Announcing
Intel® Parallel Studio XE 2011 Service Pack 1

James Reinders
Director, Evangelist, Intel Software
james.r.reinders@intel.com
Intel® Parallel Studio XE 2011 Service Pack 1
Intel continues to be the best choice for C/C++/Fortran development tools

• Performance
  – Updated compilers and libraries produce industry leading performance.
    o Up to 47% faster for C/C++ compiler, or more?
    o Up to 24% faster for Fortran compiler, or more?
  – Intel Compiler 12.1 is first compiler for IA to support IEEE 754-2008 standard, and high performance!
  – The most popular Analysis Tools¹ just got better

• Forward scaling
  – Intel® Threading Building Blocks 4.0, commercially supported. Code using TBB scales exceptionally well.
  – Intel® Cilk™ Plus v1.1 implemented with commercial support; simplifies going parallel
  – Advanced tools to develop code for Intel® Xeon® Processors (today), easily extends to Intel® MIC architecture (future)

• Tools that developers count on
  – Expanded standards support
    o OpenMP* 3.1
    o Leading support for key parts of the latest Fortran and C++ standards
  – Enhanced compatibility
    o Visual Studio* 2010 Shell for Visual Fortran*

¹ Evans Data Corp. North American Development Survey 2011 Volume I
Updated compilers and libraries produce industry leading performance

- Intel v12.1 compilers improve performance compared with:
  - Competitive compilers
  - Previous version Intel compilers

<table>
<thead>
<tr>
<th></th>
<th>Intel v12.1 Compiler on Windows® vs. nearest competitor</th>
<th>Intel v12.1 Compiler on Linux® vs. nearest competitor</th>
<th>Intel v12.1 Compiler on Windows vs. v12.0</th>
<th>Intel v12.1 Compiler on Linux vs. v12.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/C++ Integer¹</td>
<td>47% faster</td>
<td>12% faster</td>
<td>11% faster</td>
<td>6% faster</td>
</tr>
<tr>
<td>C/C++ Floating Point¹</td>
<td>21% faster</td>
<td>9% faster</td>
<td>3% faster</td>
<td>1% faster</td>
</tr>
<tr>
<td>Fortran²</td>
<td>24% faster</td>
<td>17% faster</td>
<td>22% faster</td>
<td>27% faster</td>
</tr>
</tbody>
</table>

Notes:
¹ C/C++ performance measured using SPECint®_base2006 estimated RATE benchmark running on a 64 bit operating system
² Fortran performance measured using Polyhedron* benchmark running on a 64 bit operating system. In this performance measurement, "faster" refers to percent reduction in time-to-completion.
Intel Compiler 12.1 is first compiler for IA to support IEEE 754-2008 standard, and very high performance!

- First with IA conformance of both binary and decimal floating-point specs (radix-2 and radix-10)
  - IEEE Standard 754-2008 for Floating-Point Arithmetic
  - Technical Report ISO/IEC TR 24732, Extension for the programming language C to support decimal floating-point arithmetic
The most popular analysis tools¹ just got better

- **Intel® VTune™ Amplifier XE**
  - Attach to a running process is now available for both Windows* and Linux*
  - Project explorer for standalone version
    - Simplifies management and comparison of analysis results
- **Intel® Inspector XE**
  - Project explorer for standalone version
    - Simplifies management and comparison of analysis results
  - Improved performance for memory and thread checking

¹ EDC North American Development Survey 2011 Volume I
Advanced tools to develop code for Intel® Xeon® Processors today that easily extends to Intel® MIC architecture

“By just utilizing standard programming on both Intel® Xeon processor and Intel® MIC architecture based platforms, the performance met multi-threading scalability expectations and we observed near-theoretical linear performance scaling with the number of threads.” – Hongsuk Yi, Heterogeneous Computing Team Leader, KISTI Supercomputing Center

“SGI understands the significance of inter-processor communications, power, density and usability when architecting for exascale. Intel has made the leap towards exaflop computing with the introduction of Intel® Many Integrated Core (MIC) architecture. Future Intel® MIC products will satisfy all four of these priorities, especially with their expected ten times increase in compute density coupled with their familiar X86 programming environment.” – Dr. Eng Lim Goh, SGI CTO
Intel® Threading Building Blocks 4.0, commercially supported code using TBB scales exceptionally well

