Intel® Xeon Phi™ programming

September 22nd-23rd 2015

University of Copenhagen, Denmark

Legal Disclaimer & 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 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.

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® software tools overview

Intel® Xeon® and Intel® Xeon Phi™

SOFTWARE AND SERVICES

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

Intel® Parallel Studio XE 2016 editions

	Composer Edition	Professional Edition	Cluster Edition
What it does:	***Build fast code*** using industry leading compilers and libraries including new data analytics library	***Adds analysis tools***	***Adds MPI cluster tools***
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

Intel® Parallel Studio XE 2016 compilers

Intel® Xeon® and Intel® Xeon Phi™

SOFTWARE AND SERVICES

Intel® C/C++ and Fortran Compilers

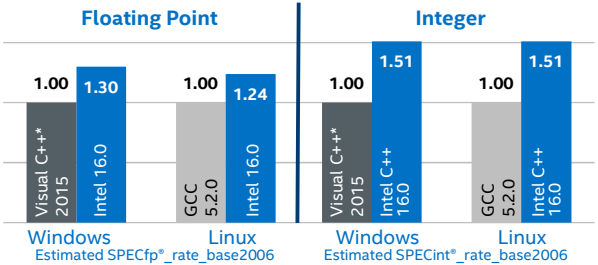
What's New:

- 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

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)

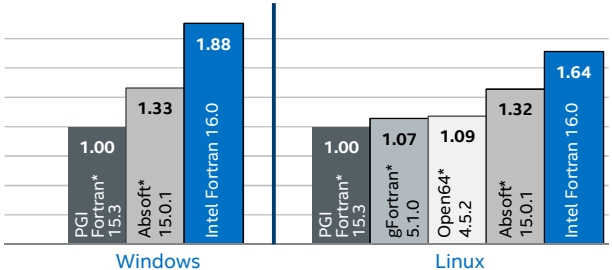


Configuration: Windows hardware: HP DL320e Gen8 v2 (single-socket server) with Intel(R) Xeon(R) CPU E3-1280 v3 @ 3.60GHz, 32 GB RAM, HyperThreading is off; Linux hardware: HP BL460c Gen9 with Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50GHz, 256 GB RAM, HyperThreading is on. Software: Intel C++ compiler 16.0, Microsoft (R) C/C++ Optimizing Compiler Version 19.00.23026 for x86/x64, GCC 5.2.0, Linux OS: Red Hat Enterprise Linux Server release 7.1 (Maipo), kernel 3.10.0-229.el7.x86_64. Windows OS: Windows 8.1. SPEC* Benchmark (www.spec.org).

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.

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



Configuration: Hardware: Intel(R) Core(TM) i7-4770K CPU @ 3.50GHz, HyperThreading is off, 16 GB RAM. Software: Intel Fortran compiler 16.0, Absoft*15.0.1., PGI Fortran* 15.3, Open64* 4.5.2, gFortran* 5.1.0. Linux OS: Red Hat Enterprise Linux Server release 7.0 (Maipo), kernel 3.10.0-123.el7.x86_64. Windows OS: Windows 7, Service pack 1. Polyhedron Fortran Benchmark (www.fortran.uk). Windows compiler switches: Absoft: -m64 -O5 -speed_math=10 -fast_math -march=core -xINTEGER -stack:0x80000000, Intel* Fortran compiler: /fast /Qparallel /link /stack:64000000. PGI Fortran: -fastsse -Munroll=n:4 -Mipa=fast,inline -Mconcur=numa. Linux compiler switches: Absoft -m64 -mavx -O5 -speed_math=10 -march=core -xINTEGER. Gfortran: -Ofast -mfpmath=sse -flto -march=native -funroll-loops -ftree-parallelize-loops=4. Intel Fortran compiler: -fast -parallel. PGI Fortran: -fast -Mipa=fast,inline -Msmartalloc -Mfprelaxed -Mstack_arrays -Mconcur=bind. Open64: -march=bdver1 -mavx -mno-fma4 -Ofast -mso –apo.

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

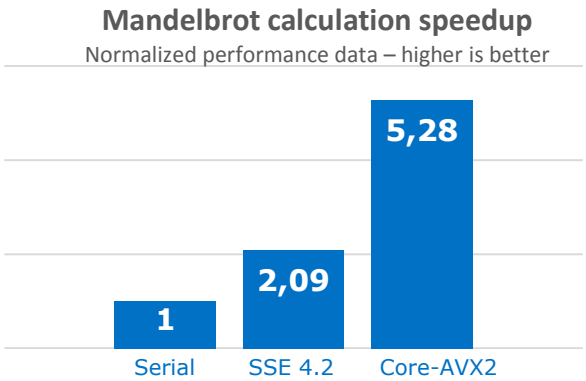
Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, 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.

Impressive performance improvement

Intel® Compiler OpenMP* 4.0 Explicit Vectorization

- Two lines added that take full advantage of both SSE or AVX
- Code portable between compilers

```
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;
}
uint32_t count[ImageWidth][ImageHeight];
 …… …. …….
      for (int32_t y = 0; y < ImageHeight; ++y) {
        float c_im = max_imag - y * imag_factor;
#pragma omp simd safelen(16)
        for (int32_t x = 0; x < ImageWidth; ++x) {
          fcomplex in_vals_tmp = (min_real + x * real_factor) + (c_im * 1.0iF);
          count[y][x] = mandel(in_vals_tmp, max_iter);
        }
      }
```



