Faster Code….
Faster

Intel® Parallel Studio XE 2016
Gergana Slavova
gerhana.s.slavova@intel.com
Intel Software and Services Group
Create Faster Code…Faster

- Intel® Parallel Studio XE
  - Design, build, verify and tune
  - C++, C, Fortran and Java*

- Highlights from what’s new for “2016” edition
  - Intel® Data Analytics Acceleration Library
  - Vectorization Advisor: Custom Analysis and Advice
  - MPI Performance Snapshot: Scalable profiling
  - Support for the latest Standards, Operating Systems and Processors

http://intel.ly/perf-tools
## What’s inside each edition of Intel® Parallel Studio XE 2016

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>What it does:</strong></td>
<td><strong>Build fast code</strong> using industry leading compilers and libraries including new data analytics library</td>
<td><strong>Adds analysis tools</strong></td>
<td><strong>Adds MPI cluster tools</strong></td>
</tr>
</tbody>
</table>
| **What’s included:**| • C++ and/or Fortran compilers  
  • Performance libraries  
  • Parallel models | • Composer edition +  
  • Performance profiling  
  • Threading design/prototyping & vectorization advisor  
  • Memory & thread debugger  
  • Data analytics acceleration library | • Professional edition +  
  • MPI cluster communications library  
  • MPI error checking and tuning |
Staying current with Support for the Latest Standards, Operating Systems & Processors

C and C++ standards

- Enhanced C11 Standard support: Unicode strings and C11 anonymous unions
- New C11 keyword support: __Alignas, __Alignof, __Static_assert, __Thread_local, __Noreturn, __Generic
- Enhanced C++14 Standard support: Generic Lambdas, Generalized lambda captures, Digit Separators, [[deprecated]] attribute

Operating systems

- Debian* 7.0, 8.0; Fedora* 21, 22; Red Hat Enterprise Linux* 5, 6, 7; SuSE LINUX Enterprise Server* 11,12; Ubuntu* 12.04 LTS (64-bit only), 13.10, 14.04 LTS, 15.04
- OS X* 10.10

Fortran standards

- Fortran 2008 Submodules: Changes in the submodule do not force recompilation of uses of the module – as long as the interface does not change
- Fortran 2008 IMPURE ELEMENTAL: New IMPURE prefix allows non-PURE elemental procedures
- Fortran 2008 EXIT from BLOCK
- Features from draft Fortran 2015 to further Fortran interoperability with C (specifically to help MPI-3)

Latest processors

- Support and tuning added for the latest Intel® Processors including Skylake microarchitecture and Knights Landing microarchitecture and AVX-512
Educating with Webinar series about tools

Expert talks about the new features
Attend live, or watch after the fact.

Educating with New Parallelism Pearls Book

Real world (very exciting) applications “modernized” to utilize parallelism.

High Performance Parallelism Pearls Volume 2
- 73 expert contributors
- 23 affiliations
- 10 countries
- 24 contributed chapters

Volume Two - Released August 2015 (Now)
(Volume One – Nov 2014)

The many examples highlight the power of supporting standard parallelism models across products. Incredible science and engineering being done!

http://lotsofcores.com
More education with software.intel.com/moderncode

- Online community growing collection of tools, trainings, support
  - features Black Belts in parallelism from Intel & industry

- Developer contest
  - starts in mid-September
  - sign up now to be ready to compete when it starts!
  - win a trip to CERN (2016) and SC15 (Nov 2015)
  - software.intel.com/moderncode/challenge

- Intel® HPC Developer Conferences
  developers share proven techniques and best practices
  - hpcdevcon.intel.com

- Hands on training for developers and partners with remote access to Intel® Xeon® processor and Xeon Phi™ coprocessor-based clusters.
  - software.intel.com/icmp
Choices to Fit Needs – Intel® Tools

All Products with support – worldwide, for purchase.

