



Released Sept 4th, 2013

Intel® Parallel Studio XE 2013 SP1 and Intel® Cluster Studio XE 2013 SP1

Flagship development suites expand standards and processor support to simplify creation of faster applications



Code the Future

New Product Announcements Embargoed until
September 4, 8am Pacific Time

Faster code + Simplified development



\$1,599-\$2,299

\$2,949

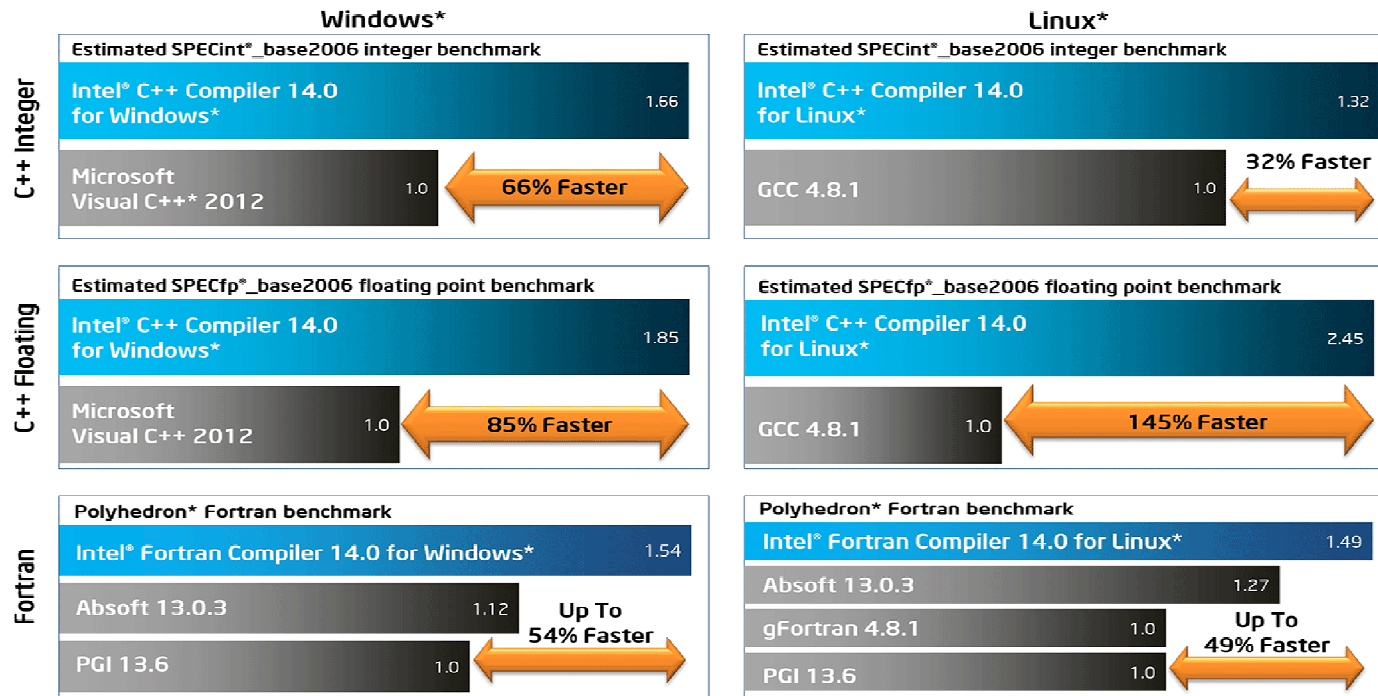
Shipping: Sept 4, 2013

More info at:
<http://intel.ly/perf-tools>

| Top New Features | | | |
|--|--|---|--|
| Performance | Analysis Efficiency | Productivity | Cross Platform Portability |
| <ul style="list-style-type: none"> Improved compiler and library performance Intel® AVX-512 ready Broadwell & Haswell-EP microarchitecture optimizations Windows* support for Intel® Xeon Phi™ coprocessor | <ul style="list-style-type: none"> Better data mining for performance tuning Simplified scalability testing for OpenMP* Incremental analysis and easier suppression management Enhanced MPI analysis interface | <ul style="list-style-type: none"> Improved conditional numerical reproducibility Enhanced GDB for Linux* and OS X* | <ul style="list-style-type: none"> OpenMP* 4.0 SIMD and target constructs Expanded C++11 Expanded Fortran 2003 & 2008 Improved MPI Performance and Scalability |



