Intel® VTune™ Amplifier XE

Vladimir Tsymbal
Performance, Analysis and Threading Lab

Agenda

- Intel® VTune Amplifier XE Overview
  - Features
  - Data collectors
  - Analysis types
- Key Concepts
- Collecting performance data in cluster environment

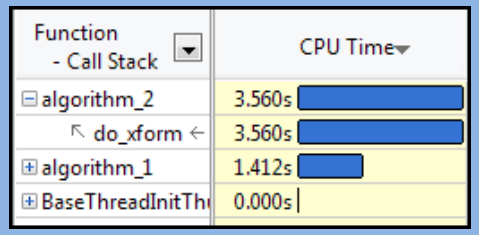
Intel® Cluster Studio XE New!

Phase	Product	Feature	Benefit
Build	 **Intel® MPI Library**	High Performance Message Passing (MPI) Library	• Enabling High Performance Scalability, Interconnect Independence, Runtime Fabric Selection, and Application Tuning Capability
	 **Intel® Composer XE**	C/C++ and Fortran compilers and performance libraries • Intel® Threading Building Blocks • Intel® Cilk™ Plus • Intel® Integrated Performance Primitives • Intel® Math Kernel Library	• Enabling solution to achieve the application performance and scalability benefits of multicore and forward scale to many-core
Verify	 **Intel® Inspector XE**	Memory & threading dynamic analysis for code quality Static Security Analysis for code quality	• Increased productivity, code quality, and lowers cost, finds memory, threading , and security defects before they happen • Now MPI enabled at every cluster node
Verify & Tune	 **Intel® Trace Analyzer & Collector**	MPI Performance Profiler for understanding application correctness & behavior	• Analyze performance of MPI programs and visualize parallel application behavior and communications patterns to identify hotspots
Tune	 **Intel® VTune™ Amplifier XE**	Performance Profiler for optimizing application performance and scalability	• Remove guesswork, saves time, makes it easier to find performance and scalability bottlenecks • Now MPI enabled at every cluster node

Intel® VTune™ Amplifier XE
Performance Profiler

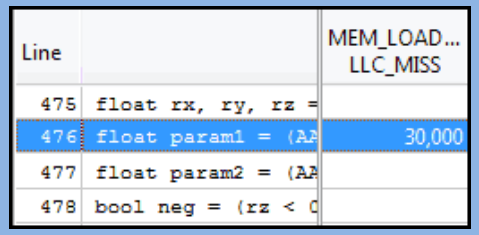
Where is my application…

Spending Time?



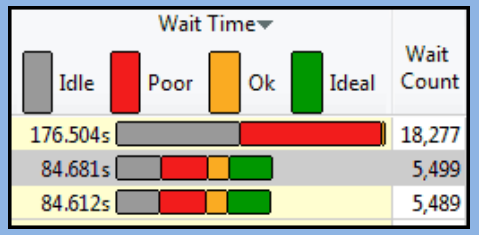
- Focus tuning on functions taking time
- See call stacks
- See time on source

Wasting Time?



- See cache misses on your source
- See functions sorted by # of cache misses

Waiting Too Long?