- **Flow Graph**
  - API Extends applicability of Intel® TBB to event-driven/reactive programming models

- **Concurrent Unordered Set**
  - Thread-safe container to store and access user objects

- **Memory Pools**
  - Enables greater flexibility and performance by getting thread-safe and scalable object allocation

- **Generic GCC* Atomics Support**
  - Library portability enables development of Intel® TBB-based solutions on a broader range of platforms

*- Other brands and names are the property of their respective owners.
Intel® Cilk™ Plus v1.1 implemented with commercial support; simplifies going parallel

- Enhanced performance and utilization of future Intel CPU features
- SIMD pragma loops, vector length, and elemental functions support
- Mac OS* support

```c
int fib(int n)
{
    if (n <= 2)
        return n;
    else {
        int x,y;
        x = fib(n-1);
        y = fib(n-2);
        return x+y;
    }
}
```

```c
cilk_for (int i=0; i<n; ++i) {
    Foo(a[i]);
}
```

Parallel loops made easy

Turn serial code

Into parallel code

Open spec at: cilkplus.org
Tools that developers count on

“I use the Intel Fortran compiler in Intel® Fortran Composer XE, and I very much like the new Object Browser. It makes it much easier to navigate in modules that have many routines. The compiler is stable and offers outstanding performance. We are a small company but big proponents of Intel Fortran!”

Warner Weiss, Manager, Sugars International LLC

“Tried the latest Intel Cilk Plus and liked the lower overhead from using Intel Cilk Plus spawning compared to OpenMP* task. The concepts behind Cilk Plus – simplification of adding parallelism – is really great.”

David Carver, Texas Advanced Computing Center, University of Texas at Austin

“The Intel tools provided an excellent return on investment. Intel® Parallel Inspector allowed the code to be validated as ‘data race-free’ on our validation suite, and Intel® Parallel Amplifier allows us to focus efforts on the hotspots.”

Andrew Cunningham Technical Staff Member, ESI Group

“Standards support is important to us so I’m glad to see the extended Fortran 2003 support in the Intel® Fortran compiler that ships in Intel® Fortran Composer XE - and the Fortran 2008 additions are also welcome. We’ve found the Intel Fortran compiler to be stable and offer outstanding performance. Please keep the updates coming”

Simon Geard, Technical Lead, CAD Schroer UK Ltd.

“The new interface is a joy to use. Intel® Vtune™ Amplifier XE gives us precise, down-to-the-metal performance data that’s invaluable for pinpointing hotspots and evaluating the effect of optimizations”

Daniel Schwarz, Performance Engineer Nik Software reference customer case study

"Using Intel TBB’s new flow graph feature, we accomplished what was previously not possible, parallelize a very sizable task graph with thousands of interrelationships - all in about a week."

Robert Link, GCAM Project Scientist, Pacific Northwest National Laboratory
Tools that developers count on

• Expanded standards support
  – OpenMP* 3.1
  – Key portions of the latest Fortran* and C++ standards
    o C++ Variadic templates
    o Lambda support
    o Fortran coarray support for distributed-memory systems
  – IEEE Standard 754-2008 for Floating-Point Arithmetic

• Enhanced compatibility
  – Visual Studio* 2010 Shell for Visual Fortran

“Especially liked the C++0x support in the Intel® C++ compilers, tested variadic templates function defaults and was happy to see the SFINAE for expressions. For library writers, this is a very powerful feature and can greatly simplify the mess of template meta-programming.”

Jesse Perla, PhD Candidate, New York University
# Pricing and availability

<table>
<thead>
<tr>
<th>Includes</th>
<th>C/C++ compiler</th>
<th>Fortran* compiler</th>
<th>For Linux*</th>
<th>For Windows*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® Parallel Studio XE 2011 SP1</td>
<td>✔️</td>
<td>✔️</td>
<td>$2249</td>
<td>$1899</td>
</tr>
<tr>
<td>Intel® C++ Studio XE 2011 SP1</td>
<td>✔️</td>
<td></td>
<td>$1499</td>
<td>$1499</td>
</tr>
<tr>
<td>Intel® Fortran*Studio XE 2011 SP1</td>
<td></td>
<td>✔️</td>
<td>$1799</td>
<td>$1599</td>
</tr>
<tr>
<td>Intel® Visual Fortran Composer XE 2011 with IMSL* for Windows*</td>
<td></td>
<td>✔️</td>
<td>NA</td>
<td>$1699</td>
</tr>
</tbody>
</table>