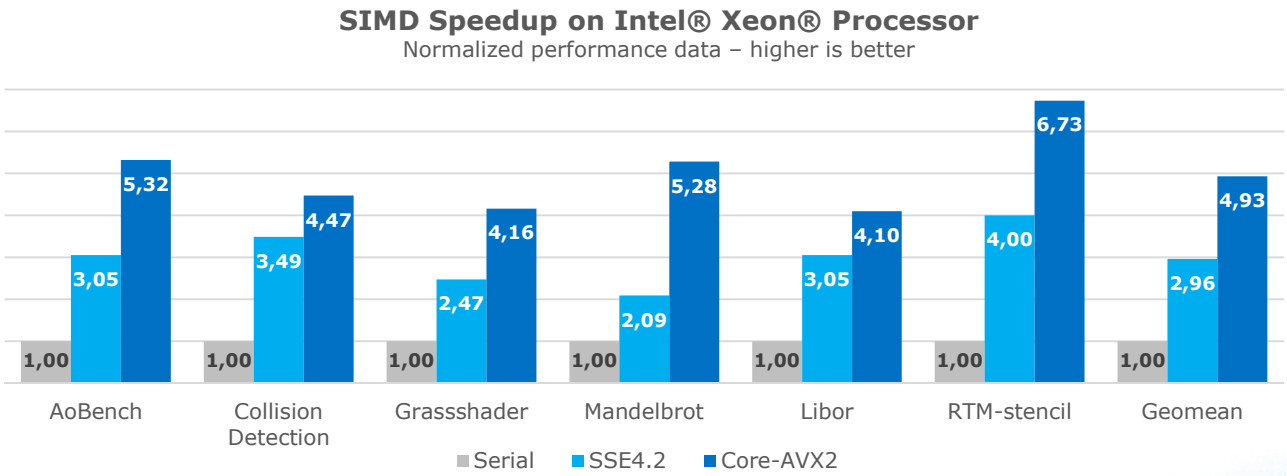
Configuration: Intel® Xeon® CPU E3-1270 @ 3.50 GHz Haswell system (4 cores with Hyper-Threading On), running at 3.50GHz, with 32.0GB RAM, L1 Cache 256KB, L2 Cache 1.0MB, L3 Cache 8.0MB, 64-bit Windows* Server 2012 R2 Datacenter. Compiler options:, SSE4.2: –O3 –Qopenmp -simd –QxSSE4.2 or AVX2: -O3 -Qopenmp -simd -QxCORE-AVX2. For more information go to http://www.intel.com/performance

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.

Impressive performance improvement

Intel C++ Explicit Vectorization using OpenMP* 4.0 SIMD or Intel® Cilk™ Plus



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:, SSE4.2: –O3 –Qopenmp -simd –QxSSE4.2 or AVX2: -O3 –Qopenmp –simd -QxCORE-AVX2. For more information go to http://www.intel.com/performance

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 instructions sets covered by this notice. Notice rev. 20110804

Intel® Parallel Studio XE 2016 libraries

Intel® Threading Building Blocks
Intel® Integrated Performance Primitives
Intel® Math Kernel Library
Intel® Data Analytics Acceleration Library

Intel® Xeon® and Intel® Xeon Phi™

SOFTWARE AND SERVICES

Intel® Threading Building Blocks

SOFTWARE AND SERVICES

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 OSX, 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)

- Parallel algorithms and data structures
- Threads and synchronization
- Memory allocation and task scheduling

Generic Parallel Algorithms

Efficient scalable way to exploit the power of multi-core without having to start from scratch.

Flow Graph

A set of classes to express parallelism as a graph of compute dependencies and/or data flow

Concurrent Containers

Concurrent access, and a scalable alternative to containers that are externally locked for thread-safety

Synchronization Primitives

Atomic operations, a variety of mutexes with different properties, condition variables

Task Scheduler

Sophisticated work scheduling engine that empowers parallel algorithms and the flow graph

Timers and Exceptions

Thread-safe timers and exception classes

Threads

OS API wrappers

Thread Local Storage

Efficient implementation for unlimited number of thread-local variables

Memory Allocation

Scalable memory manager and false-sharing free allocators

Intel® Threading Building Blocks

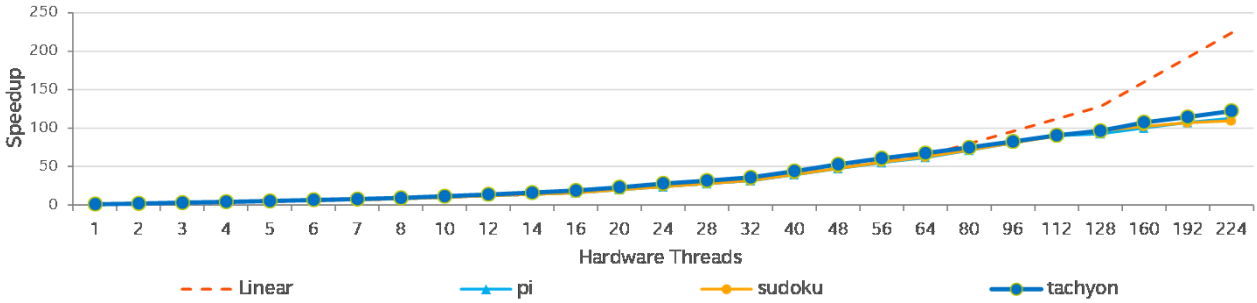
What's new:

- Fully supported **`tbb::task_arena`**
  - Task arenas provide improved control over workload isolation and the degree of concurrency
- Dynamic replacement of standard memory allocation routines for OS X*
  - Utilize the powerful TBB scalable allocator easily on OS X
- Binary files for 64-bit Android* applications were added as part of the Linux* package
- Improvements to the Flow Graph features
  - Check out Flow Graph Designer!
- Several improvements to examples and documentation

