Using Intel® VTune™ Amplifier XE for High Performance Computing

Levent Akyil, Heinrich Bockhorst, Gergana Slavova, Alexander Supalov

Intel Corporation

September 27, 2011
Agenda

• Brief overview of Intel software tools targeting parallelism
• Quick introduction to the new VTune™ Amplifier XE
• Using VTune™ Amplifier XE for hybrid parallel analysis
Common Tools and Programming Models

**Multicore**

Intel® Xeon® processors are designed for intelligent performance and smart energy efficiency.

Continuing to advance Intel® Xeon® processor family and instruction set (e.g., Intel® AVX, etc.)

**Code**

Use One Software Architecture

**Many-core**

Intel® MIC Architecture - co-processors are ideal for highly parallel computing applications.

Software development platforms ramping now

Use One Software Architecture Today. Scale Forward Tomorrow.
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, Intel® ArBB, and Intel® IPP
Intel® Inspector XE, Intel® VTune™ Amplifier XE, Intel® Advisors
Intel® MPI Library
Intel® Trace Analyzer and Collector
Intel® Parallel Studio XE

Performance. Scale Forward. Proven
Agenda

• Brief overview of Intel software tools targeting parallelism
• Quick introduction to the new VTune™ Amplifier XE
• Using VTune™ Amplifier XE for hybrid parallel analysis
**Intel® VTune™ Amplifier XE Evolution**

**Tune**
Analyze and optimize performance issues

**VTune™ Amplifier XE**
Linux OS & Windows OS* GUI/CLI support

*Other brands and names are the property of their respective owners.
Intel® VTune™ Amplifier XE
Quick Overview

• Fast, Accurate Performance Profiles
  – Hotspot (Statistical call tree)
  – Hardware-Event Based Sampling (EBS)

• Thread Profiling
  – Visualize thread interactions on timeline
  – Balance workloads

• Easy set-up
  – Pre-defined performance profiles
  – Use a normal production build

• Compatible
  – Microsoft*, GCC*, Intel compilers
  – C/C++, Fortran, Assembly, C#, .NET
  – Latest Intel® Architecture Processors and compatible processors

• Windows OS* or Linux OS*
  – Visual Studio* integration (Windows)
  – Standalone user i/f and command line
  – 32 and 64-bit
Application Level Analysis: Hotspots

Quickly identify what is important

Hottest Call Stack

Hottest Functions
Application Level Analysis
Concurrency Analysis

Frame is a time step or iteration

Thread active
Thread waiting
Thread transitions
Fast
Good
Slow
Frames / iterations

Frame Domain / Frame Type / Frames / Thread / Function / Call Stack

Selected 1 row(s):

CPU Time by Utilization

Overhead Time

Wait Time by Utilization

Frame Time

Threads

mainCRTStartup (0x1d2)
renderWork (0x1d50)
aiWork (0x1898)
particlesWork (0x180c)
networkThread (0x2158)
Thread (0x2320)
TBB Worker Thread (0x...)

CPU Usage

Thread Concurrency

Frame Rate
Hardware-Event Based Sampling (EBS)
EBS Made Easier

System Wide Event Based Sampling (EBS)
uses the on chip PMU to count performance events like cache misses, clock ticks and instructions retired.

Every Intel® Processor has an on chip Performance Monitoring Unit (PMU).

Predefined EBS Profiles
Easy EBS setup for newer processors. No memorizing complex event names. Profiles vary by microarchitecture. (Full custom profiles also available)

Opportunities Highlighted
General Exploration turns the cell pink when it suspects a tuning opportunity is present. Hover gives suggestions.

Pinpoint tuning opportunities
See opportunities like cache misses. View results on the timeline, in the grid view or on your source.
New in VTune™ Amplifier XE: Pre-Configured Profiles!

The Intel® Microarchitecture Codename Sandy Bridge: General Exploration profile should be used for a top-level analysis of potential issues. It is the subject of this guide.

All the events required are pre-configured – no research needed! Simply click Start to run the analysis.
The Old Way vs. The New Way

The Old Way: To see if there is an issue with branch misprediction, multiply event value (86,400,000) by 20 cycles, then divide by CPU_CLK_UNHALTED.THREAD (5,214,000,000). Then compare the resulting value to a threshold. If it is too high, investigate.

The New Way: Look at the Branch Mispredict metric, and see if any cells are pink. If so, investigate.
Agenda

• Brief overview of Intel software tools targeting parallelism
• Quick introduction to the new VTune™ Amplifier XE
• Using VTune™ Amplifier XE for hybrid parallel analysis
Nesting Multiple Levels of Parallelism

Distributed parallelism
- Explicit coordination through message-passing
- Example: Intel® MPI Library, Intel® Trace Analyzer and Collector (ITAC)

Thread-level parallelism
- Data parallelism and/or tasking
- Examples: OpenMP*, ITAC, VTune™ Amplifier XE

Instruction-level parallelism
- Examples: SIMD, VTune™ Amplifier XE

*Other brands and names are the property of their respective owners.
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
- Simplify and accelerate clusters
  - “Intel® Cluster Ready”
- Sustainable scalability beyond 60K cores
  - Native InfiniBand* interface support allows for lower latencies, higher bandwidth, and reduced memory requirements
- More robust MPI applications
  - Seamless interoperability with Intel® Trace Analyzer and Collector
**Intel® Trace Analyzer and Collector**

An analysis tool for MPI applications

- 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
  - Comparison of multiple profiles
  - Powerful aggregation and filtering functions
  - Fail-safe MPI tracing
  - Instrument user-level code via the API
  - Verify your code with MPI correctness checking and memory checking
  - Identify bottlenecks and application imbalances with the Idealizer
MPI Analysis
Hybrid program: 2 MPI processes + 12 Threads per process
Hybrid Analysis

• 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 and post-process the results
   - 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
Hybrid Analysis
Collect Performance/Correctness Data

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

• The list of analysis types available can be viewed using “amplxe-cl -help collect” command

• An example of full command line for hot spot data collection would be:

$ mpiexec -n 4 amplxe-cl -r my_result -collect hotspots -- my_app [my_app_ options]

• A number of result directories will be created in the current directory, named as my_result.0 – my_result.3
**Hybrid Analysis**

**Collect Performance/Correctness Data only on Selected Processes**

- 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:

  ```bash
  $ mpiexec -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)
Hybrid Analysis
Pre-view 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 inspxe-cl -help report or 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

• Windows: navigate to the result directory and double click on icon 📘
Hybrid Analysis (hotspots)
Hybrid program: 2 MPI processes + 12 Threads per process (1/2)

- OpenMP regions with routines called inside
- OpenMP threads shown together with MPI (dapl) service threads
MPI Analysis (hotspots)
Hybrid program: 2 MPI processes + 12 Threads per process (2/2)

MPI functions shown with the call stack
Summary

• **Intel® VTune™ Amplifier XE** is more powerful and intuitive
  – Simplifies algorithmic and micro-architectural performance analysis and tuning
• **Intel® VTune™ Amplifier XE Update 5** is available now
• **Upcoming updates**
  – Will be part of **Intel® Cluster Studio XE**
  – Will support installing on clusters
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
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.


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

Copyright © 2011. Intel Corporation.

http://intel.com/software/products