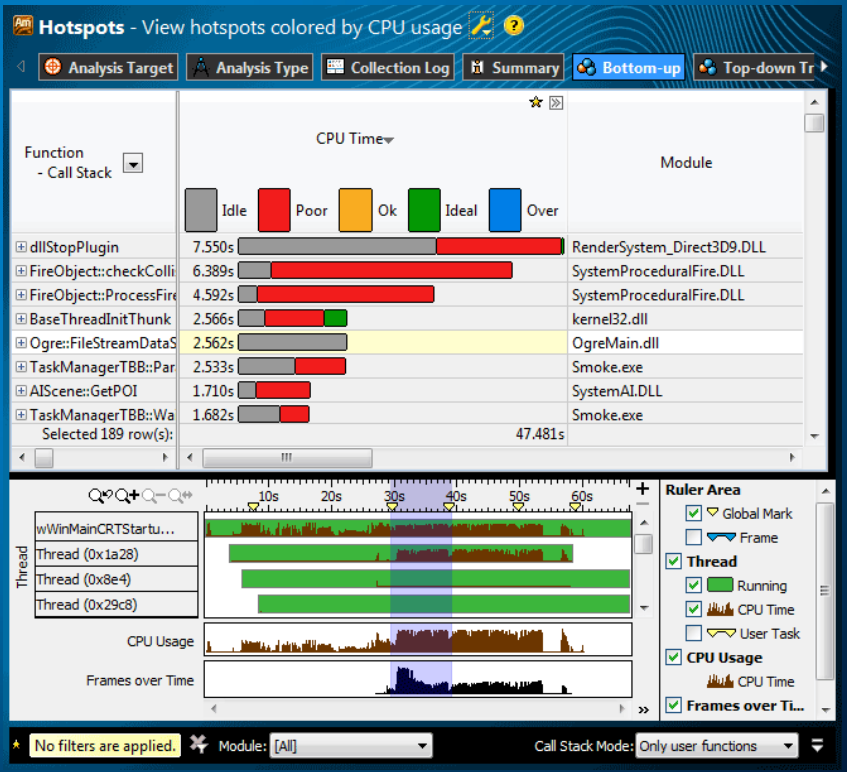
- See locks by wait time
- Red/Green for CPU utilization during wait

- Windows & Linux
- Low overhead
- No special recompiles

Advanced Profiling For Scalable Multicore Performance

Intel® VTune™ Amplifier XE
Tune Applications for Performance

- **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® processors and compatible processors[1]
- **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.

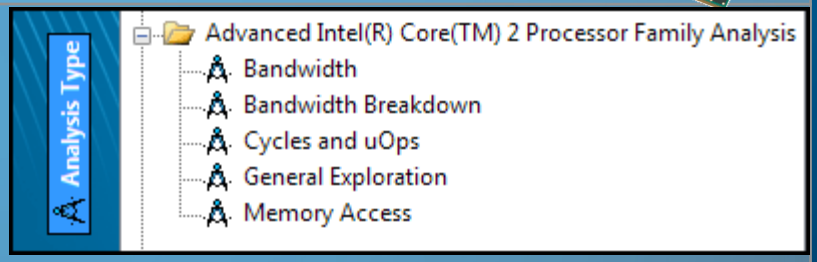
Intel® VTune™ Amplifier XE

Powerful EBS Made Easier

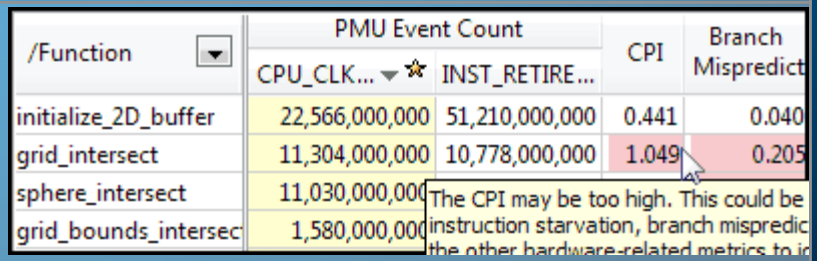
System Wide Event Based Sampling (EBS)
uses the on chip PMU to count performance events like cache misses, clock ticks and instructions retired.

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

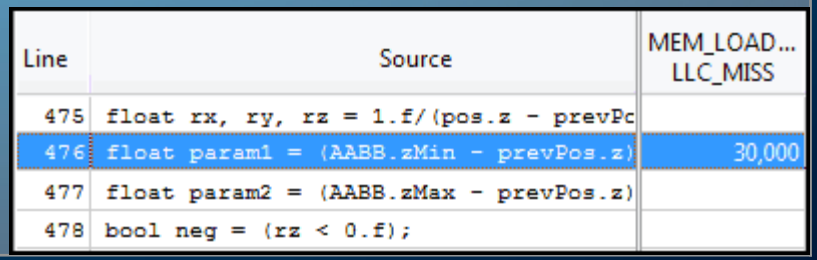
Predefined EBS Profiles
Easy EBS setup for newer processors.
No memorizing complex event names.
Profiles vary by microarchitecture.
(Full custom profiles also available)



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.



