Using Intel® VTune™ Amplifier XE for High Performance Computing
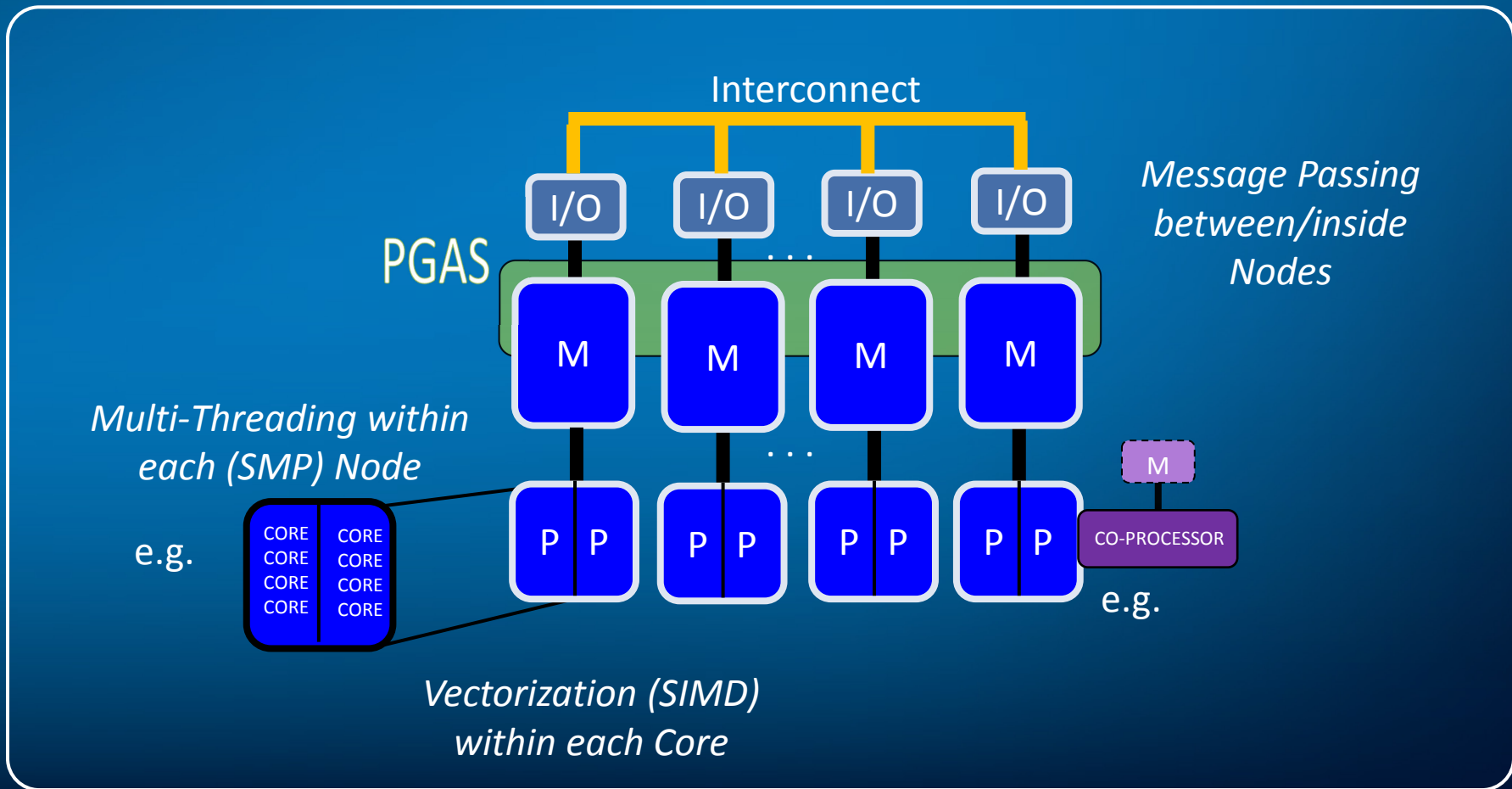
Vladimir Tsymbal

Performance, Analysis and Threading Lab

Software & Services Group, Developer Products Division

intel Software

The Majority of all HPC-Systems are Clusters

Interconnect

Message Passing between/inside Nodes

PGAS

Multi-Threading within each (SMP) Node

e.g.