Additional configurations including floating and academic are available at www.intel.com/software/products
Intel® Visual Fortran Composer XE 2011 with IMSL* for Windows*

• Same 12.1 compiler that is available separately
  – Highly optimizing Fortran compiler featuring scalable multi-threading with OpenMP, and introducing coarray Fortran (part of the Fortran 2008 standard), including new parallelism models
  – VAX Fortran and Compaq Visual Fortran compatibility
  – In addition to IMSL, it includes Intel® Math Kernel Library (Intel® MKL)

• Rogue Wave IMSL® 6.0 Math Library highlights
  – Over 1,000 mathematical and statistical algorithms for developers of Fortran applications
  – Support for shared memory and distributed memory computing environments
  – High performance linear programming optimizer
  – ScalAPACK integration for MPI, LAPACK integration for SMP
  – New probability density functions and inverses
  – Time series and forecasting additions
  – New Deployment Licensing terms and pricing
Intel® Parallel Studio XE 2011 Service Pack 1

Intel continues to be the best choice for C/C++/Fortran development tools

- **Performance**
  - Updated compilers and libraries produce industry leading performance.
    - Up to 47% faster for C/C++ compiler, or more?
    - Up to 24% faster for Fortran compiler, or more?
  - Intel Compiler 12.1 is first compiler for IA to support IEEE 754-2008 standard, and high performance!
  - The most popular Analysis Tools¹ just got better

- **Forward scaling**
  - Intel® Threading Building Blocks 4.0, commercially supported. Code using TBB scales exceptionally well.
  - Intel® Cilk™ Plus v1.1 implemented with commercial support; simplifies going parallel
  - Advanced tools to develop code for Intel® Xeon® Processors (today), easily extends to Intel® MIC architecture (future)

- **Tools that developers count on**
  - Expanded standards support
    - OpenMP* 3.1
    - Leading support for key parts of the latest Fortran and C++ standards
  - Enhanced compatibility
    - Visual Studio* 2010 Shell for Visual Fortran*

---

¹ Evans Data Corp. North American Development Survey 2011 Volume I

Copyright© 2011, Intel Corporation. All rights reserved.

*Other brands and names are the property of their respective owners.
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.

Performance tests and ratings are measured using specific computer systems and/or components and reflect
the approximate performance of Intel products as measured by those tests. Any difference in system hardware
or software design or configuration may affect actual performance. Buyers should consult other sources of
information to evaluate the performance of systems or components they are considering purchasing. For more
information on performance tests and on the performance of Intel products, reference

Copyright © 2011, Intel Corporation. All rights reserved. Intel, the Intel logo, Xeon, Core, VTune, and Cilk are
trademarks of Intel Corporation in the U.S. and other countries. *Other names and brands may be claimed as
the property of others.

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
Benchmarks
For an Intel processor on Linux* 64, Intel Compilers (new and old) vs. best of Pathscale*, Open64* and GCC

### Industry Leading Performance using the Intel® C/C++ Compiler

**Intel® Xeon™ Processor running on Linux* 64 (Higher is Better)**

<table>
<thead>
<tr>
<th>Compiler</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® C++ Compiler 12.1 for Linux*</td>
<td>1.12X</td>
</tr>
<tr>
<td>Intel® C++ Compiler 12.0 for Linux</td>
<td>1.06X</td>
</tr>
</tbody>
</table>

**Next Best Compilers**

- Best of Pathscale 3.2, Open64 4.2.5 and gcc 4.6.0 C++ Compilers

**Estimated SPECint*_base2006_rate C/C++ floating point benchmark**

<table>
<thead>
<tr>
<th>Compiler</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® C++ Compiler 12.1 for Linux*</td>
<td>1.12X</td>
</tr>
</tbody>
</table>

**Next Best Compilers**

- Best of Pathscale 3.2, Open64 4.2.5 and gcc 4.6.0 C++ Compilers

**Estimated SPECfp*_base2006_rate C/C++ integer benchmark**

<table>
<thead>
<tr>
<th>Compiler</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® C++ Compiler 12.1 for Linux*</td>
<td>1.09X</td>
</tr>
</tbody>
</table>