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

SOFTWARE AND SERVICES

Intel® Integrated Performance Primitives (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	Computer Vision	Data Compression	Signal Processing	Cryptography
• Healthcare (including medical imaging) • Special effects for photo/video processing • Object compression/ decompression • Image scaling, image combination • Noise reduction • Optical correction	• Digital Surveillance • Industrial/Machine Control • Image Recognition • Bio-metric identification • Remote operation of equipment and gesture interpretation • Automated sorting of materials or objects	• Internet portal data center • Data storage centers • Databases • Enterprise data management	• 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	• Internet portal data center • Information Security • Telecommunications • Enterprise data management • Transaction security • Smart card interfaces • ID verification • Copy protection • Electronic signature

Intel® Integrated Performance Primitives

What's new:

- Additional optimization for Intel® Quark™, Intel® Atom™, and the processors with Intel® AVX2 instructions support
  - Intel® Quark™: data compression, cryptography optimization
  - Intel® Atom™: computation vision, image processing optimization
  - Intel® AVX2: computer vision, image processing optimization
- New APIs to support external threading
- Improved CPU dispatcher
  - Auto-initialization. No need for the CPU initialization call in static libraries.
  - Code dispatching based on CPU features
- Optimized cryptography functions to support SM2/SM3/SM4 algorithm
- Custom dynamic library building tool
- New APIs to support external memory allocation

Intel® Math Kernel Library

SOFTWARE AND SERVICES

Intel® Math Kernel Library (Intel® MKL)

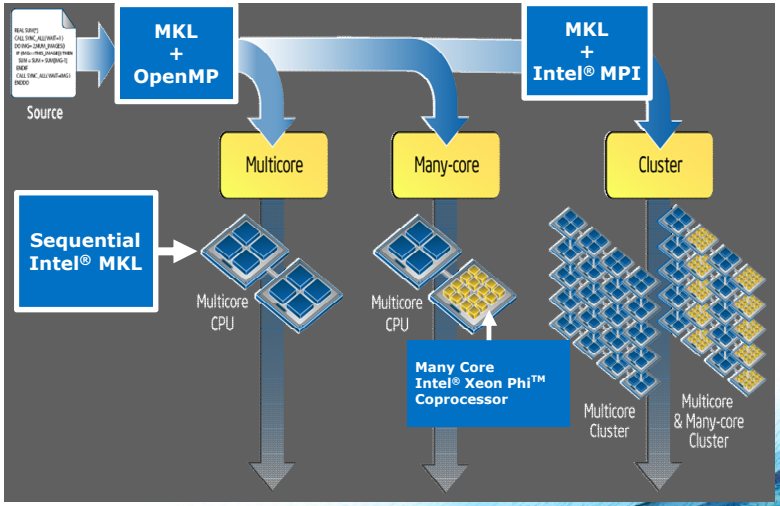
- Speeds math processing in scientific, engineering and financial applications
- Functionality for dense and sparse linear algebra (BLAS, LAPACK, PARDISO), FFTs, vector math, summary statistics and more
- Provides scientific programmers and domain scientists
  - Interfaces to de-facto standard APIs from 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 (1015 floating-point operations/second) clusters and beyond
- Included in Intel® Parallel Studio XE and Intel® System Studio Suites

Intel® Math Kernel Library (Intel® MKL)

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

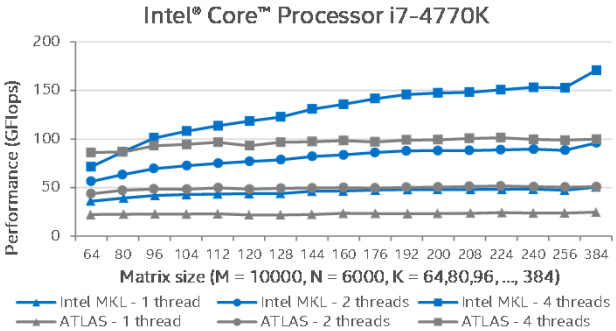
Automatic performance scaling from the core, multicore, many-core and beyond

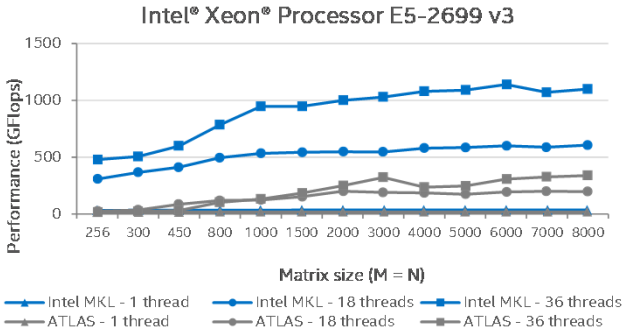
- **Extracting performance from the computing resources**
  - Core: **vectorization**, prefetching, cache utilization
  - Multi-Many core (processor/socket) level **parallelization**
  - Multi-socket (node) level **parallelization**
  - Clusters **scaling**



The latest version of Intel® MKL unleashes the performance benefits of Intel architectures

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





Configuration Info - Versions: Intel® Math Kernel Library (Intel® MKL) 11.3, ATLAS* 3.10.2; Hardware: Intel® Xeon® Processor E5-2699v3, 2 Eighteen-core CPUs (45MB LLC, 2.3GHz), 64GB of RAM; Intel® Core™ Processor i7-4770K, Quad-core CPU (8MB LLC, 3.5GHz), 8GB of RAM; Operating System: RHEL 6.4 GA x86_64;

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® MKL 11.3

What's New:

- Certified component of the VXF 2016 Reference platform
- Batch GEMM functions
  - Improve the performance of multiple, simultaneous matrix multiply operations
  - Provides grouping (the same sizes and leading dimensions) and batching across groups
- Sparse BLAS inspector-executor API
  - Matrix structure analysis brings performance benefit for relevant applications (i.e. iterative solvers)
  - Parallel triangular solver
  - Both 0-based and 1-based indexing, row-major and column-major ordering
  - Extended BSR support
- GEMMT functions calculate C = A * S * AT, where S is symmetric and/or diagonal
- Counter-based pseudorandom number generators
  - ARS-5 based on the Intel AES-NI instruction set
  - Philox4x32-10
- Intel MKL PARDISO scalability
  - Improved Intel MKL PARDISO and Cluster Sparse Solver scalability on Intel Xeon Phi coprocessors
- Cluster components extension
  - MPI wrappers provide compatibility with most MPI implementations including custom ones
  - Cluster components support on OS X

Intel® Data Analytics Acceleration Library

SOFTWARE AND SERVICES

Intel® Data Analytics Acceleration Library

- Advanced analytics algorithms supporting all data analysis stages.

