Performance Analysis for Intel architecture
Intel® VTune™ Amplifier XE 2013
Second Generation VTune™ Analyzer

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

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, .NET, Java
- Latest Intel® processors and compatible processors

Find Answers Fast
- Filter extraneous data
- View results on the source / assembly
- Event multiplexing

Windows or Linux
- Visual Studio Integration (Windows)
- Standalone user i/f and command line
- 32 and 64-bit

---

1 IA32 and Intel® 64 architectures. Many features work with compatible processors. Event based sampling requires a genuine Intel® Processor.

*Other brands and names are the property of their respective owners.
Hotspots analysis
Hotspots analysis – Source View
Advanced Processor Analysis – General Exploration

• Predefined Analysis Type that collects different types of CPU performance events
• Good for first look at whether any CPU event categories are affecting performance
• GUI highlights those events and functions that have performance problems
CPU HW Sampling results

Performance problems are highlighted

Hovering the mouse over a highlighted problem displays a tooltip with a problem definition and high level suggestions for fixes or analysis next steps
Locks-and-Waits View

Sync Object
- Function
- Call Stack

Wait Time
Wait Count
Module
Object Type
Object Creation

Mutex 0x33e2bb60
25.316s
492
[Unknown]
Mutex
[Unknown]

Idle
Poor
Ok

Mutex analysis showing
idle, poor, and ok
states.

Other brands and names are the property of their respective owners.
Locks-and-Waits Source View

Source:
```
162    drawing_area drawing(startx, totaly-y, stopx):
163    // Acquire mutex to protect pixel calculation
165    pthread_mutex_lock(&rgb_mutex);
166    for(int x = startx; x < stopx; x++) {
167        color_t c = render_one_pixel(x, y, local_r,
168        drawing.put_pixel(c);
```

Wait Time:
```
25.316s
```

Address:
```
0x323a  call 0x804ae00 <
0x323f  Block 5:
0x323f  add $0x14, %esp
0x3242  pushl $0x805d54c
0x3247  call 0x8049b10 <
0x324c  Block 6:
0x324c  movl $0x805d574,
0x3252  add $0x10, %esp
```

Highlight 2 rows:
Summary

- The Intel® VTune Amplifier XE can be used to find:
  - Source code for performance bottlenecks
  - Characterize the amount of parallelism in an application
  - Determine which synchronization locks or APIs are limiting the parallelism in an application
  - Easily find CPU performance events that are causing additional CPU clocks
## 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 (Intel 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 #20110228
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

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino Atom, Centrino Atom Inside, Centrino
Inside, Centrino logo, Cilk, Core Inside, FlashFile, i960, InstantIP, Intel, the Intel logo, Intel386,
Intel486, IntelDX2, IntelDX4, IntelSX2, Intel Atom, Intel Atom Inside, Intel Core, Intel Inside,
NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel
XScale, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium
Inside, skoool, Sound Mark, The Journey Inside, Viiv Inside, vPro Inside, VTune, Xeon, and Xeon
Inside 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 © 2013. Intel Corporation.

http://intel.com/software/products
Starting VTune™ Amplifier XE - The First Time

First create a project

Welcome to Intel VTune Amplifier XE 2013

- New Project...
- Open Project...

Recent Projects:
- Test
- Tachyon2013
- Java Mandelbroi
- Office 13
- Projects

Recent Results:
- 000hs
- 002hs
- 010hs
- 011hs
- 012hs
Specify optional app to launch

Indicate if you want to start an app.
Indicate type of profiling (ex: Lightweight Hotspots)

1. Click New analysis button
2. Select profiling type
3. Click “Start” to begin profiling
Running the General Exploration collector

1. Click “New Analysis” button

2. Select “General Exploration” for your CPU architecture

3. Click “Start” to begin profiling
AutoDetect DirectX® Frames

- Find occasional slow video frames
  - Identify causes of intermittently slow frames by comparing slow frame functions to fast frames
  - Definition of “slow” is user configurable

2.6% of DirectX frames were too slow

- Expand “Slow” and “Fast” nodes to see the differences and identify slow frame causes
“JIT” APIs

Profiling Runtime generated code

APIs to indicate attributes of code
• Code memory address
• Symbol information
  – Function names, Line Numbers

Drill down to source code when viewing profiling analysis

APIs are defined in jitprofiling.h
Locks and Waits Collection

Identifies those threading items that are causing the most thread block time

- Synchronization locks
- Threading APIs
- I/O
Running the “Locks and Waits” collector

1. Click “New Analysis” button
2. Select “Locks and Waits”
3. Click “Start” to begin profiling