- backed by Intel
- Intel® Premier Support - private direct support from Intel
- support for past versions
- software.intel.com/products

Most Products without Premier support – via special programs for those who qualify

- students, educators, classroom use, open source developers, and academic researchers
- software.intel.com/qualify-for-free-software

- Intel® Performance Libraries without Premier support -Community licensing for Intel performance libraries
  - no royalties, no restrictions based on company or project size
  - software.intel.com/nest
Performance without Compromise

Intel® C++ and Fortran Compilers on Windows*, Linux* & OS X*

Boost C++ application performance on Windows* & Linux* using Intel®C++ Compiler
(higher is better)

<table>
<thead>
<tr>
<th>Floating Point</th>
<th>Integer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Estimated SPEC®:rate_base2006</td>
<td>Linux Estimated SPEC®:rate_base2006</td>
</tr>
<tr>
<td>GCC 5.2.0</td>
<td>Intel 16.0</td>
</tr>
<tr>
<td>Intel 16.0</td>
<td>GCC 5.2.0</td>
</tr>
<tr>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>1.30</td>
<td>1.24</td>
</tr>
<tr>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>1.51</td>
<td>1.51</td>
</tr>
</tbody>
</table>

Relative geomean performance, SPEC® benchmark - higher is better

Boost Fortran application performance on Windows* & Linux* using Intel®Fortran Compiler
(higher is better)

| Windows Relative geomean performance, Polyhedron* benchmark—higher is better |
|-----------------|-----------------|-------------------|-----------------|
| PGI Fortran* 15.3 | Absoft® 15.0.1 | Intel Fortran 16.0 |
| 1.00 | 1.00 | 1.07 |
| 1.33 | 1.64 | 1.09 |
| 5.1.0 | 4.5.2 | 15.0.1 |

Optimization Notice: Intel® 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 optimization on microproces-
sors not manufactured by Intel. Microprocessor-dependent optimizations in this product
are intended for use with Intel microprocessors. Optimization Notice: Compilation of the
application itself and any other application dependencies for Intel microarchitectures may
result in different optimization benefits than compilation with other products.

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 optimization on microproces-
sors not manufactured by Intel. Microprocessor-dependent optimizations in this product
are intended for use with Intel microprocessors. Certain optimizations may be specific to Intel microarchitectures
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 #20150916.
Impressive Performance Improvement
Intel® Compiler OpenMP® 4.0 Explicit Vectorization

- Two lines added that take full advantage of both SSE or AVX
- Pragmas ignored by other compilers so code is portable

```c
typedef float complex fcomplex;
const uint32_t max_iter = 3000;
#pragma omp declare simd uniform(max_iter), simdlen(16)
uint32_t mandel(fcomplex c, uint32_t max_iter)
{
    uint32_t count = 1; fcomplex z = c;
    while ((cabsf(z) < 2.0f) && (count < max_iter)) {
        z = z * z + c; count++;
    }
    return count;
}
```

Mandelbrot calculation speedup
Normalized performance data – higher is better

<table>
<thead>
<tr>
<th></th>
<th>Serial</th>
<th>SSE 4.2</th>
<th>Core-AVX2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandelbrot</td>
<td>2.09</td>
<td>1.00</td>
<td>5.28</td>
</tr>
</tbody>
</table>

Configuration: Intel® Xeon® CPU E3-1270 @ 3.50 GHz, Haswell system (4 cores with Hyper-Threading On), running at 3.50GHz, with 32.0GB RAM, L1 Cache 256KB, L2 Cache 1.0MB, L3 Cache 8.0MB, 64-bit Windows® Server 2012 R2 Datacenter. Compiler options: -O3 -Qopenmp -Qsimd -QxSSE4.2 or AVX2 -O3 -Qopenmp -Qsimd -QxAVX2. 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. 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 preserved 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® C/C++ and Fortran Compilers

What’s New:

- More of C++14, generic lambdas, member initializers and aggregates
- More of C11, _Static_assert, _Generic, _Noreturn, and more
- OpenMP 4.0 C++ User Defined Reductions, Fortran Array Reductions
- OpenMP 4.1 asynchronous offloading, simdlen, simd ordered
- F2008 Submodules, IMPURE ELEMENTAL Functions
- F2015 TYPE(*), DIMENSION(..), RANK intrinsic, relaxed restrictions on interoperable dummy arguments
- Significant improvement in alignment analysis, vectorization robustness
- Much improved Neighboring Gather optimization
Libraries

Intel® Threading Building Blocks
Intel® Integrated Performance Primitives
Intel® Math Kernel Library
Intel DAAL
Intel® Threading Building Blocks
Intel® Threading Building Blocks (Intel® TBB)

Specify tasks instead of manipulating threads
- Intel® TBB maps your logical tasks onto threads with full support for nested parallelism

Targets threading for scalable performance
- Uses proven, efficient parallel patterns
- Uses work stealing to support the load balance of unknown execution time for tasks

Flow graph feature allows developers to easily express dependency and data flow graphs

Has high level parallel algorithms and concurrent containers and low level building blocks like scalable memory allocator, locks and atomic operations.

Open-sourced and license versions available on Linux, Windows, Mac OS X, Android

Commercial support for Intel® Atom™, Core™, Xeon® processors, and for Intel® Xeon Phi™ coprocessors
# Rich Feature Set for Parallelism

Intel® Threading Building Blocks (Intel® TBB)

<table>
<thead>
<tr>
<th>Generic Parallel Algorithms</th>
<th>Flow Graph</th>
<th>Concurrent Containers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficient scalable way to</td>
<td>A set of classes to express parallelism as a graph of compute dependencies</td>
<td>Concurrent access,</td>
</tr>
<tr>
<td>exploit the power of multi-</td>
<td>and/or data flow</td>
<td>and a scalable</td>
</tr>
<tr>
<td>core without having to start</td>
<td></td>
<td>alternative to</td>
</tr>
<tr>
<td>from scratch.</td>
<td></td>
<td>containers that are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>externally locked for</td>
</tr>
<tr>
<td></td>
<td></td>
<td>thread-safety</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task Scheduler</th>
<th>Timers and Exceptions</th>
<th>Threads</th>
<th>Thread Local Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sophisticated</td>
<td>Thread-safe timers</td>
<td>OS API</td>
<td>Efficient</td>
</tr>
<tr>
<td>work scheduling</td>
<td>and exception classes</td>
<td>wrappers</td>
<td>implementation</td>
</tr>
<tr>
<td>engine that</td>
<td></td>
<td></td>
<td>for unlimited number</td>
</tr>
<tr>
<td>empowers</td>
<td></td>
<td></td>
<td>of thread-local</td>
</tr>
<tr>
<td>parallel</td>
<td></td>
<td></td>
<td>variables</td>
</tr>
<tr>
<td>algorithms</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and the flow</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>graph</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Memory Allocation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalable memory</td>
</tr>
<tr>
<td>manager and false-</td>
</tr>
<tr>
<td>sharing free</td>
</tr>
<tr>
<td>allocators</td>
</tr>
</tbody>
</table>
Scalability and Productivity

Intel® Threading Building Blocks (Intel® TBB)

Excellent Performance Scalability with Intel® Threading Building Blocks 4.4 on Intel® Xeon® Phi™ Coprocessor

Configuration Info: SW Versions: Intel® C++ Intel® 64 Compiler, Version 16.0, Intel® Threading Building Blocks (Intel® TBB) 4.4; Hardware: Intel® Xeon Phi™ Coprocessor 7120 (16GB, 1.238 GHz, 61C/244T); MPSS Version: 3.5; Flash Version: 2.1.02.0391; Host: 2x Intel(R) Xeon(R) CPU E5-2680 0 @ 2.70GHz (16C/32T); 64GB Main Memory.; OS: Red Hat Enterprise Linux Server release 6.5 (Santiago), kernel 2.6.32-431.el6.x86_64; Benchmarks are measured only on Intel® Xeon Phi™ Coprocessor. Benchmark Source: Intel Corp. Note: sudoku 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. * Other brands and names are the property of their respective owners. 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.
Intel® Integrated Performance Primitives
Superior Performance, Portability and Compatibility with Intel® IPP