Performance for Compilers: Continued Performance Leadership



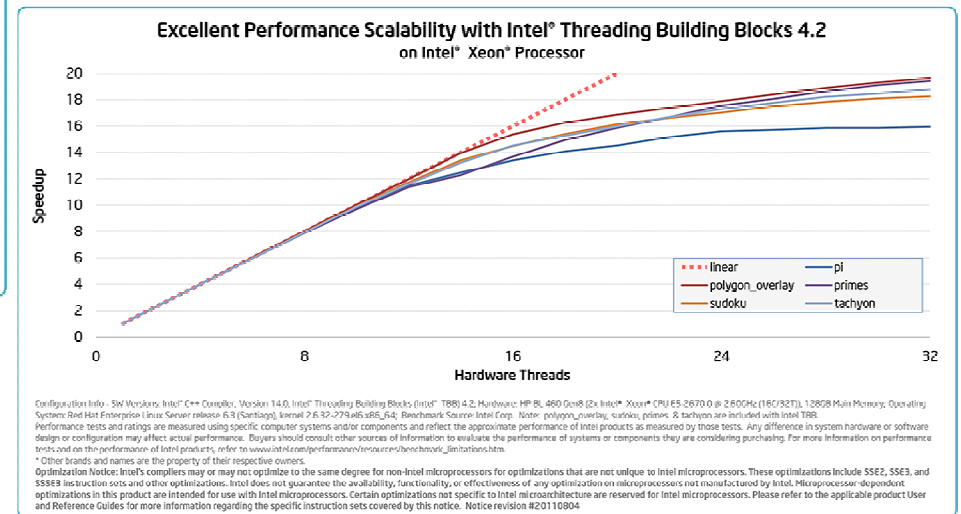
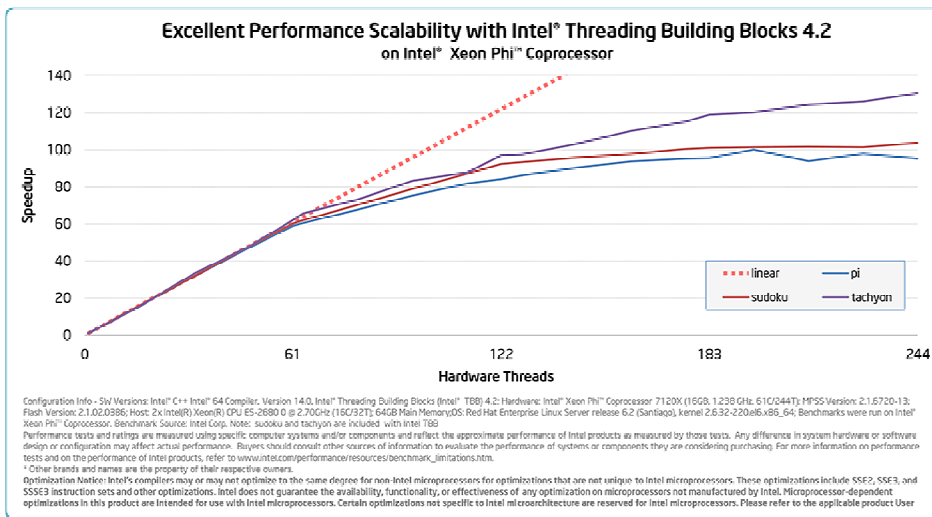
[Click here to view Intel* Compiler 14.0 benchmark configuration on page 22](#)

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

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

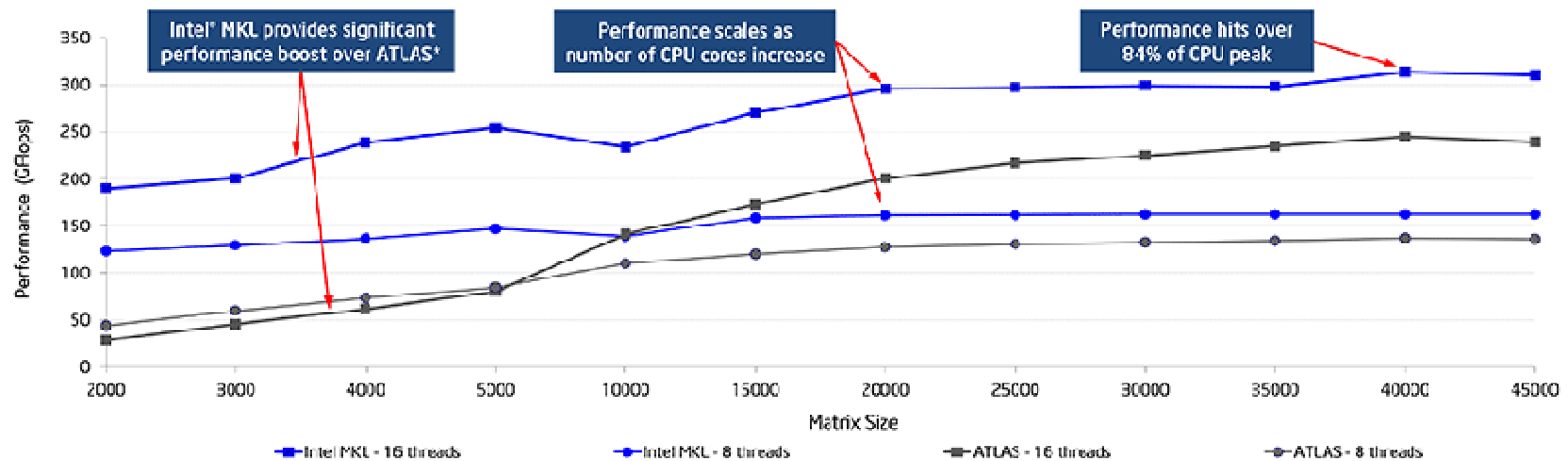


Performance with Intel® TBB: Excellent Scalability on Intel® Xeon® Processors and Intel® Xeon Phi™ Coprocessors



Performance with Intel® MKL: Superior Math Processing

Significant LAPACK Performance Boost using Intel® Math Kernel Library versus ATLAS*
 DGETF on Intel® Xeon® E5-2690 Processor