High-level Features

VTune™ Amplifier XE

High-level Features

- Hardware Event-based sampling (EBS)
  - Allows micro-architectural tuning
  - Improved usability
- Hotspot analysis (Software sampling)
  - Provides the time consuming regions of your application
  - Provides associated call-stacks that let you know how you got to these time consuming regions
  - Call-tree built using these call stacks
- Concurrency and Lock& Waits Analysis (Thread Profiling)
  - Visualize thread activity and lock transitions in the timeline
  - Provides lock profiling capability
  - Shows CPU/Core utilization and concurrency information

VTune™ Amplifier XE

High-level Features

- Attach to running processes
  - Hotspot and Concurrency analysis modes can attach to running processes
- System wide data collection
  - EBS modes allows system wide data collection and the tool provides the ability to filter this data
- GUI
  - Standalone GUI available on Windows* and Linux
  - Microsoft* Visual Studio integration
- Command line support
  - Comprehensive support for regression analysis and remote collection
- Platform & application support
  - Windows* and Linux
  - Microsoft* C# applications

VTune™ Amplifier XE

Improved data collection and visualization

- Timeline correlation of thread and event data
  - Populates thread active time with event data collected for that thread
  - Ability to filter regions on the timeline
- Advanced source and assembly views
  - See event data graphed in the source/assembly display
  - Visualize and analyze assembly as basic blocks
- Provides pre-defined tuning experiments

Data Collectors and Analysis Types

VTune™ Amplifier XE

Data Collectors

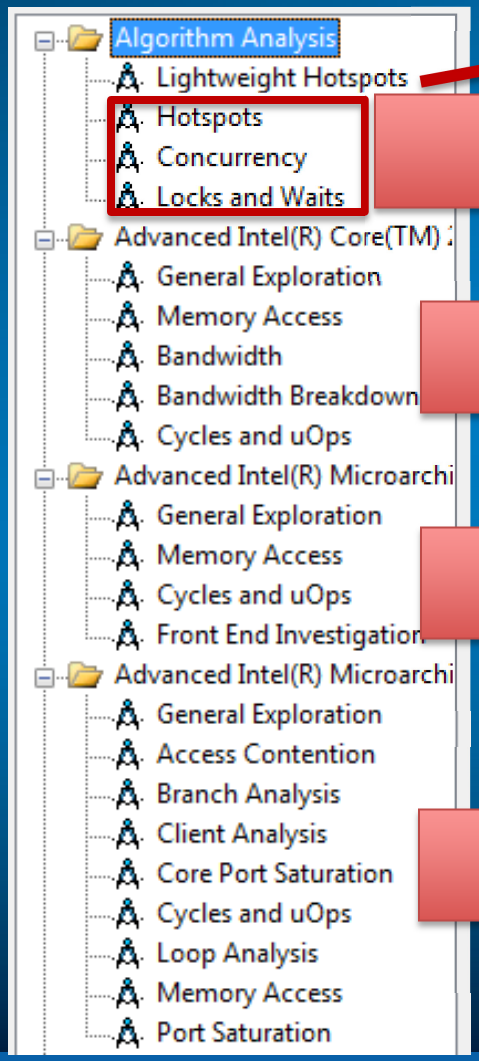
- Collectors come in two flavors
  - Event based sampling (EBS)
  - User mode sampling (Software sampling)
    - Thread profiling and stack sampling collector
    - Uses Pin based dynamic instrumentation technology
    - Application recompilation not necessary
    - Provides call-stacks with each sample
    - Statistical Call-tree constructed from call stacks

VTune™ Amplifier XE

Analysis types

- Software sampling based analysis types:
  - Hotspots
  - Concurrency
  - Locks and Waits
- Hardware EBS based analysis types:
  - Lightweight hotspots (pre-defined)
  - Advanced Hardware-level analysis (pre-defined)
    - Intel® Core™ i7 processor family
    - Intel® Xeon™ processor family
- Supports custom analysis types
  - Can be based on existing profiles