A software developer’s competitive edge

- Multi-core-ready, computationally intensive and optimized functions for large dataset problem processing and high performance computing.
- Reduces cost and time associated with software development and maintenance.
- Developers can focus their efforts only on their application code.
- Cross platform support and optimized for current and future processors

Unleash your potential through access to silicon

- Yields the best system performance for the target processor
- Takes into account memory bandwidth and caching behavior of the target environment.
- Automatic dispatching feature picks the flow optimized for that specific architecture without changing the code.
## Intel® IPP Domain Applications

### Image Processing/Color Conversion
- Healthcare (including medical imaging)
- Special effects for photo/video processing
- Object compression/decompression
- Image scaling, image combination
- Noise reduction
- Optical correction

### Computer Vision
- Digital Surveillance
- Industrial/Machine Control
- Image Recognition
- Bio-metric identification
- Remote operation of equipment and gesture interpretation
- Automated sorting of materials or objects

### Data Compression
- Internet portal data center
- Data storage centers
- Databases
- Enterprise data management

### Signal Processing
- Telecommunications
- Energy
- Recording, enhancement and playback of audio and non-audio signals
- Echo cancellation: filtering, equalization and emphasis
- Simulation of environment or acoustics
- Games involving sophisticated audio content or effects

### Cryptography
- Internet portal data center
- Information Security
- Telecommunications
- Enterprise data management
- Transaction security
- Smart card interfaces
- ID verification
- Copy protection
- Electronic signature
Intel® Math Kernel Library
Powered by the
Intel® Math Kernel Library (Intel® MKL)

- Speeds math processing in scientific, engineering and financial applications
- Functionality for dense and sparse linear algebra (BLAS, LAPACK, PARDISO), FFTs, vector math, summary statistics and more
- Provides scientific programmers and domain scientists
  - Interfaces to de-facto standard APIs from C++, Fortran, C#, Python and more
  - Support for Linux*, Windows* and OS X* operating systems
  - Extract great performance with minimal effort
- Unleash the performance of Intel® Core, Intel® Xeon and Intel® Xeon Phi™ product families
  - Optimized for single core vectorization and cache utilization
  - Coupled with automatic OpenMP*-based parallelism for multi-core, manycore and coprocessors
  - Scales to PetaFlop ($10^{15}$ floating-point operations/second) clusters and beyond
- Included in Intel® Parallel Studio XE and Intel® System Studio Suites
Optimized Mathematical Building Blocks
Intel® MKL

**Linear Algebra**
- BLAS
- LAPACK
- ScaLAPACK
- Sparse BLAS
- Sparse Solvers
- Iterative
- PARDISO* SMP & Cluster

**Fast Fourier Transforms**
- Multidimensional
- FFTW interfaces
- Cluster FFT

**Vector Math**
- Trigonometric
- Hyperbolic
- Exponential
- Log
- Power
- Root

**Vector RNGs**
- Congruential
- Wichmann-Hill
- Mersenne Twister
- Sobol
- Neiderreiter
- Non-deterministic

**Summary Statistics**
- Kurtosis
- Variation coefficient
- Order statistics
- Min/max
- Variance-covariance

**And More…**
- Splines
- Interpolation
- Trust Region
- Fast Poisson Solver

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

Extracting performance from the computing resources

- **Core**: vectorization, prefetching, cache utilization
- **Multi-Many core** (processor/socket) level parallelization
- **Multi-socket (node) level parallelization**
- **Clusters** scaling

Optimization Notice
Copyright © 2015, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.
The latest version of Intel® MKL unleashes the performance benefits of Intel architectures