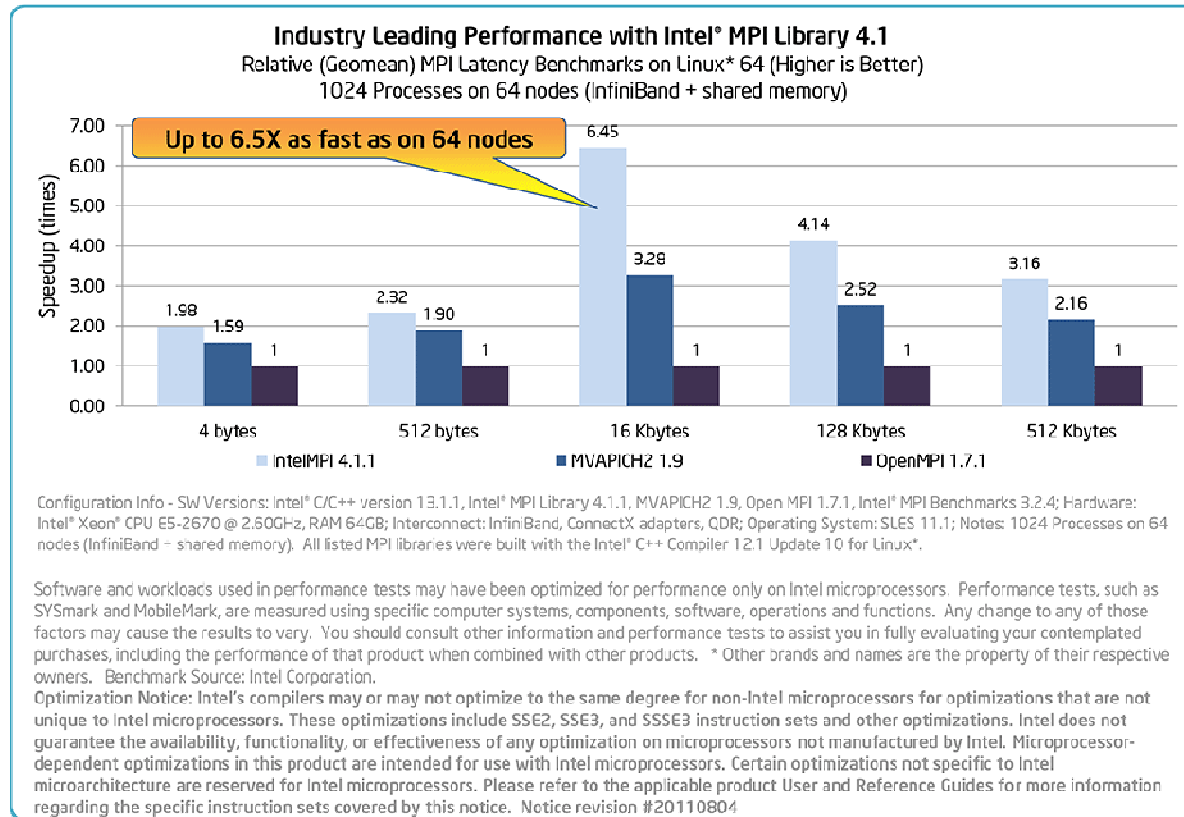


Configuration Info - Versions: Intel® Math Kernel Library (Intel® MKL) 11.1, ATLAS 3.10.1; Hardware: Intel® Xeon® E5-2690 Processor, 2 Eight-Core CPUs (20MB LLC, 2.9GHz), 32GB of RAM; Operating System: RHEL 6 SA x86_64.

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.

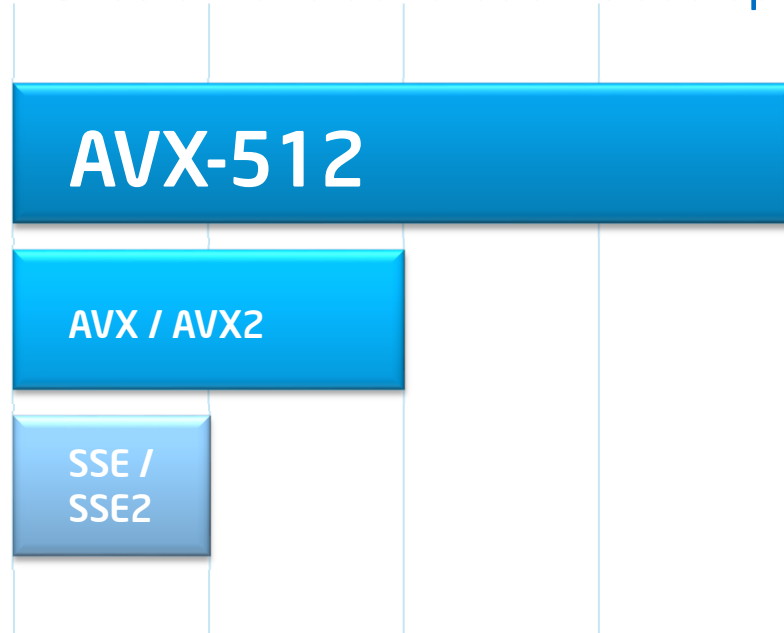
Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSE4 instruction sets and other optimizations. Intel does not guarantee the availability, 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.