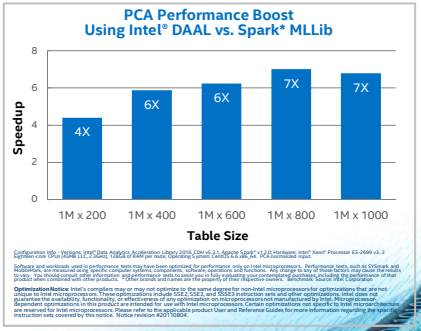
DATA SOURCES

Business
Scientific
Engineering
Web/Social

Pre-processing	Transformation	Analysis	Modeling	Validation	Decision Making
• Decompression • Filtering • Normalization	• Aggregation • Dimension Reduction	• Summary Statistics • Clustering.	• Machine Learning • Parameter Estimation • Simulation	• Hypothesis testing • Model errors	• Forecasting • Decision Trees • Etc.

- 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)

Designed and Built by Intel to Delight Data Scientists



Configuration Info - Versions: Intel® Data Analytics Acceleration Library 2016, CDH v5.2.1, Apache Spark* v1.2.0; Hardware: Intel® Xeon® Processor E5-2699 v3, 2 Sockets/18 core CPUs (36HT), 2.3GHz, 128GB of DIMM per node; Operating System: CentOS 6.6 x86_64. PCA normalized input.

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

List of Algorithms

- 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 uisng 2 algorithms: K-Means and "EM for GMM"

Intel® Parallel Studio XE 2016 tools

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

Intel® Xeon® and Intel® Xeon Phi™

SOFTWARE AND SERVICES

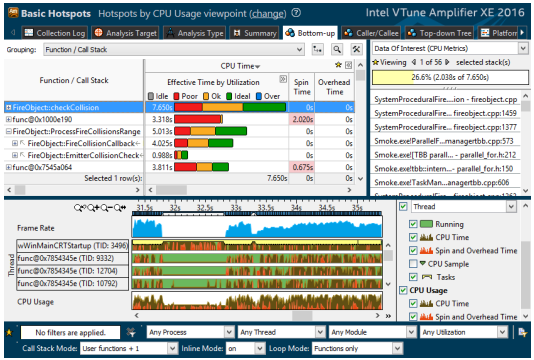
Intel® VTune™ Amplifier XE

SOFTWARE AND SERVICES

Intel® VTune™ Amplifier Performance Profiler

Get Faster Code Faster with accurate data & meaningful analysis

- Accurate CPU, GPU and threading data
- OpenMP region efficiency analysis
- Powerful data analysis & filtering
- Data displayed on the source code
- Easy set-up, no special compiles



http://intel.ly/vtune-amplifier-xe

Collecting data
Intel® VTune™ Amplifier

Software Collector	Hardware Collector	
Uses OS interrupts	Uses the on chip Performance Monitoring Unit (PMU)	
Collects from a single process tree	Collect system wide or from a single process tree.	
~10ms default resolution	~1ms default resolution (finer granularity - finds small functions)	
Either an Intel® or a compatible processor	Requires a genuine Intel® processor for collection	
Call stacks show calling sequence	Optionally collect call stacks	
Works in virtual environments	Works in a VM only when supported by the VM (e.g., vSphere*, KVM)	
No driver required	Requires a driver	- Easy to install on Windows - Linux requires root (or use default perf driver without stacks)

No special recompiles - C, C++, C#, Fortran, Java, Assembly

A rich set of performance data
Intel® VTune™ Amplifier

Software Collector	Hardware Collector
Basic Hotspots Which functions use the most time?	**Advanced Hotspots** Which functions use the most time? Where to inline? – Statistical call counts
Concurrency Tune parallelism. Colors show number of cores used.	**General Exploration** Where is the biggest opportunity? Cache misses? Branch mispredictions?
Locks and Waits Tune the #1 cause of slow threaded performance: – waiting with idle cores.	**Advanced Analysis** Dig deep to tune access contention, etc.
Any IA86 processor, any VM, no driver	Higher res., lower overhead, system wide