CORE CORE
CORE CORE
CORE CORE
CORE CORE

Vectorization (SIMD) within each Core

I/O I/O ... I/O I/O

M M M M

P P P P P P P P

M

CO-PROCESSOR

e.g.

High Performance Software Products
Supporting Multicore and Many-core Development

Intel® Parallel Studio XE*
Advanced Performance

Intel® Cluster Studio XE*
Distributed Performance
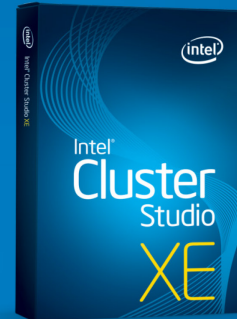
Intel® C/C++ and Fortran Compilers
w/OpenMP

Intel® MKL, Intel® Cilk Plus, Intel® TBB Library, Intel® ArBB Library Intel® IPP Library

Intel® Inspector XE, **Intel® VTune™ Amplifier XE**, Intel® Advisor

Intel® MPI Library

Intel® Trace Analyzer and Collector

Intel® Parallel Studio XE

Performance. Scale Forward. Proven

MPI Analysis
VTune Amplifier XE and MPI interoperability

- To analyze the performance and correctness of an MPI program at the <u>inter-process</u> level
  - use the Intel® Trace Analyzer and Collector
    - collects information about events at the MPI level between processes
    - allows analyzing the performance and correctness of the MPI calls, deadlock detection, data layout errors, as well as risky or incorrect MPI constructs
    - The Intel Trace Analyzer and Collector data is correlated and aggregated across all processes and all nodes, that participated in the execution run