VTune™ Amplifier XE
Pre-defined Analysis Types



Lightweight Hotspot analysis based on the underlying architecture

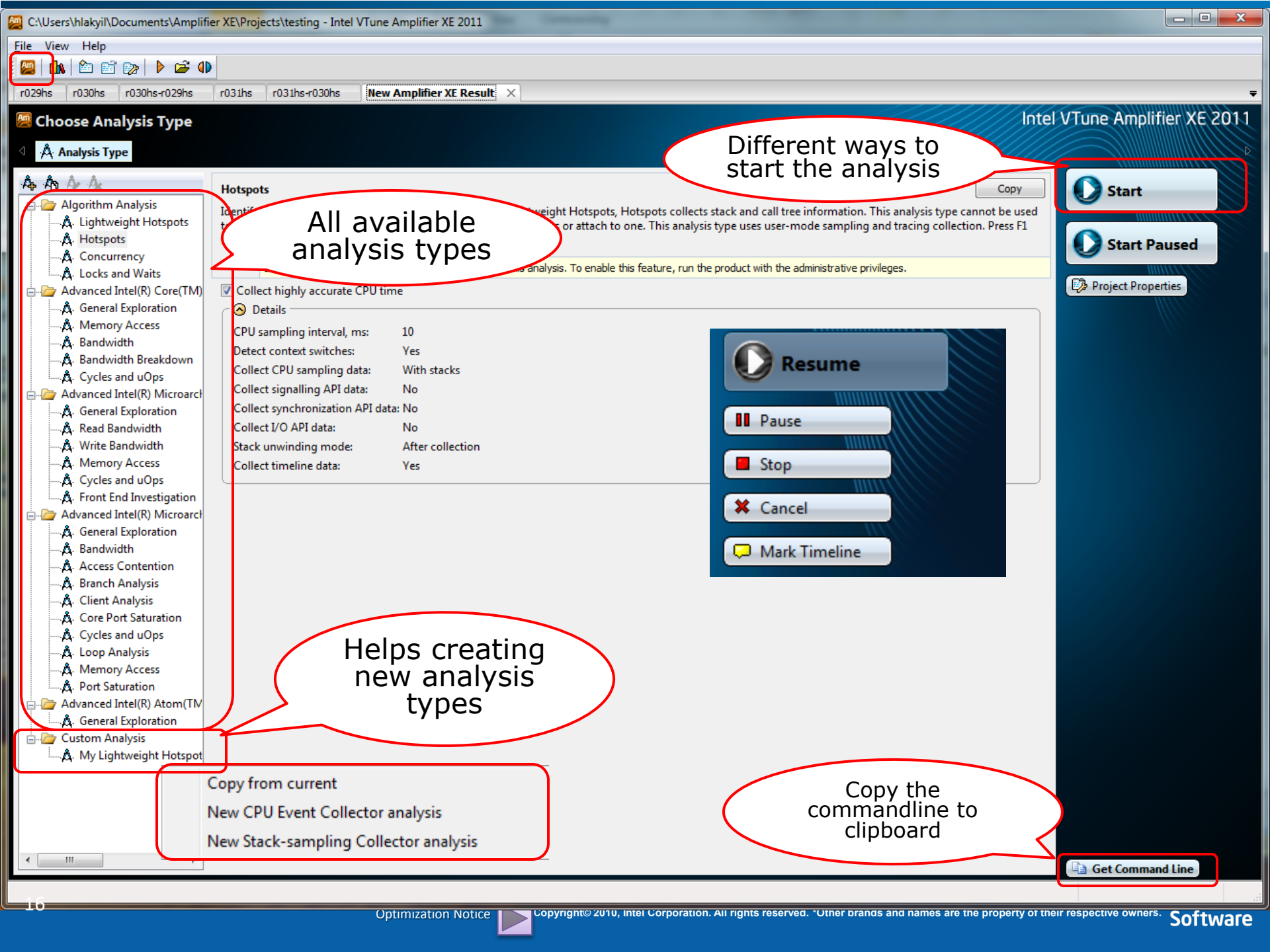
User mode stack sampling
Threading, IO, Signaling API instrumentation