Performance with Intel® MPI Library: Up to 6.5x Faster MPI Communications



Intel® AVX-512 Ready

Will Enable Next Generation Development for Intel® Xeon Phi™ Products



Peak single precision floating point performance

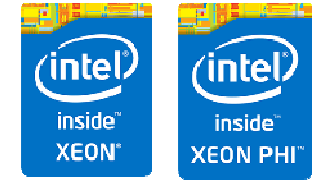
Intel® Compilers and Intel® Math Kernel Library AVX-512 support comes in Q4 update

- Significant performance available with 512-bit SIMD support
- Mixes well with AVX and AVX2
- Uses EVEX prefix to enable additional functionality
- First implemented in the future Intel® Xeon Phi™ coprocessor and processor, code named Knights Landing

Enables higher performance for the most demanding computational tasks

Microarchitecture Optimizations for the Latest Intel® Processors and Coprocessors

| | Intel® Haswell microarchitecture | Intel® Broadwell microarchitecture | Intel® Xeon Phi™ coprocessor |
|---------------------------------|----------------------------------|------------------------------------|------------------------------|
| Intel® C++ and Fortran Compiler | ✓ | ✓ | ✓ |
| Intel® TBB library | ✓ | ✓ | ✓ |
| Intel® MKL library | ✓ | ✓ | ✓ |
| Intel® MPI library | ✓ | ✓ | ✓ |
| Intel® VTune™ Amplifier XE† | ✓ | ✓ | ✓ |
| Intel® Inspector XE†† | ✓ | ✓ | ✓ |



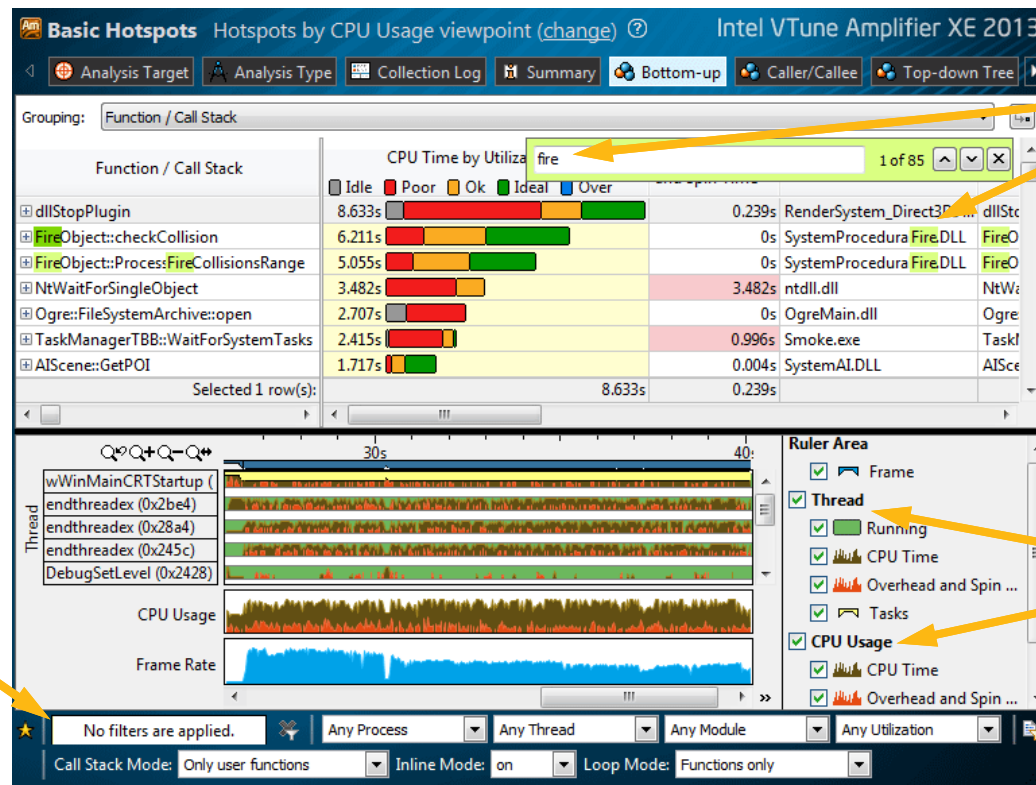
Windows* support for Intel® Xeon Phi™ Coprocessor

† Hardware events for new processors added as new processors ship.

†† Analysis runs on multicore processors, provides analysis for multicore and many-core processors.