No special recompiles - C, C++, C#, Fortran, Java, Assembly

Find answers fast

Intel® VTune™ Amplifier

Adjust data grouping

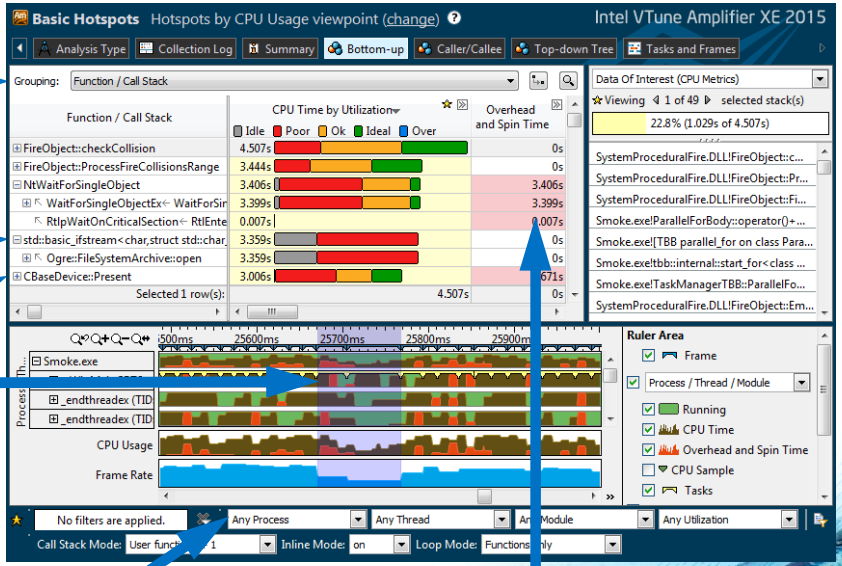
Function - Call Stack
Module - Function - Call Stack
Source File - Function - Call Stack
Thread - Function - Call Stack
... (Partial list shown)

Double click function to view source

Click [+] for call stack

Filter by timeline selection (or by grid selection)

Zoom In And Filter On Selection
Filter In by Selection
Remove All Filters



Filter by process & other controls

Tuning opportunities shown in pink. Hover for tips

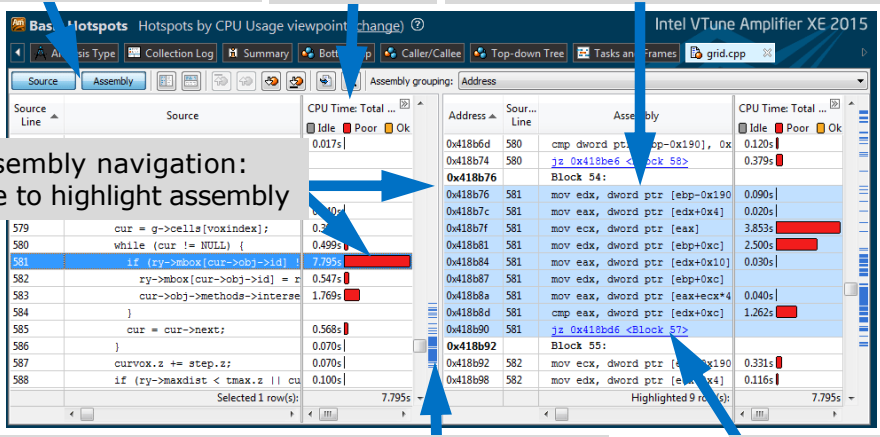
SOFTWARE AND SERVICES

Profile data on source / assembly
Intel® VTune™ Amplifier

View source / assembly or both

CPU Time

Right click for instruction reference manual

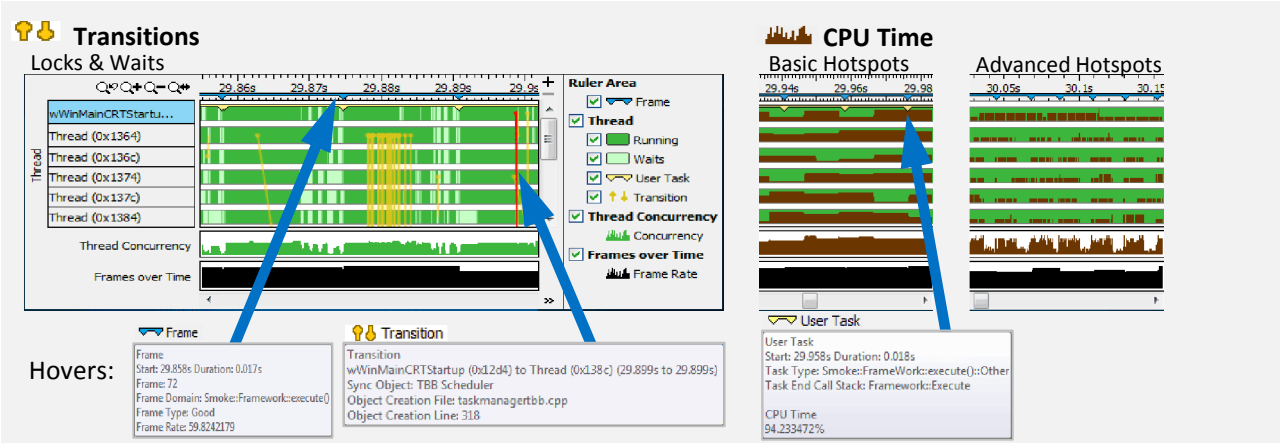


Quick assembly navigation:
Select source to highlight assembly

Scroll Bar "Heat Map" is an overview of hot spots

Click jump to scroll assembly

Thread behavior timeline
Intel® VTune™ Amplifier



- Optional: Use API to mark frames and user tasks 

- Optional: Add a mark during collection 

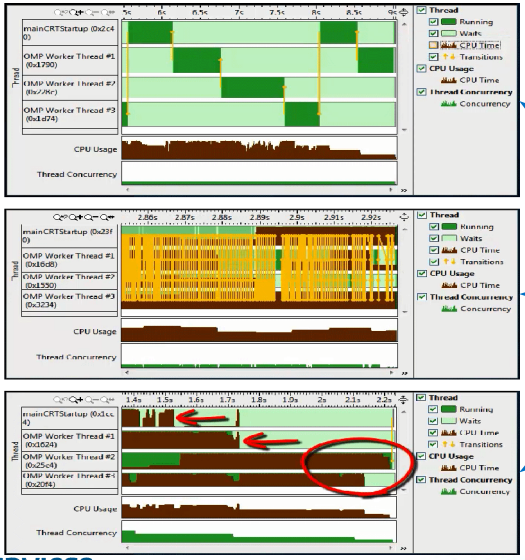
Parallel performance issues

Intel® VTune™ Amplifier

Coarse grain locks

High lock contention

Load imbalance



Low concurrency

OpenMP analysis
Intel® VTune™ Amplifier

Fast Answers: Is my OpenMP scalable? How much faster could it be?