DGEMM Performance Boost by using Intel®MKL vs. ATLAS*

Configurations:
- **Intel® Core™ Processor i7-4770K**
  - Intel MKL - 1 thread
  - Intel MKL - 2 threads
  - Intel MKL - 4 threads
  - ATLAS - 1 thread
  - ATLAS - 2 threads
  - ATLAS - 4 threads

- **Intel® Xeon® Processor E5-2699 v3**
  - Intel MKL - 1 thread
  - Intel MKL - 18 threads
  - Intel MKL - 36 threads
  - ATLAS - 1 thread
  - ATLAS - 18 threads
  - ATLAS - 36 threads

Matrix size (M = 10000, N = 6000, K = 64,80,96, …, 384)

Performance (GFlops) vs. Matrix size (M = N)

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. * Other brands and names are the property of their respective owners. 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.
Intel® Data Analytics Acceleration Library
Turn Big Data Into Information Faster with Intel® Data Analytics Acceleration Library

Advanced analytics algorithms supporting all data analysis stages.

Simple to incorporate object-oriented APIs for C++ and Java

Easy connections to:

- Popular analytics platforms (Hadoop, Spark)
- Data sources (SQL, non-SQL, files, in-memory)

Optimization Notice

Designed and Built by Intel to Delight Data Scientists
Low Order Moments
- computing min, max, mean, standard deviation, variance, … for a dataset.

Quantiles
- splitting observations into equal-sized groups defined by quantile orders.

Correlation matrix and variance
- The basic tool in understanding statistical dependence among variables.

Correlation distance matrix
- Measuring pairwise distance between items using correlation distance.

Cosine distance matrix
- Measuring pairwise distance using cosine distance.

Data transformation through matrix decomposition
- Supports Cholesky, QR*, and SVD* decomposition algorithms.

Outlier detection
- Identifying observations that are abnormally distant from typical distribution of other observations.

Association rules mining – Also known as “shopping basket mining”.
- Detecting co-occurrence patterns.

Linear regression*
- The simplest regression method.

Classification
- Building a model to assign items into different labeled groups.

Clustering
- Grouping data into unlabeled groups using 2 algorithms: K-Means* and “EM for GMM”.

Intel® Data Analytics Acceleration Library
List of Algorithms
Intel Software Analysis Tools

Intel® VTune™ Amplifier XE Performance Profiler
Intel® Inspector XE Memory & Thread Debugger
Intel® Advisor XE Vectorization Optimization and Thread Prototyping
Intel® Vtune™ amplifier XE
Performance Profiler
Intel® VTune™ Amplifier XE
Faster, Scalable Code Faster

Get the Data You Need
- Hotspot (Statistical call tree), Call counts (Statistical)
- Thread Profiling – Concurrency and Lock & Waits Analysis
- Cache miss, Bandwidth analysis…¹
- GPU Offload and OpenCL™ Kernel Tracing

Find Answers Fast
- View Results on the Source / Assembly
- OpenMP Scalability Analysis, Graphical Frame Analysis
- Filter Out Extraneous Data – Organize Data with Viewpoints
- Visualize Thread & Task Activity on the Timeline

Easy to Use
- No Special Compiles – C, C++, C#, Fortran, Java, ASM
- Visual Studio* Integration or Stand Alone
- Graphical Interface & Command Line
- Local & Remote Data Collection
- Analyze Windows* & Linux* data on OS X*²

¹ Events vary by processor.
² No data collection on OS X*
Tune OpenMP for Efficiency and Scalability
Get The Data You Need Fast with VTune™ Amplifier

Data Needed:

1) Is the serial time of my application significant enough to prevent scaling?
2) How much performance can be gained by tuning OpenMP?
3) Which OpenMP regions / loops / barriers will benefit most from tuning?
4) What are the inefficiencies with each region? (click the link to see details)

VTune Amplifier Summary Report:

1) Serial Time
   - Serial Time of your application is high. It directly impacts application Elapsed Time and scalability. Explore options for parallelization, algorithm or microarchitecture tuning of the serial part of the application.
   - Estimated Ideal Time: 7.115s (49.1%)
   - Potential Gain: 3.354s (23.1%)

2) Top OpenMP Regions by Potential Gain
   - This section lists OpenMP regions with the highest potential for performance improvement. The Potential Gain metric shows the elapsed time that could be saved if the region was optimized to have no load imbalance assuming no runtime overhead.
   - OpenMP Region
     - conf_grad_1omp5parallel:24@/home/avtune/work/apps/NPB/PRB3.3.3-OMPICC/caf-514-695
       - Potential Gain: 3.294s 22.7% 10.208s
     - MAIN_omp5parallel:24@/home/avtune/work/apps/NPB/PRB3.3.3-OMPICC/caf-185-231
       - Potential Gain: 0.099s 0.4% 0.260s
Tune OpenMP for Efficiency and Scalability

See the wall clock impact of inefficiencies, identify their cause

Focus On What’s Important

- What region is inefficient?
- Is the potential gain worth it?
- Why is it inefficient? Imbalance? Scheduling? Lock spinning?
- Intel® Xeon Phi systems supported
Intel® Inspector XE
Memory & Thread Debugger
Find & Debug Memory & Threading Errors
Intel® Inspector XE – Memory & Thread Debugger

Correctness Tools Increase ROI By 12%-21%¹
- Errors found earlier are less expensive to fix
- Several studies, ROI% varies, but earlier is cheaper

Diagnosing Some Errors Can Take Months
- Races & deadlocks not easily reproduced
- Memory errors can be hard to find without a tool

Debugger Integration Speeds Diagnosis
- Breakpoint set just before the problem
- Examine variables & threads with the debugger

Diagnose in hours instead of months

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® Inspector XE dramatically sped up our ability to track down difficult to isolate threading errors before our packages are released to the field.

Peter von Kaenel, Director, Software Development, Harmonic Inc.
Correctness Tools Increase ROI By 12%-21%

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

Size and complexity of applications is growing

Reworking defects is 40%-50% of total project effort

Correctness tools find defects during development prior to shipment

Reduce time, effort, and cost to repair

Find errors earlier when they are less expensive to fix
Race Conditions Are Difficult to Diagnose
They only occur occasionally and are difficult to reproduce

<table>
<thead>
<tr>
<th>Thread 1</th>
<th>Thread 2</th>
<th>Shared Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Read count</td>
<td>➡️ 0</td>
<td></td>
</tr>
<tr>
<td>Increment</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Write count</td>
<td>➡️ 1</td>
<td></td>
</tr>
<tr>
<td>Read count</td>
<td>➡️ 1</td>
<td></td>
</tr>
<tr>
<td>Increment</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Write count</td>
<td>➡️ 2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Thread 1</th>
<th>Thread 2</th>
<th>Shared Counter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Read count</td>
<td>➡️ 0</td>
<td></td>
</tr>
<tr>
<td>Increment</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Write count</td>
<td>➡️ 1</td>
<td></td>
</tr>
<tr>
<td>Read count</td>
<td>➡️ 0</td>
<td></td>
</tr>
<tr>
<td>Increment</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Write count</td>
<td>➡️ 1</td>
<td></td>
</tr>
</tbody>
</table>
Incrementally Diagnose Memory Growth
Intel® Inspector XE

As your app is running…

Memory usage graph plots memory growth

Select a cause of memory growth

See the code snippet & call stack

Speed diagnosis of difficult to find heap errors
Intel® Advisor XE
Vectorization optimization and thread Prototyping for Software Architects
Get Faster Code Faster! Intel® Advisor XE Thread Prototyping

Have you:

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

Data Driven Threading Design:

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

"Intel® Advisor XE has allowed us to quickly prototype ideas for parallelism, saving developer time and effort"

Simon Hammond
Senior Technical Staff
Sandia National Laboratories

Add Parallelism with Less Effort, Less Risk and More Impact

Optimization Notice
Copyright © 2015, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.
What’s new: Intel® Advisor XE
Vectorization Optimization

Have you:

- Recomplied for AVX2 with little gain
- Wondered where to vectorize?
- Recoded intrinsics for new arch.?  
- Struggled with compiler reports?