Better Data Mining for Performance Tuning



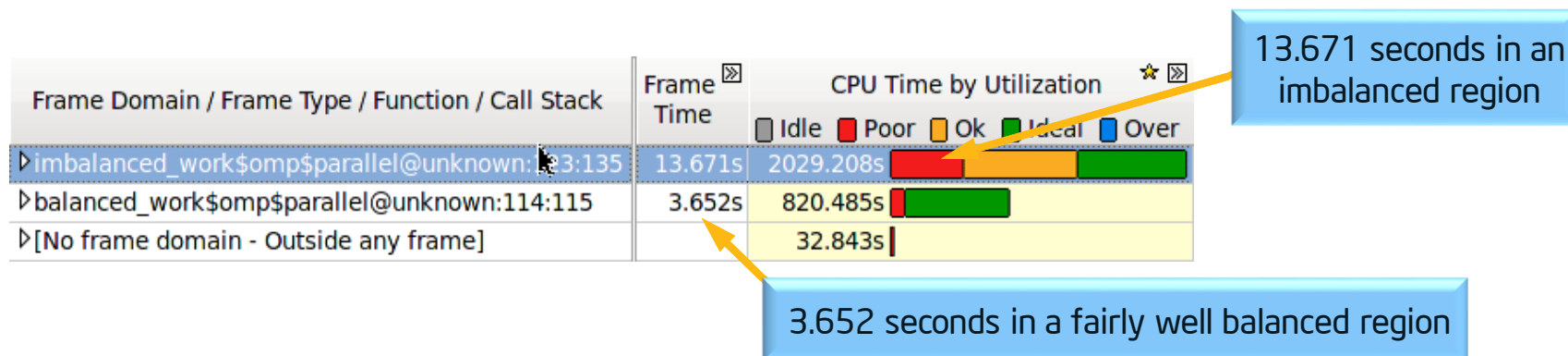
Search added to all grid views

Configurable:
 - Band Height
 - Sorting
 - Time Scale

Visualize overhead and spin times

Simplified scalability testing for OpenMP*

- Visualize time regions from the fork to the join
- See what is serial, balanced and imbalanced

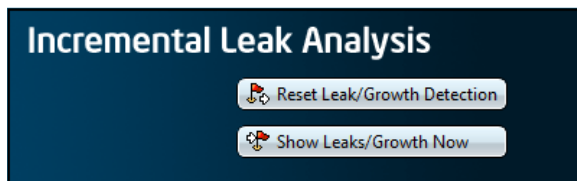


Actionable Data for Better OpenMP Scalability

Incremental Analysis and Easier Suppression Management

More ways to eliminate memory errors

Incremental Leak Analysis



Incremental leak reports, no waiting! Set a base line and see the leaks as they are detected.

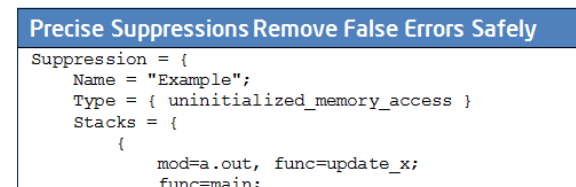
Import Suppressions from Purify* & Valgrind* on Linux*

```
bash-4.1$ inspxe-cl - convert-suppression-
file -from=valgrind.sup -to=inspector.sup

Converted old format or third-party
suppression file /tmp/my_app/valgrind.sup to
/tmp/my_app/inspector.sup.
```

Users of third party tools can leverage their investment creating suppression files

Improved Error Suppression



More precise, easy to edit, team shareable. Choose which stack frame to suppress. Eliminate the false, not the real errors

Enhanced MPI Analysis Interface

Streamlined MPI trace analysis

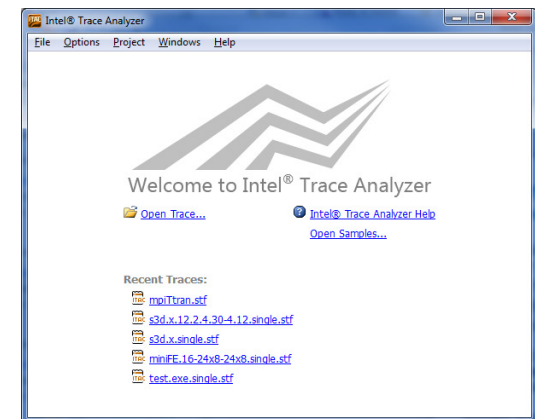
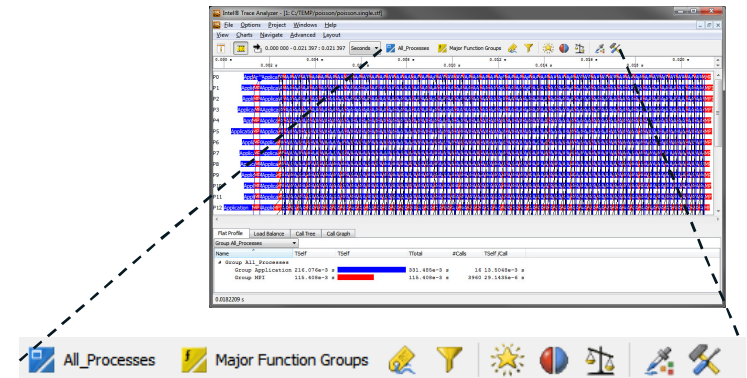
Fresh look-and-feel to the enhanced Intel® Trace Analyzer graphical interface

- More streamlined analysis flow
- Easy access to past projects

Support of dynamic profiling tool command

- MPI_PControl supported

New GUI-based installer on Linux*



Improved Conditional Numerical Reproducibility

Intel® Math Kernel Library now supports unaligned data for conditional numerical reproducibility

- Extends the feature to remove memory alignment as a prerequisite
- Balances performance with reproducible results
- Allows greater flexibility in code branch choice and ensures algorithms are deterministic

"I'm a C++ and Fortran developer and have high praise for the Intel® Math Kernel Library. One nice feature I'd like to stress is the bitwise reproducibility of MKL which helps me get the assurance I need that I'm getting the same floating point results from run to run."