OpenMP Analysis. Collection Time: 14.490

1) ▶ Serial Time (outside any parallel region): 4.020s (27.7%)

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.

Parallel Region Time: 10.469s (72.3%)

Estimated Ideal Time: 7.115s (49.1%)

2) ▶ Potential Gain: 3.354s (23.1%)

The time wasted on load imbalance or parallel work arrangement is significant and negatively impacts the application performance and scalability. Explore OpenMP regions with the highest metric values. Make sure the workload of the regions is enough and the loop schedule is..

3) ▶ **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.

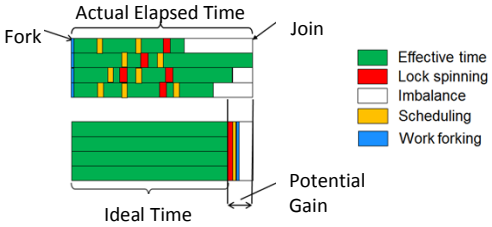
OpenMP Region	Potential Gain	(%)	Elapsed Time
4) ▶ conj_grad_ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:514:695	3.294s	22.7%	10.208s
MAIN__ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:185:231	0.059s	0.4%	0.260s

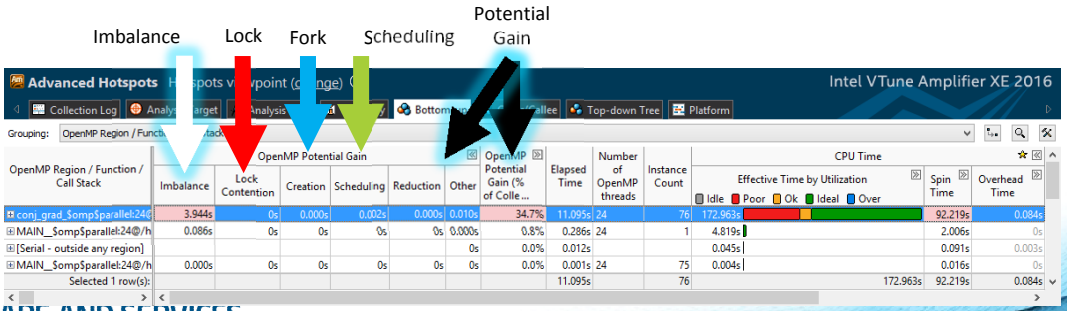
The summary view shown above gives fast answers to four important OpenMP tuning questions:

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)

OpenMP analysis
Intel® VTune™ Amplifier

- 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





Results comparison
Intel® VTune™ Amplifier

- Quickly identify cause of regressions.
  - Run a command line analysis daily
  - Identify the function responsible so you know who to alert
- Compare 2 optimizations – What improved?
- Compare 2 systems – What didn't speed up as much?

Grouping: Function / Call Stack

Function / Call Stack	CPU Time:Difference	Module	CPU Time:r007hs	CPU Time:r006hs
FireObject::checkCollision	4.850s	SystemProceduralFire.DLL	6.281s	1.431s
FireObject::ProcessFireCollisionsRange	4.644s	SystemProceduralFire.DLL	5.643s	0.999s
dllStopPlugin	3.765s	RenderSystem_Direct3D9.DLL	9.184s	5.419s

Linux* improvements
Intel® VTune™ Amplifier

Previously added in 2015:

- Auto-rebuild Intel EBS driver
  - Does advanced analysis stop working when an OS update is installed?
  - Do you have to ask IT to rebuild the driver?
  - No longer! Just setup the driver to auto-rebuild when the OS is updated.
- Auto-disable NMI watchdog
  - Tired of turning off NMI watchdog to run advanced EBS profiling?
  - Now you don't have to. We turn it off, then put it back the way it was.

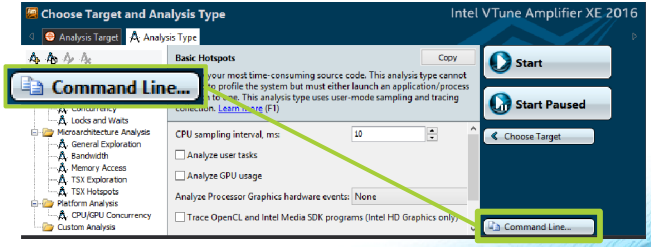
Added in 2016

- Perf can collect stacks
- Use pre-installed perf driver
  - Intel EBS driver provides additional features not available in perf:
    - Uncore events
    - Multiple precise events
    - New events for the latest processors, even on an older OS

Easier access to the on-chip PMU for advanced performance profiling

Command line interface
Intel® VTune™ Amplifier

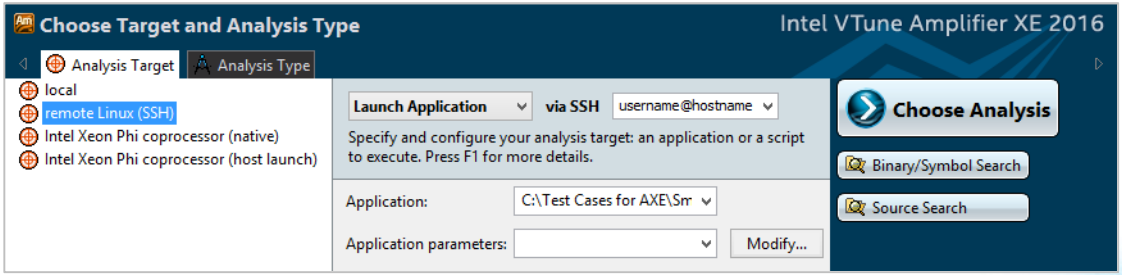
- Command line tool **`amplxe-cl`**
  - Windows:
    `C:\Program Files (x86)\Intel\VTune Amplifier XE \bin[32|64]\amplxe-cl.exe`
  - Linux:
    `/opt/intel/vtune_amplifier_xe/bin[32|64]/amplxe-cl`
- Help: **`amplxe-cl -help`**
- Use UI to setup
  1) Configure analysis in UI
  2) Press "Command Line…" button
  3) Copy & paste command



