS – Iterative Sparse Solvers</strong></td>
<td>Conjugate Gradient (CG) solver for symmetric positive definite systems</td>
</tr>
<tr>
<td></td>
<td>Generalized Minimal Residual (GMRes) for non-symmetric indefinite systems</td>
</tr>
<tr>
<td></td>
<td>Rely on Reverse Communication Interface (RCI) for matrix vector multiply</td>
</tr>
</tbody>
</table>