*Franz Bernasek
CEO and Senior Developer, MSTC Modern Software Technology*

Enhanced GDB for Linux* and OS X*

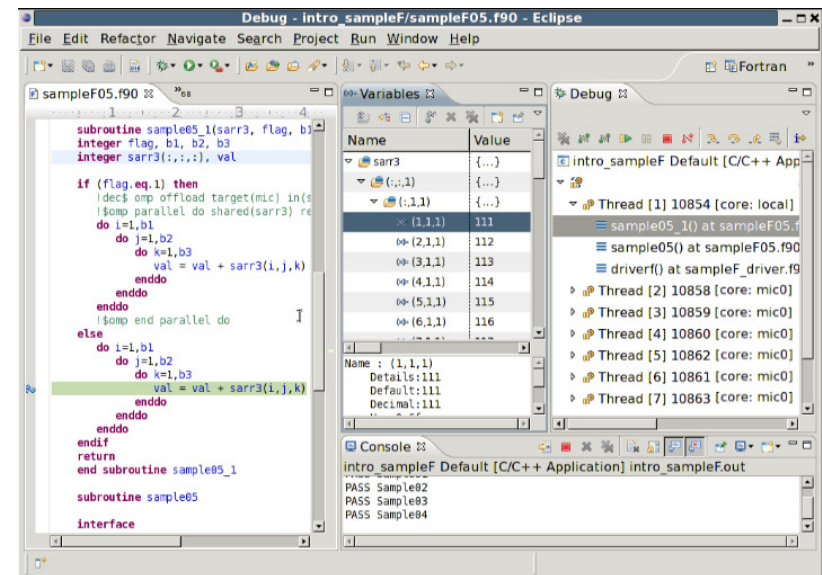
Debugs Code on Processors and Coprocessors

New debugger solution for Linux* and OS X* is based on GNU Project Debugger (GDB)* and includes additional improvements & features:

- Improved Fortran support
- GNU GDB 7.5 based

Intel® MIC Architecture Debugging

- Integration into Eclipse* IDE:
 - Supports C/C++ & Fortran
 - Support for offload extensions
 - Multiple coprocessor cards
- Command line debugging of native coprocessor applications also possible with GDB
- Included with Windows*, Linux*, and OS X*



OpenMP* 4.0 for SIMD and Target Constructs

- C++ and Fortran directives enable vectorization and offloading of execution on devices such as coprocessors or accelerators
 - TARGET Constructs enable creation of a data environment for attached devices, movement of data between host and devices, and execution of constructs on devices
 - SIMD Constructs enable loops and functions to be executed concurrently by a thread team using SIMD vector instructions
- See <http://openmp.org> and the OpenMP API Specification Version 4.0 RC2 for our current implementation of the supported features

Expanded C++ 11 support

New in the compiler:

- Unrestricted unions (Linux*, OS X*)
- Non-static data member initializers
- Explicit virtual overrides
- Allowing move constructors to throw
- Defining move special member functions
- Inline namespaces
- Rvalue references v2

Full list C++ 11 support:

<http://software.intel.com/en-us/articles/c0x-features-supported-by-intel-c-compiler>

Expanded Fortran capabilities

Fortran 2003

- Added support for user defined derived-type input and output

Fortran 2008

- ATOMIC_DEFINE and ATOMIC_REF
- Initialization of polymorphic INTENT(OUT) dummy arguments
- Standard handling of G format and of printing the value zero
- Polymorphic source allocation

Co-array now supports Intel® Xeon Phi™ coprocessor

Intel Fortran: Leadership performance with a leading feature set

Improved MPI performance and scalability

Improved MPI application performance and scalability

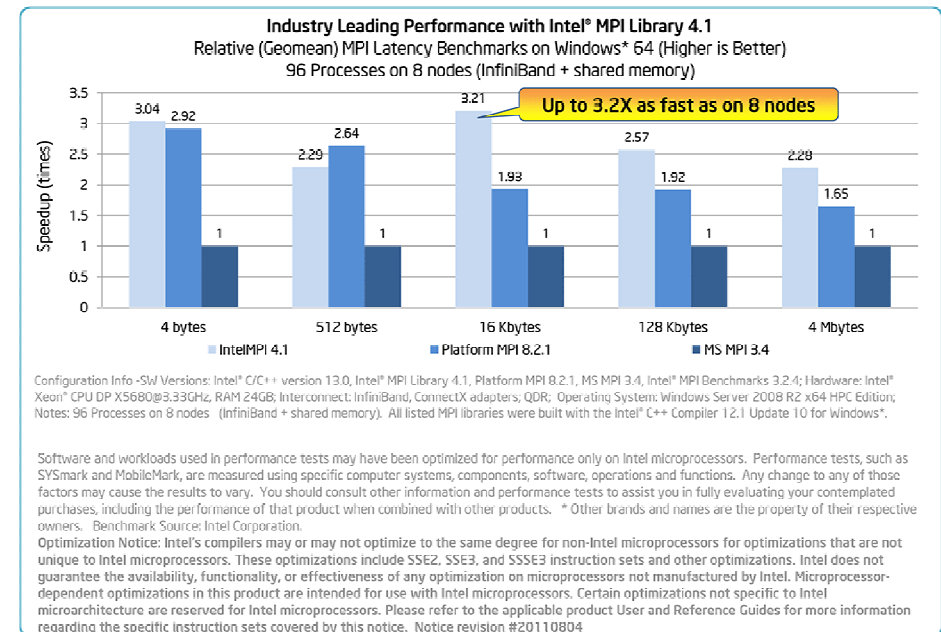
- Better Scalability at OFA fabric
- Improved support for NUMA applications and advanced process pinning controls
- Addition of a DAPL* auto-provider functionality for selecting best fabric at startup

Extended support for the Intel® Xeon Phi™ coprocessor architecture for improved bandwidth and latency through:

- Native port of the Tag Matching Interface (TMI) over the Qlogic* PSM fabric
- Extending support for Checkpoint/Restart (BLCR*) on the Intel Xeon Phi coprocessor

Backwards compatibility with existing Intel® MPI Library 4.x applications and Compliant with MPI 2.x

New GUI-based installer





Webinars - Technical Training Online!