MPI Analysis (ITAC)
Hybrid program: 2 MPI processes + 12 Threads per process
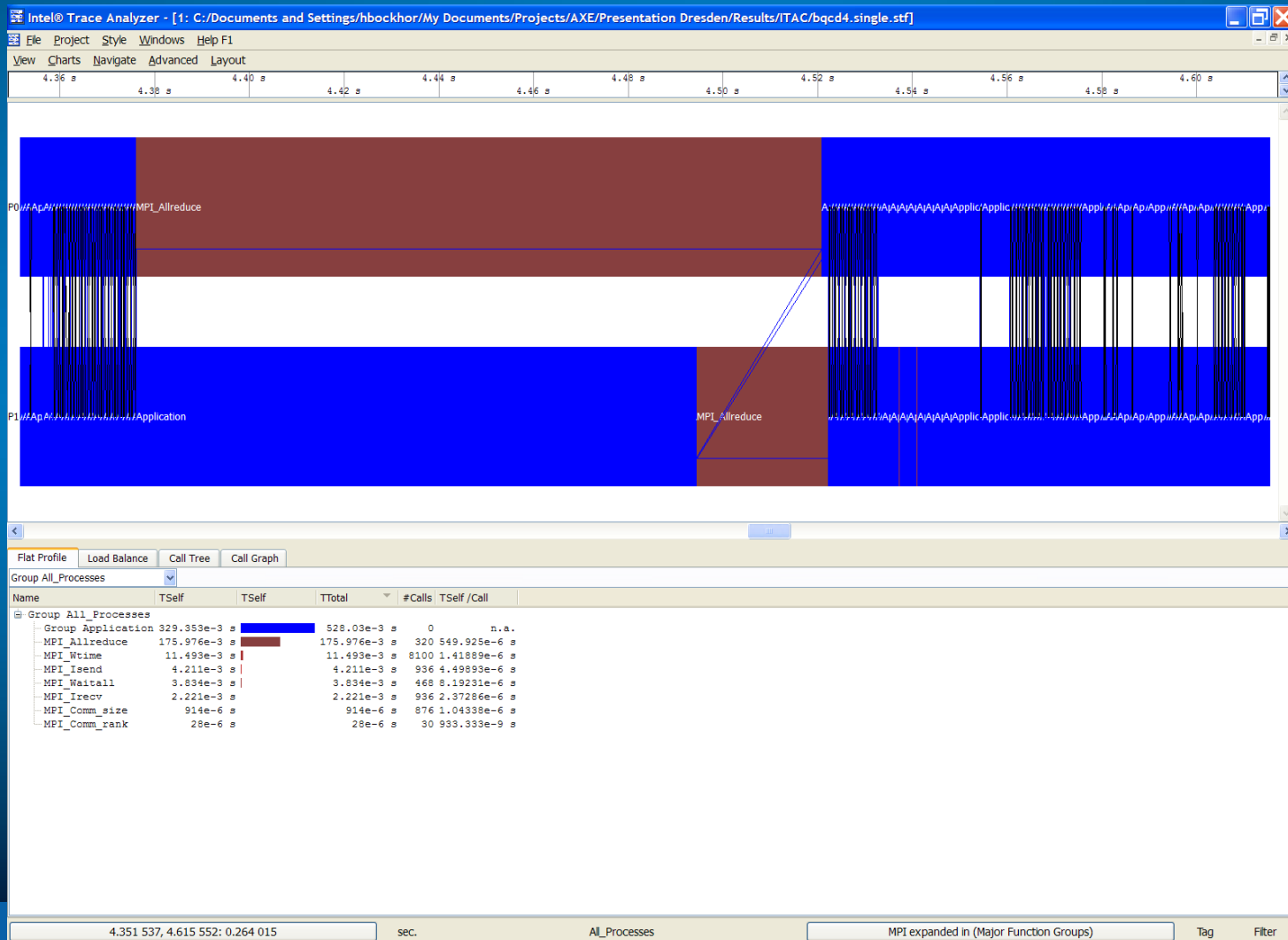
MPI Analysis
VTune Amplifier XE and MPI interoperability

- Hybrid programming
  - Beyond the inter-process level of MPI parallelism, the processes that make up the programs on a modern cluster often also use fork-join threading through OpenMP* and Intel® TBB

  - VTune Amplifier XE performance analyzer and the Intel Inspector XE checker can be used to analyze the performance and correctness of an MPI program

Hybrid Analysis in 2 steps

1. Use the 'amplxe-cl' command line tools to collect data

   – By default, all processes are analyzed, but it is possible to filter the data collection to limit it to a subset of processes.

   – An individual result directory is created for each spawned MPI program process that was analyzed with MPI process rank value captured.

   – Post-processing, also called "finalization" or "symbol resolution", is done automatically for each result directory once the collection has finished.

2. Open and analyze each result directory through the GUI standalone viewer

(intel)
Software

Hybrid Analysis
Collecting Performance Data

$ mpirun –n <N> amplxe-cl –r my_result -collect <analysis type> my_app [my_app_ options]

where <abbr> is "amplxe" or "inspxe"

- The list of analysis types available can be viewed using "amplxe-cl –help collect" command.
- For hotspots: an example of full command line for collection would be:

$ mpirun –n 4 amplxe-cl –r my_result –collect hotspots -- my_app 12

- Number of result directories will be created in the current directory, named as my_result.0 – my_result.3

Hybrid Analysis
Collecting Performance Data

- Sometimes it is necessary to collect data for a subset of the MPI processes in the workload.