Great for regression analysis – send results file to developer
Command line results can also be opened in the UI

Interactive remote data collection
Intel® VTune™ Amplifier

- Interactive analysis
  1) Configure SSH to a remote Linux* target
  2) Choose and run analysis with the GUI
- Command line analysis
  1) Run command line remotely on Windows* or Linux* target
  2) Copy results back to host and open in GUI



Conveniently use your local UI to analyze remote systems

Intel® Inspector XE

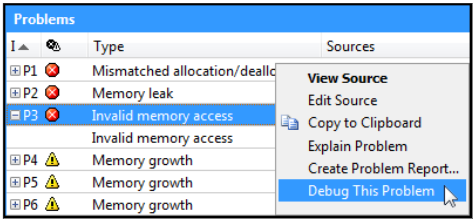
SOFTWARE AND SERVICES

Intel® Inspector XE – Memory & Thread Debugger

Find & debug memory & threading errors

- Correctness tools increase ROI By 12%-21%
  - 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

Debugger Breakpoints



http://intel.ly/inspector-xe

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

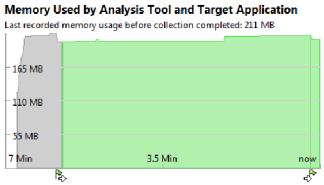
Memory growth diagnostics
Intel® Inspector XE

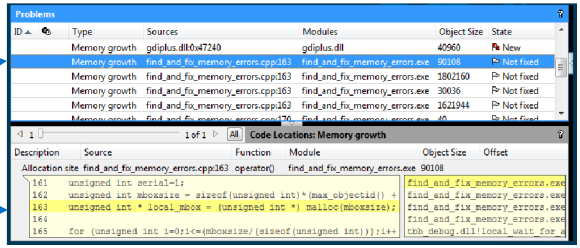
As your app is running…

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

SOFTWARE AND SERVICES

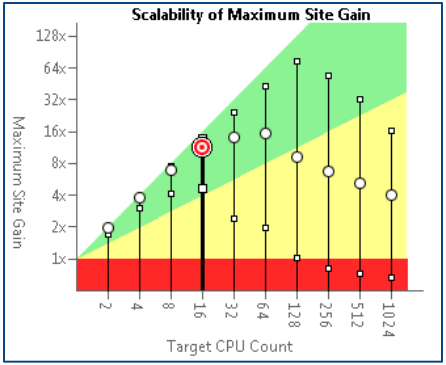
Intel® Advisor XE

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



http://intel.ly/advisor-xe

Add Parallelism with Less Effort, Less Risk and More Impact

Thread Prototyping
Intel® Advisor XE

Design Parallelism

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

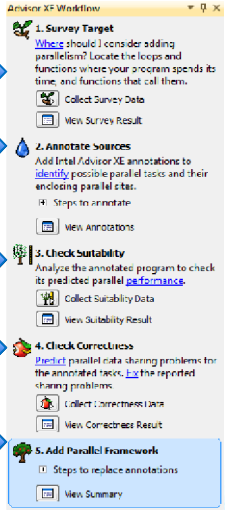
1) Analyze it.

2) Design it.
(Compiler ignores these annotations.)

3) Tune it.

4) Check it.

5) Do it!



Implement Parallelism

Less Effort, Less Risk, More Impact

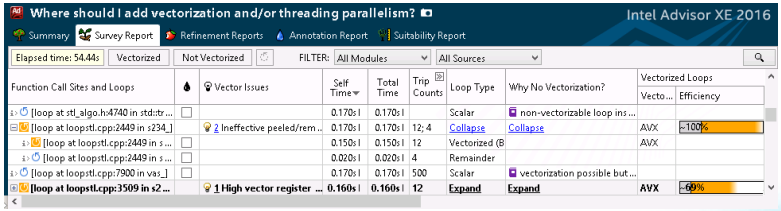
Vectorization Optimization
Intel® Advisor XE

Have you:

- Recompiled 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 use `pragma omp simd`?



High impact vectorization

Intel® Advisor XE

Filter by which loops are vectorized!

Trip Counts

What prevents vectorization?

Intel Advisor XE 2016

Summary | Survey Report | Refinement Reports | Annotation Report | Suitability Report

Elapsed time: 54.44s | Vectorized | Not Vectorized | FILTER: All Modules | All Sources

Function Call Sites and Loops	Vector Issues	Self Time	Total Time	Trip Counts	Loop Type	Why No Vectorization?	Vectorized Loops: Vecto...	Vectorized Loops: Efficiency	Vectorized Loops: Vector L...
[loop at stl_algo.h:4740 in std::tr...		0.170s	0.170s		Scalar	non-vectorizable loop ins...			
[loop at loopstl.cpp:2449 in s234_]	2 Ineffective peeled/rem...	0.170s	0.170s	12; 4	Collapse	Collapse	AVX	~100%	4
[loop a loopstl.cpp:2449 in s...		0.150s	0.150s	12	Vectorized (Body)		AVX		4
[loop loopstl.cpp:2449 in s...		0.020s	0.020s	4	Remainder				
[loop at opstl.cpp:7900 in vas_]		0.170s	0.170s	500	Scalar	vectorization possible but...			4
[loop a opstl.cpp:3509 in s2...	**1 High vector register ...**	**0.160s**	**0.160s**	**12**	**Expand**	**Expand**	**AVX**	**~69%**	**8**
[loop opstl.cpp:3891 in s279_]	2 Ineffective peeled/rem...	0.150s	0.150s	125; 4	Expand	Expand	AVX	~98%	8
[loop opstl.cpp:6249 in s414_]		0.150s	0.150s	12	Expand	Expand	AVX	~100%	4
[loo d_numeric.h:247 in std...	1 Assumed dependency...	0.150s	0.150s	49	Scalar	vector dependence prev...			