Code the Future

New Product Announcements Embargoed until September 4,
8am Pacific Time

Introduction to High Performance Application Development for Multicore and Manycore-Live webinar- 2 Day Series

When: Pacific Time Zone - Day 1 - Sept 24th & Day 2 - Sep 25th

7:45am/8:00am Start

EMEA Time Zone (GMT+1) - Day 1 - Oct 8th & Day 2 - Oct 9th

8:45am/9:00am Start

Where: Online

Who: High Performance Application Developers

Agenda for the Days (Must Register for Each Day)

| Live Webinar Day 1 | |
|--------------------|--|
| Session 1 | Welcome and Introduction to Developing Applications for Intel® Xeon and Intel® Xeon Phi processors and coprocessors |
| Session 2 | Introduction to Intel® Xeon Phi™ coprocessor hardware and software architecture: native and offload execution basics |
| Session 3 | Compilation for Intel® Xeon Phi™ coprocessor: vectorization, programming models, alignment, pre-fetch, & more |
| Session 4 | Debugging on Intel® Xeon Phi™ coprocessor: using The GNU Project Debugger (GDB) |

| Live Webinar Day 1 | |
|--------------------|---|
| Session 1 | Intel® Math Kernel Library (Intel® MKL) on the Intel® Xeon Phi™ coprocessor |
| Session 2 | Message Passing Interface (MPI) on Intel® Xeon Phi™ coprocessor: special considerations for MPI on Intel Xeon Phi and Intel® Trace Analyzer & Collector |
| Session 3 | Performance analysis and events: Intel® VTune Amplifier introduction, GUI and command line, setup and collection, hot spots, bandwidth, events & more |
| Session 4 | Attendee Q&A, wrap-up |



Fall/Winter Technical Webinar Series

| Day | Time** | Title |
|-------------|----------------|---|
| 17-Sep | 9:00AM | Announcing Intel® Parallel Studio XE SP1 & Intel® Cluster Studio XE SP1 - What's New |
| 18-Sep | 9:00 AM | Powering up your Fortran Applications in the neo-heterogenous World with Intel® Xeon Processors & Intel® Xeon Phi™ Coprocessors on Linux and Windows* |
| Sep 24 & 25 | 6:45 & 7:00 AM | Intel® Xeon® & Xeon® Phi™- Introduction to High Performance Application Development for Multicore and Manycore-Live |
| 1-Oct | 9:00 AM | Become a C++ Hot Spot Hotshot with Intel® Parallel Studio XE |
| 8-Oct | 9:00AM | Finding the right fit for your application on Intel® Xeon and Intel® Xeon Phi™ processors |
| 15-Oct | 9:00 AM | Powered by MKL Accelerating NumPy and SciPy performance with Intel(r) MKL- Python |
| 22-Oct | 9:00AM | Introduction to OpenMP 4.0 for SIMD and affinity features with Intel® Xeon processors and Intel® Xeon Phi™ coprocessors |
| 29-Oct | 9:00 AM | Secrets of Performance Profiling – an introduction to Intel® VTune™ Amplifier XE |
| 29-Oct | 11:00 AM | Profiling MPI Communications - Tips and Techniques for High Performance |
| 30-Oct | 9:00 AM | Advanced Profiling with Intel® VTune™ Amplifier XE Part 1: Find the bottleneck |
| 30-Oct | 11:00 AM | Advanced Profiling with Intel® VTune™ Amplifier XE Part 2: Tune for Haswell (Sandy Bridge and Ivy Bridge) |
| 5-Nov | 9:00 AM | Software Architects: Design and prototype scalable threading using Intel® Advisor XE |
| 6-Nov | 9:00 AM | Precision Memory Leak Detection using the new on-demand leak detection in Intel® Inspector XE |
| 12-Nov | 9:00 AM | Porting and Tuning of Lattice QCD* and MPI-HMMER* for Intel® Xeon Processors & Intel® Xeon Phi™ Coprocessors |

**Time Zone – Pacific Time

Abstracts & Registration at - <http://software.intel.com/en-us/articles/intel-software-tools-technical-webinar-series>



Pricing and Availability



Shipping: September 4, 2013

| Includes | Intel® C++ Composer XE | Intel® Fortran Composer XE | Intel® Advisor XE | Intel® Inspector XE | Intel® VTune™ Amplifier XE | Intel® MPI Library | Intel® Trace Analyzer and Collector | Price |
|---------------------------|------------------------|----------------------------|-------------------|---------------------|----------------------------|--------------------|-------------------------------------|---------|
| Intel® Parallel Studio XE | ■ | ■ | ■ | ■ | ■ | | | \$2,299 |
| Intel® C++ Studio XE | ■ | | ■ | ■ | ■ | | | \$1,599 |
| Intel® Fortran Studio XE | | ■ | ■ | ■ | ■ | | | \$1,899 |
| Intel® Cluster Studio XE | ■ | ■ | ■ | ■ | ■ | ■ | ■ | \$2,949 |

Additional configurations including, floating and academic, are available at:
<http://intel.ly/perf-tools>

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.