- Example: 16 processes in the job distributed across the hosts and hotspots data should be collected for only two of them:

$ mpirun –host myhost -n 14 ./a.out : -host myhost -n 2 amplxe-cl –r foo -c hotspots ./a.out

  - 2 directories will be created in the current directory: foo.14 and foo.15 (given that process ranks 14 and 15 were assigned to the last 2 processes in the job).

intel Software

Hybrid Analysis
Finalizing the Collected Data

- Finalization of the data (symbol resolution, conversion to the database) happens automatically after the collection has finished

  - happens on the same compute node where the command line collection was executing
  - Binaries and symbol files will be located automatically
  - In cases where the search algorithm needs to be adjusted (common reason: need to point to symbol files stored elsewhere), the –search-dir option should be used with amplxe-cl as follows:

$ mpirun –np 128 amplxe-cl –q -collect hotspots –search-dir sym=/home/foo/syms ./a.out

intel

Software

Hybrid Analysis
Viewing the Collected Data

- Once the results are collected, the user can open any of them in the standalone GUI or generate a command line report.
  - Use amplxe-cl –help report to see the options available for generating reports.
  - Here is an example of viewing the text report for functions and modules after a VTune Amplifier XE analysis:

```
$ amplxe-cl -R hotspots -q -format text –r r003hs
        Function  Module   CPU Time

        ------------     ----------  ---------------

        F          a.out      6.070
        Main       a.out      2.990
$ amplxe-cl -R hotspots -q -format text -group-by module –r r003hs
        Module    CPU Time

        ----------  ---------------

        a.out      9.060
```

Hybrid Analysis
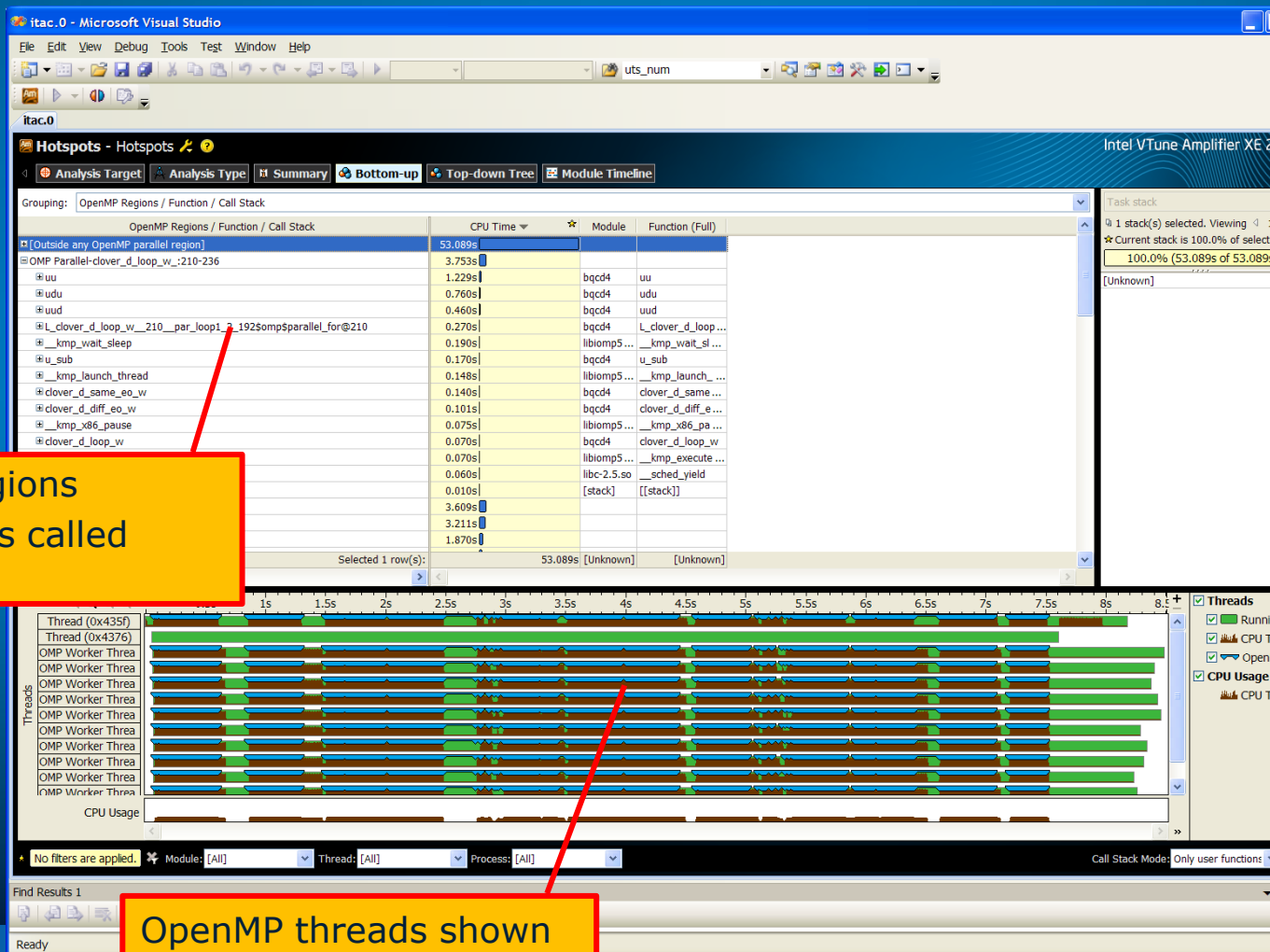## Visualize results in VTune™ Amplifier XE