Core™ 2 Architecture Analysis types

Core™ i7 (a.k.a Nehalem) Architecture analysis types

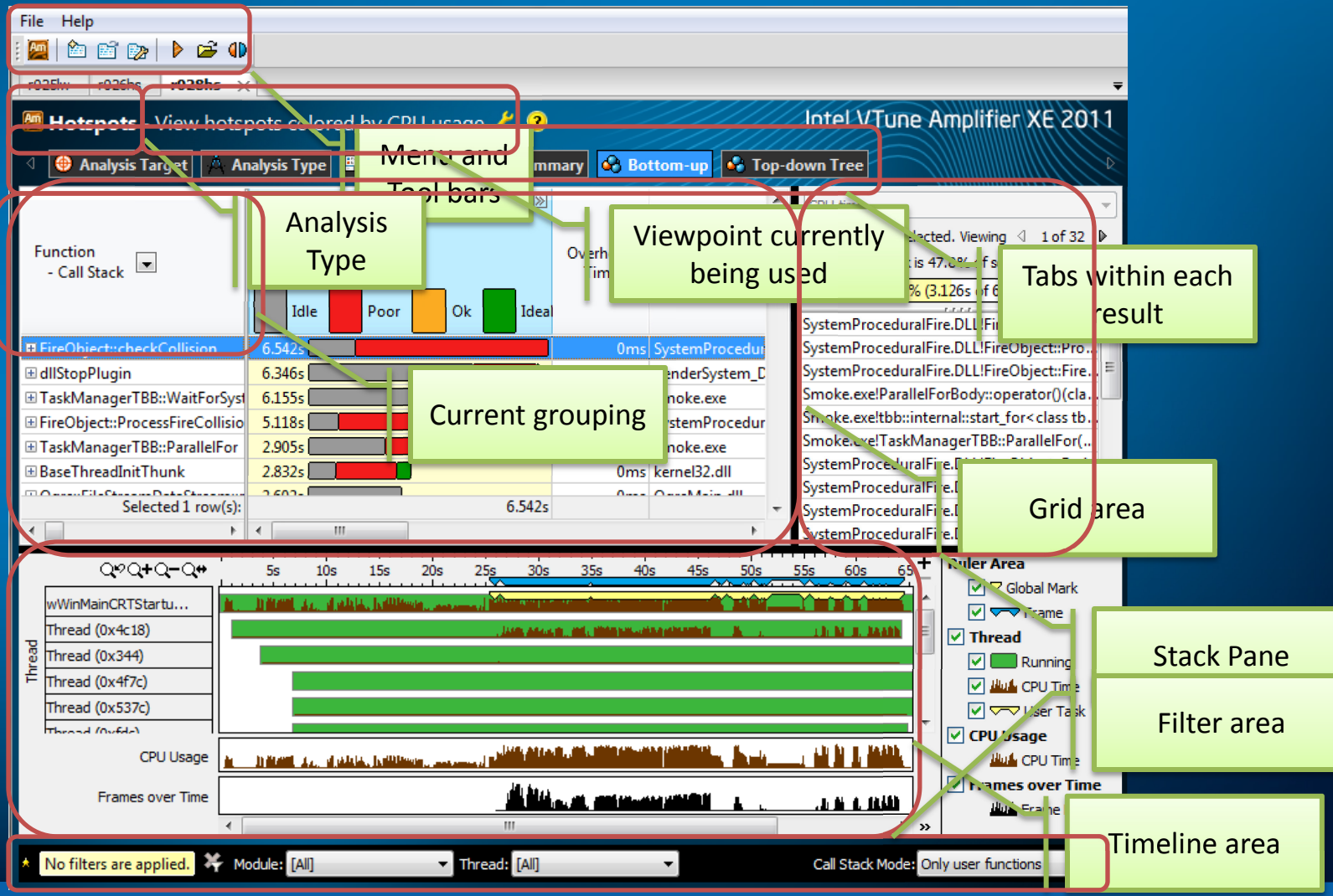
2nd Generation Core Architecture (a.k.a SandyBridge) analysis types