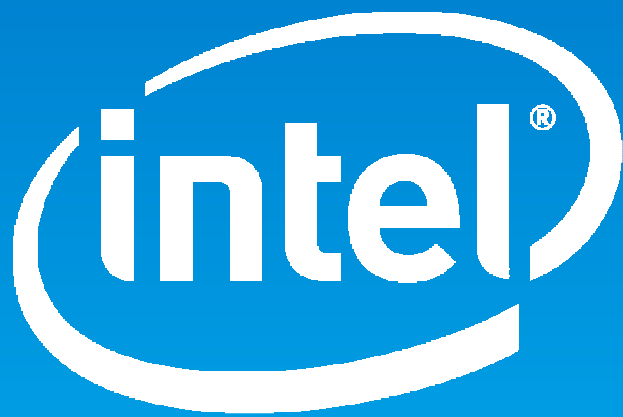
Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may 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 ©, Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, Xeon Phi, Core, VTune, and Cilk are trademarks of Intel Corporation in the U.S. and other countries.

Optimization Notice

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

Notice revision #20110804



Intel® Compiler 14.0 benchmark configuration

C++ benchmark configuration Info - SW Versions: Intel® C/C++ 14.0, Microsoft Visual C++ 2012 (Windows), GCC 4.8.1 (Linux); Hardware: HP ProLiant DL360 G8*, 2 x Intel® Xeon® processor E5-2670 (2.60GHz, 20480KB LLC, TurboBoost is on, HyperThreading is on), 128GB RAM, SAS; Windows Operating System: Windows 7 Enterprise, Service pack 1; Linux Operating System: Red Hat Enterprise Linux Server release 6.2 (Santiago), Kernel 2.6.32-220.el6.x86_64. ||| Compiler Options: INT Speed (Linux): Intel compiler 14.0- C: -xAVX -ipo -O3 -no-prec-div -static -parallel -opt-prefetch -auto-p32. C++: xAVX -ipo -O3 -no-prec-div -opt-prefetch -auto-p32. GCC 4.8.1- C: -m64 -Ofast -ffast-math -flto -march=native -mfpmath=sse -funroll-all-loops -static -ftree-parallelize-loops=16. C++: -m64 -Ofast -ffast-math -flto -march=native -mfpmath=sse -funroll-all-loops. ||| FP Speed (Linux): Intel Compiler 14.0- C: -xAVX -ipo -O3 -no-prec-div -static -parallel -opt-prefetch -ansi-alias. C++: -xAVX -ipo -O3 -no-prec-div -static -opt-prefetch -ansi-alias. GCC 4.8.1- C: -m64 -Ofast -ffast-math -flto -march=native -mfpmath=sse -funroll-all-loops -static -ftree-parallelize-loops=16. C++: -m64 -Ofast -ffast-math -flto -march=native -mfpmath=sse -funroll-all-loops. ||| INT Speed (Windows): Intel compiler 14.0- C: -Qvc11 -Qstd=c99 -QxAVX -Qipo -O3 -Qprec-div- -Qopt-prefetch -Qparallel -Qauto-ilp32. C++: -Qvc11 -QxAVX -Qipo -O3 -Qprec-div- -Qopt-prefetch -Qcxx_features -Qauto-ilp32. Visual C++* 2012- C: /O2 /Ob2 /GL /arch:AVX /favor:EM64T /fp:fast /Qpar. C++: /O2 /Ob2 /GL /arch:AVX /favor:EM64T /fp:fast /Qpar -EHsc -GR. ||| FP Speed (Windows): Intel compiler 14.0- C: -Qvc11 -Qstd=c99 -QxAVX -Qipo -O3 -Qprec-div- -Qopt-prefetch -Qansi-alias -Qparallel. C++: -Qvc11 -QxAVX -Qipo -O3 -Qprec-div- -Qopt-prefetch -Qcxx_features -Qansi-alias. Fortran: -QxAVX -Qipo -O3 -Qprec-div- -Qopt-prefetch -Qparallel.

Fortran benchmark configuration Info - Compiler Versions: Intel® Fortran 14.0, PGI 13.6, Absoft 13.0.3, gFortran 4.8.1; Hardware: Blue Hills ATX Media IVB Desktop DZ77BH-55K-IDD; Intel® Core™ i7-3770K CPU @ 3.50GHz, TurboBoost is on, HyperThreading is off, 16GB RAM; Windows Operating System: Windows 7 Enterprise, Service pack 1; Linux Operating System: Red Hat Enterprise Linux Server release 6.3 (Santiago); Kernel 2.6.32-279.el6.x86_64 ; Compiler Options (Windows and Linux): Intel Fortran compiler 14.0: ifort -O3 -fast -parallel -ipo -no-prec-div , PGI 13.6: pgf95 -fastsse -Munroll=n:4 -Mipa=fast,inline -Mconcur=bind . Polyhedron benchmark (www.polyhedron.com) performed by Intel Corp on August 20, 2013.