- Linux: start

$ amplxe-gui

- Open results

 r003hs.amplxe

Hybrid Analysis (hotspots)
Hybrid program: 2 MPI processes + 12 Threads per process
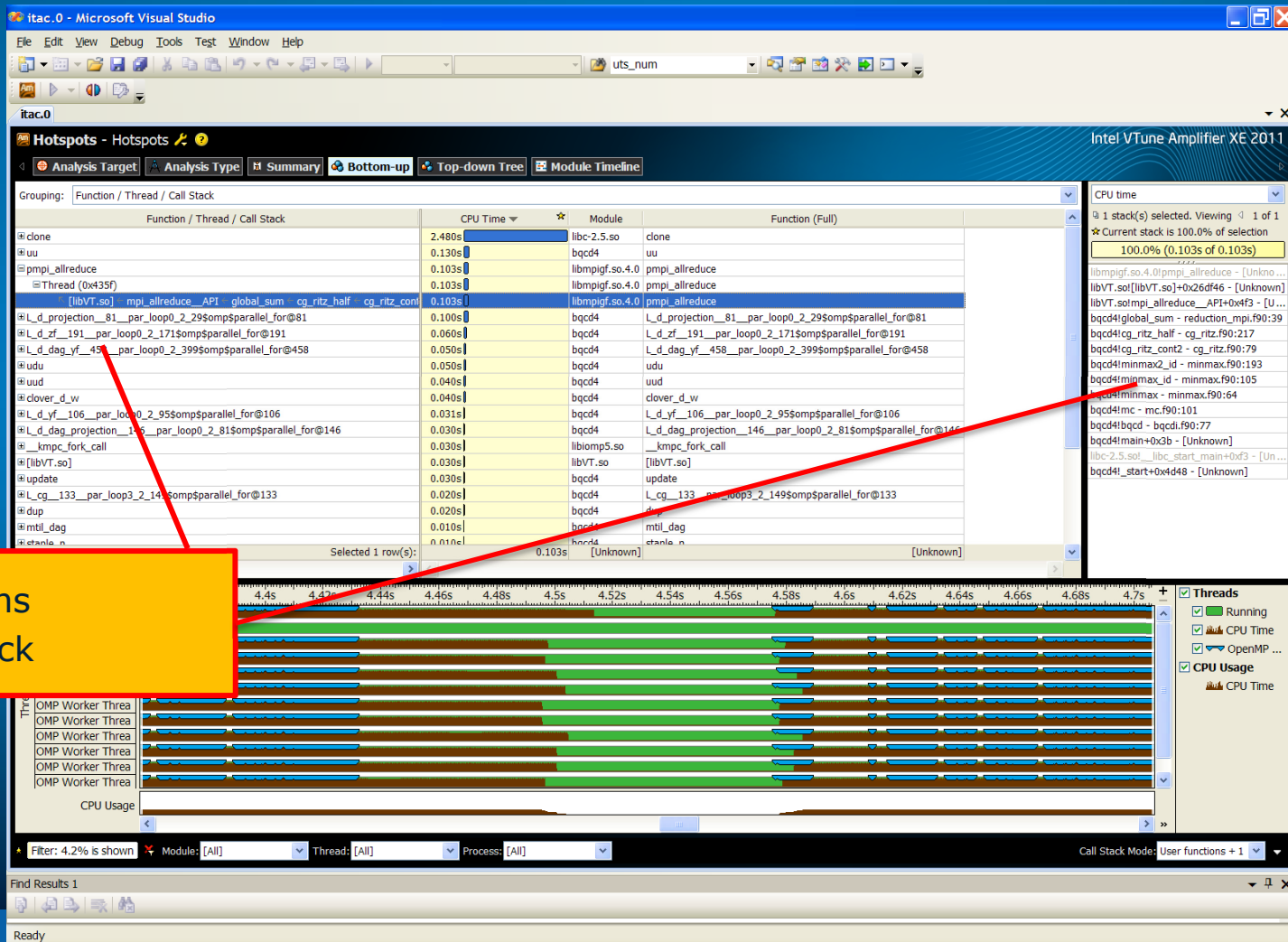