Data Driven Vectorization:

- What vectorization will pay off most?
- What’s blocking vectorization? Why?
- Are my loops vector friendly?
- Will reorganizing data increase performance?
- Is it safe to just use pragma simd?
Design Then Implement
Intel® Advisor XE Thread Prototyping

Design Parallelism

- No disruption to regular development
- All test cases continue to work
- Tune and debug the design before you implement it

Implement Parallelism

Less Effort, Less Risk, More Impact
Get Faster Code Faster! Intel® Advisor XE
Vectorization Optimization and Thread Prototyping
Customer Reviews

“Intel® VTune™ Amplifier XE analyzes complex code and helps us identify bottlenecks rapidly. By using it and other Intel® Software Development Tools, we were able to improve PIPESIM performance up to 10 times compared with the previous software version.”

Rodney Lessard
Senior Scientist
Schlumberger

“Intel® Advisor XE has been extremely helpful in identifying the best pieces of code for parallelization. We can save several days of manual work by targeting the right loops. At the same time, we can use Advisor to find potential thread safety issues to help avoid problems later on.”

Carlos Boneti
HPC software engineer,
Schlumberger

Intel® Inspector XE has dramatically sped up our ability to find/fix memory problems and track down difficult to isolate threading errors before our packages are released to the field.

Peter von Kaenel, Director,
Software Development,
Harmonic Inc.
Cluster Tools

Intel® MPI Library

Intel® Trace Analyzer and Collector
Intel® MPI Library Overview

Optimized MPI application performance
- Application-specific tuning
- Automatic tuning

Lower latency and multi-vendor interoperability
- Industry leading latency
- Performance optimized support for the latest OFED capabilities through DAPL 2.0

Faster MPI communication
- Optimized collectives

Sustainable scalability up to 340K cores
- Native InfiniBand* interface support allows for lower latencies, higher bandwidth, and reduced memory requirements

More robust MPI applications
- Seamless interoperability with Intel® Trace Analyzer and Collector
Intel® Trace Analyzer and Collector Overview

Intel® Trace Analyzer and Collector helps the developer:

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

Features

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

Automatically detect performance issues and their impact on runtime
Scalable Profiling for MPI and Hybrid Clusters with MPI Performance Snapshot

- **Lightweight** – Low overhead profiling up to 32K Ranks
- **Scalability** – Performance variation at scale can be detected sooner
- **Identifying Key Metrics** – Shows PAPI counters and MPI/OpenMP* imbalances

**MPI Performance Snapshot**

- **Overview**
  - **MPI Time**: 0.12 sec
  - **MPI Imbalance**: 0.04 sec
  - **Computation Time**: 1.30 sec

- **Performance by Metric**
  - **WallClock time**: 1.44 sec
  - **MPI Time**: 0.12 sec
  - **MPI Imbalance**: 0.04 sec
  - **Computation Time**: 1.30 sec

- **Memory Usage**
  - **Peak memory consumption (rank 1)**: 13.18 MB
  - **Mean memory consumption**: 12.69 MB

*Other names and brands may be claimed as the property of others.*
Intel® Parallel Studio XE 2017 Beta

Easily Build IA Optimized Data Analytics Application

Intel® Data Analytics Acceleration Library (DAAL) Beta adds support for neural networks to accelerate deep learning applications. New Python* API allows data scientists to accelerate Python* code for their data analytics applications.