**Next Best Compilers**

- Best of Pathscale 3.2, Open64 4.2.5 and gcc 4.6.0 C++ Compilers

**Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to www.intel.com/technology/performance/benchmark_limitations.htm.**

**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
For an Intel processor on Windows* 64, Intel Compilers (new and old) vs. best of Microsoft & PGI

Industry Leading Performance using the Intel® C/C++ Compiler
Intel® Xeon™ Processor running on Windows* 64 (Higher is Better)

Estimated SPECint*_base2006_rate C/C++ floating point benchmark

<table>
<thead>
<tr>
<th>Compiler</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® C++ Compiler 12.1 for Windows*</td>
<td>1.47X</td>
</tr>
<tr>
<td>Intel® C++ Compiler 12.0 for Windows</td>
<td>1.33X</td>
</tr>
<tr>
<td>Next Best Compilers</td>
<td>Baseline</td>
</tr>
<tr>
<td>Best of Microsoft Visual Studio® 2010 and PGI® C++ Compiler 11.2</td>
<td>1.00X</td>
</tr>
</tbody>
</table>

47% Faster

Estimated SPECfp*_base2006_rate C/C++ integer benchmark

<table>
<thead>
<tr>
<th>Compiler</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® C++ Compiler 12.1 for Windows*</td>
<td>1.21X</td>
</tr>
<tr>
<td>Intel® C++ Compiler 12.0 for Windows</td>
<td>1.18X</td>
</tr>
<tr>
<td>Next Best Compilers</td>
<td>Baseline</td>
</tr>
<tr>
<td>Best of Microsoft Visual Studio® 2010 and PGI® C++ Compiler 11.2</td>
<td>1.00X</td>
</tr>
</tbody>
</table>

21% Faster

Configuration Info - SW Versions: Intel® C/C++ version 12.1; Hardware: Intel® Xeon® CPU X5670 @ 2.93GHz, 2x2.93GHz, RAM 48GB, CACHE 12288KB; Operating System: Windows 2008 x64 SP2; Benchmark Source: Intel Corp.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and the performance of Intel products, refer to www.intel.com/performance/resources/benchmark_limitations.htm. *Other brands and names are the property of their respective owners.

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
For an Intel processor on Windows* 64, Intel Compilers (new and old) vs. best of PGI and Absoft

<table>
<thead>
<tr>
<th>Industry Leading Performance using the Intel® Fortran Compiler</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® Core™ i7 Processor running on Windows* 64 (Lower is Better)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Polyhedron* Fortran benchmark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel® Fortran Compiler 12.1 for Windows*</td>
</tr>
</tbody>
</table>

| Intel® Fortran Compiler 12.0 for Windows | 0.98 |

<table>
<thead>
<tr>
<th>Next Best Compilers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best of PGI 11.4 and Absoft 11.1</td>
</tr>
</tbody>
</table>

Baseline

*Other brands and names are the property of their respective owners.

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
For an Intel processor on Linux* 64, Intel Compilers (new and old) vs. best of PGI and Absoft

Industry Leading Performance using the Intel® Fortran Compiler
Intel® Core™ i7 Processor running on Linux* 64 (Lower is Better)

Polyhedron® Fortran benchmark

Intel® Fortran Compiler 12.1 For Linux* 0.83

17% less time

Intel® Fortran Compiler 12.0 for Linux 1.13

Next Best Compilers
Best of PGI 11.4 and Absoft 11.1 Baseline

Configuration Info - SW Versions: Intel® C/C++ version 12.1; Hardware: Intel® Core™ i7-2600 CPU @ 3.40GHz 3.40GHz RAM: 16.0 GB (1-socket Desktop); Operating System: Red Hat Enterprise Linux Server release 6.0 (Santiago); Kernel 2.6.32-71.el6.x86_64; Benchmark Source: Intel Corp.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to www.intel.com/performance/resources/benchmark_limitations.htm. * Other brands and names are the property of their respective owners.

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
For an Intel processor, Intel IPP vs. OpenSSL open source solution highlighting AES

Intel Integrated Performance Primitives (Intel IPP) 7.0.2 OpenSSL Provides Better Performance than Original OpenSSL (Higher is Better)

Use of AES New Instructions results in up to 2.45X better performance!