OpenMP Regions with routines called inside

OpenMP threads shown together with dapl (MPI) thread

MPI Analysis (hotspots)
Hybrid program: 2 MPI processes + 12 Threads per process

MPI Functions with call stack

Summary

- VTune™ Amplifier XE is the coming together of distinct tools
  - Correlation of the data from various capabilities makes it very powerful
  - Supports EBS data collection with better usability
  - Supports statistical call-graph capability
  - Combines these two types of capabilities with powerful Thread Profiling capability
- VTune™ Amplifier XE provides a standalone GUI with the same look and feel on both Windows* and Linux
- Complete re-design to make it a very extensible tool

Software & Services Group, Developer Products Division

(intel)

Software

Software & Services Group, Developer Products Division

Optimization Notice

Optimization Notice

Intel compilers, associated libraries and associated development tools may include or utilize options that optimize for instruction sets that are available in both Intel and non-Intel microprocessors (for example SIMD instruction sets), but do not optimize equally for non-Intel microprocessors. In addition, certain compiler options for Intel compilers, including some that are not specific to Intel micro-architecture, are reserved for Intel microprocessors. For a detailed description of Intel compiler options, including the instruction sets and specific microprocessors they implicate, please refer to the "Intel Compiler User and Reference Guides" under "Compiler Options." Many library routines that are part of Intel compiler products are more highly optimized for Intel microprocessors than for other microprocessors. While the compilers and libraries in Intel compiler products offer optimizations for both Intel and Intel-compatible microprocessors, depending on the options you select, your code and other factors, you likely will get extra performance on Intel microprocessors.

Intel compilers, associated libraries and associated development tools 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 Intel® Streaming SIMD Extensions 2 (Intel® SSE2), Intel® Streaming SIMD Extensions 3 (Intel® SSE3), and Supplemental Streaming SIMD Extensions 3 (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.

While Intel believes our compilers and libraries are excellent choices to assist in obtaining the best performance on Intel and non-Intel microprocessors, Intel recommends that you evaluate other compilers and libraries to determine which best meet your requirements. We hope to win your business by striving to offer the best performance of any compiler or library; please let us know if you find we do not.

Notice revision #20110307

Legal Disclaimer

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 www.intel.com/software/products.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright ©2010. Intel Corporation.

http://www.intel.com/software/products