Boost Speed of Machine Learning Applications

Intel® Math Kernel Library 2017 Beta introduces machine learning support by including DNN (Deep Neural Networks) primitives which helps in accelerating the machine learning workloads.

Easy, Accessible High Performance Python

Intel® Distribution for Python* Beta powered by Intel® Math Kernel Library (Intel® MKL) is a performance oriented Distribution that delivers robust solutions for faster performance from your Python applications, in an easy to access integrated package.
Intel® Parallel Studio XE 2017 Beta

Profile Mixed Python, Go + Native Code
Intel® VTune™ Amplifier XE Beta, unlike pure Python profilers, can now profile mixed Python + native C, C++ or Fortran code. Find the real performance bottleneck in your Python or Go application even if it does call native code.

Discover Untapped Performance Faster
Intel® VTune™ Amplifier XE Beta introduces two new performance snapshots – joining to MPI Performance Snapshot feature of Intel® Trace Analyzer and Collector - for a quick and easy way to prioritize performance projects based on data instead of guesswork. Application Performance Snapshot shows the potential benefit of code modernization. Storage Performance Snapshot (coming soon in the beta update) shows if faster storage may improve performance.

Boost Code Performance with Intel Compilers
Intel® C++ and Fortran Compiler 17.0 Beta includes several enhancements to improve your performance and vectorization capabilities – enjoy a significant boost in Coarray Fortran performance and vectorize your C++ code using the new SIMD Data Layout Template.
Legal Disclaimer & 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.

Copyright © 2015, 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.

<table>
<thead>
<tr>
<th>Optimization Notice</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

Notice revision #20110804
## Pricing and Configurations

**Intel® Parallel Studio XE 2016**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>from $699</td>
<td>from $1,699</td>
<td>from $2949</td>
</tr>
<tr>
<td>Intel® C++ Compiler</td>
<td>Intel® C++ Compiler</td>
<td>Intel® C++ Compiler</td>
</tr>
<tr>
<td>Intel® Fortran Compiler</td>
<td>Intel® Fortran Compiler</td>
<td>Intel® Fortran Compiler</td>
</tr>
<tr>
<td>Intel® Data Analytics Acceleration Library</td>
<td>Intel® Data Analytics Acceleration Library</td>
<td>Intel® Data Analytics Acceleration Library</td>
</tr>
<tr>
<td>Intel® Threading Building Blocks</td>
<td>Intel® Threading Building Blocks</td>
<td>Intel® Threading Building Blocks</td>
</tr>
<tr>
<td>Intel® Integrated Performance Primitives</td>
<td>Intel® Integrated Performance Primitives</td>
<td>Intel® Integrated Performance Primitives</td>
</tr>
<tr>
<td>Intel® Math Kernel Library</td>
<td>Intel® Math Kernel Library</td>
<td>Intel® Math Kernel Library</td>
</tr>
<tr>
<td>Intel® Cilk™ Plus &amp; Intel® OpenMP*</td>
<td>Intel® Cilk™ Plus &amp; Intel® OpenMP*</td>
<td>Intel® Cilk™ Plus &amp; Intel® OpenMP*</td>
</tr>
<tr>
<td></td>
<td>Intel® Advisor XE</td>
<td>Intel® Advisor XE</td>
</tr>
<tr>
<td></td>
<td>Intel® Inspector XE</td>
<td>Intel® Inspector XE</td>
</tr>
<tr>
<td></td>
<td>Intel® VTune™ Amplifier XE</td>
<td>Intel® VTune™ Amplifier XE</td>
</tr>
<tr>
<td>Bundle or Add-on:</td>
<td>Add-on:</td>
<td>Add-on:</td>
</tr>
<tr>
<td>Rogue Wave IMSL* Library</td>
<td>Rogue Wave IMSL* Library</td>
<td>Rogue Wave IMSL* Library</td>
</tr>
</tbody>
</table>

Additional configurations including, floating and academic, are available at: [http://intel.ly/perf-tools](http://intel.ly/perf-tools)