Configuration Info - Versions: Intel® Integrated Performance Primitives (Intel® IPP) 7.0.2, gcc 3.4.6; Hardware: Intel® Core® i7-2600K Processor, 3.40 GHz, 8MB cache, 4GB Memory; Operating System: Linux Red Hat Enterprise Linux Server release 6.0 (Santiago) 2.6.32-71.el6.x86_64 #1 SMP; Benchmark Source: Intel Corporation. Benchmark comments: OpenSSL-0.9.8; AES-128-CBC.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by these tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to www.intel.com/performanceresources/benchmark_limitations.htm.

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

Copyright © 2011, Intel Corporation. All rights reserved.

*Other brands and names are the property of their respective owners.
For an Intel processor, Intel IPP vs. original gzip yields faster encode and decode.
For an Intel processor, LAPACK Performance Intel MKL compared with ATLAS* open source solution

![LAPACK Performance Improves using Intel Math Kernel Library versus ATLAS* DGETRF](image)

Intel® MKL offers significant performance increase over ATLAS*

Performance scales as number of CPU cores increase

Performance up to 67% of CPU Gflop/s peak!

Configuration Info - Versions: Intel® Math Kernel Library (Intel® MKL) 10.3.1 FFTW3.2.2; Hardware: Intel® Xeon® I7-2600 Processor, 3.40Ghz, 8 MB L2 cache, 4 GB Memory; Operating System: Fedora 14 x86_64; Benchmark Source: Intel Corp.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by these tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to www.intel.com/performanceresources/benchmark_limitations.htm. * Other brands and names are the property of their respective owners.

Optimization Notice: Intel’s compilers may or may not optimize to the same degree for non-QNIntel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804
For an Intel processor, 3D FFT performance Intel MKL compared with FFTW*

3D FFT Performance Improves using Intel® Math Kernel Library (Intel®MKL) versus FFTW*

Performance scales as number of CPU cores increases
Performance is good even when problem size exceeds cache memory

Intel® MKL offers significant performance increase over FFTW*

Transform Size (Power of two): Single-precision, Complex, In-place

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to www.intel.com/performance/resources/benchmark_limitations.htm. * Other brands and names are the property of their respective owners.

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
For an Intel processor, 2D FFT performance
Intel MKL compared with FFTW*

2D FFT Performance Improves using
Intel® Math Kernel Library (Intel® MKL) versus FFTW*

Performance scales as number of CPU cores increase

Intel® MKL offers significant performance increase over FFTW*

Configuration Info - Versions: Intel® Math Kernel Library (Intel® MKL) 10.3.1 FFTW3.2.2; Hardware: Intel® Xeon® i7-2600 Processor, 3.40Ghz, 8 MB L2 cache, 4 GB Memory; Operating System: Fedora 14 x86_64; Benchmark Source: Intel Corp.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to www.intel.com/performance/resources/benchmark_limitations.htm. * Other brands and names are the property of their respective owners

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 #20110604
For an Intel processor, DGEMM Performance
Intel MKL vs. ATLAS* open source solution

Performance Improves using
Intel® Math Kernel Library (Intel® MKL) versus ATLAS - DGEMM

Intel® MKL offers ~2x performance increase over ATLAS*
Performance scales as number of CPU cores increase
Performance up to 93% of CPU Gflop/s peak!

Configuration Info - Versions: Intel® Math Kernel Library (Intel® MKL) 10.3.1 ATLAS 3.8.3; Hardware: Intel® Core® i7-2600 Processor, 3.4Ghz, 8 MB L2 cache, 4 GB Memory; Operating System: Fedora 14 x86_64; Benchmark Source: Intel Corporation.

Performance tests and ratings are measured using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance. Buyers should consult other sources of information to evaluate the performance of systems or components they are considering purchasing. For more information on performance tests and on the performance of Intel products, refer to
www.intel.com/performance/resources/benchmark_limitations.htm  * Other brands and names are the property of their respective owners

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
OpenMP* 3.1 details

• Intel Compiler 12.1 support of new capabilities available in OpenMP 3.1
  – taskyield directive
  – task clauses final and mergeable
  – atomic read/write/update/capture
  – reduction min/max