Focus on hot loops

What vectorization issues do I have?

Which Vector instructions are being used?

How efficient is the code?

Get Faster Code Faster! Intel® Advisor XE
Vectorization Optimization and Thread Prototyping

Intel® Parallel Studio XE 2016 cluster tools

Intel® MPI Library
Intel® Trace Analyzer and Collector

Intel® Xeon® and Intel® Xeon Phi™

SOFTWARE AND SERVICES

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

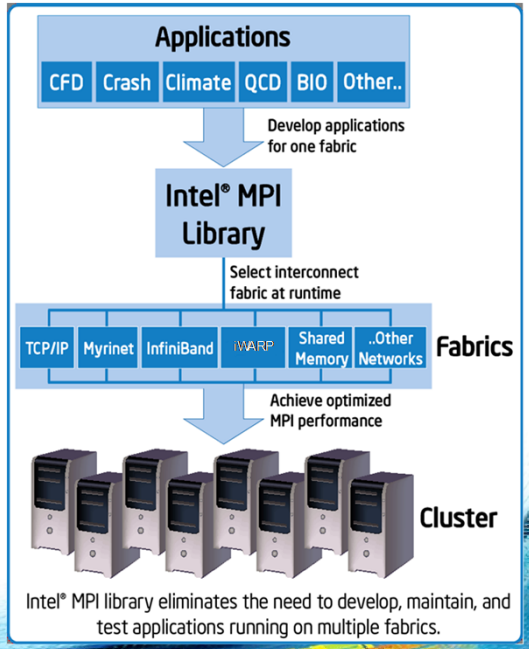
- Optimized collectives

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® MPI Library

What's New:

- Added support for OpenFabrics Interface* (OFI*) v1.0 API
- Added support for Fortran* 2008
- Updated the default value for **`I_MPI_FABRICS_LIST`**
- Added brand new Troubleshooting chapter to the Intel® MPI Library User's Guide
- Added new application-specific features in the Automatic Tuner and Hydra process manager
- Added support for the **`MPI_Pcontrol`** feature for improved internal statistics
- Increased the possible space for MPI_TAG
- Changed the default product installation directories
- Various bug fixes for general stability and performance
- Note: Support in Intel Fortran compiler for *draft* Fortran 2015 feature for interoperability with C specifically helps with MPI-3.

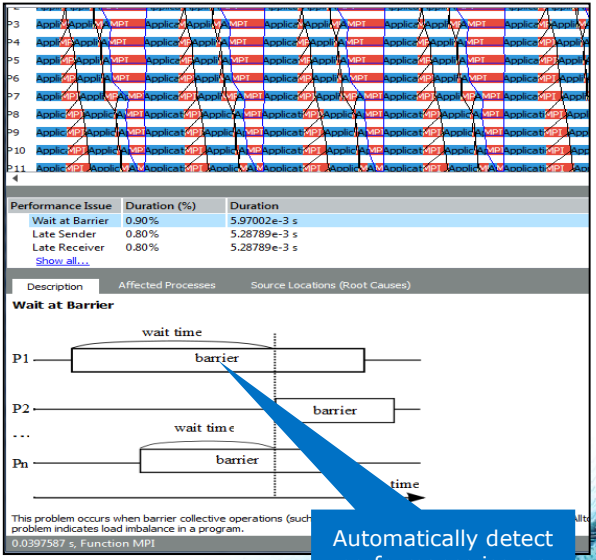
Intel® Trace Analyzer and Collector
Intel® MPI Library

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

Intel® Trace Analyzer and Collector
Intel® MPI Library

What's new:

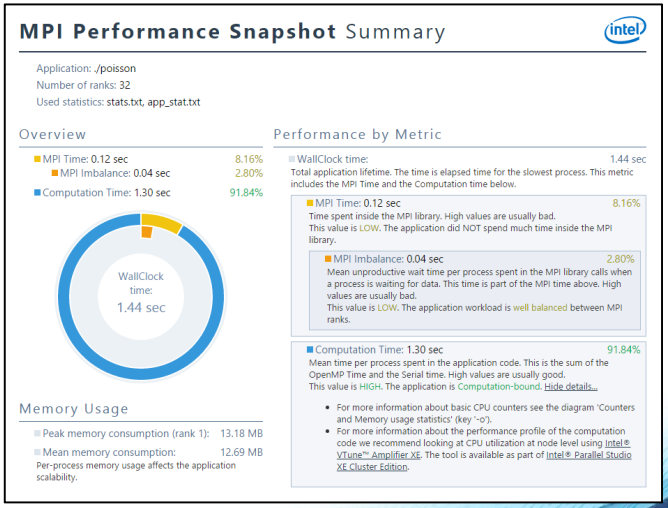
- Addition of MPI Performance Snapshot
  - Lightweight scalable MPI+OpenMP profiler
- Support for collection of CPI ad Memory Bound performance metrics
- Addition of new application summary details in the HTML report
- New command-line options
- The mps tool for statistical analysis is now available on Windows*
- Various bug fixes for general stability and performance

MPI Performance Snapshot
Intel® MPI Library

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



SOFTWARE AND SERVICES

intel®