GUI Layout



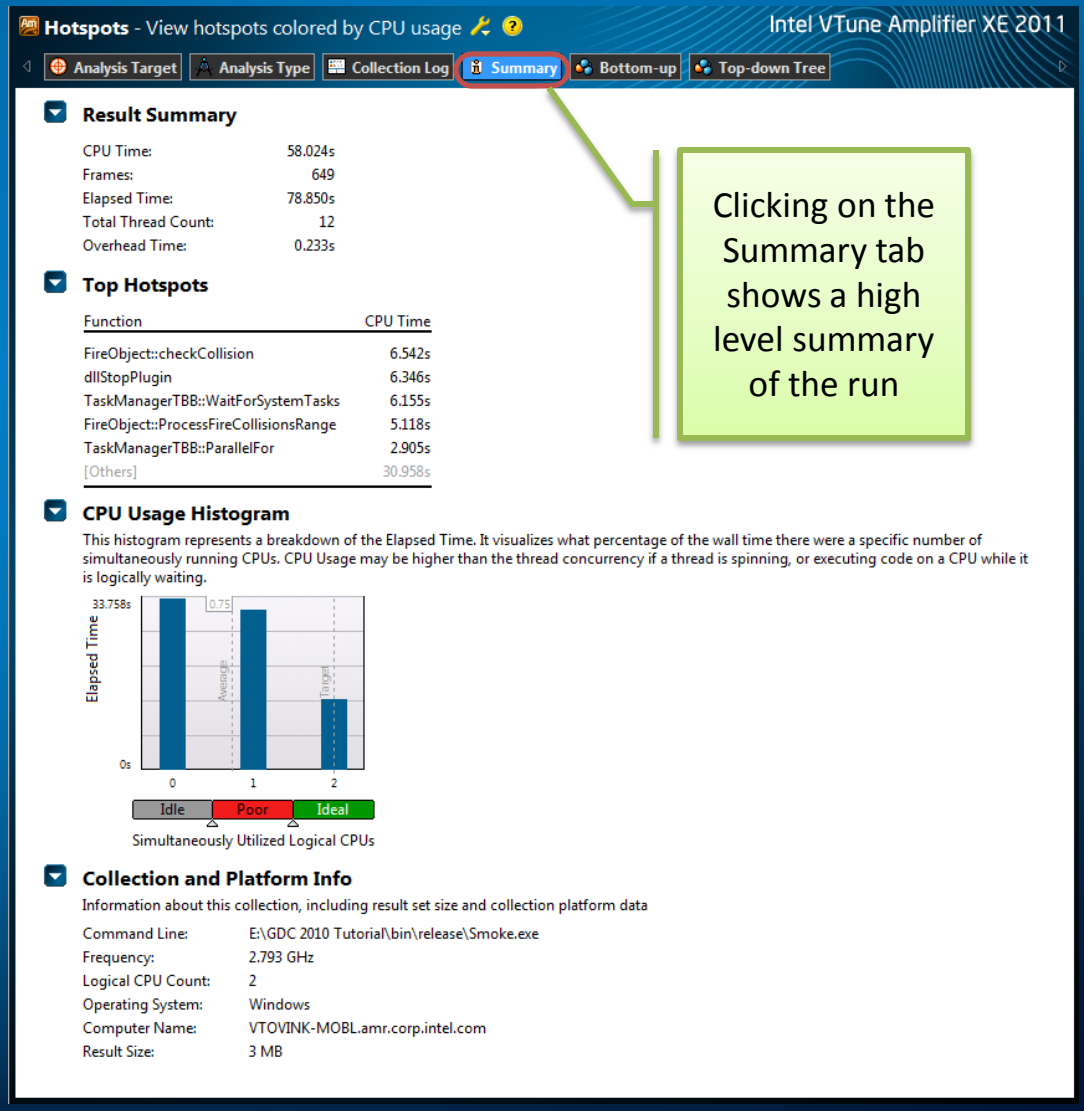
VTune™ Amplifier XE

GUI Layout



VTune™ Amplifier XE

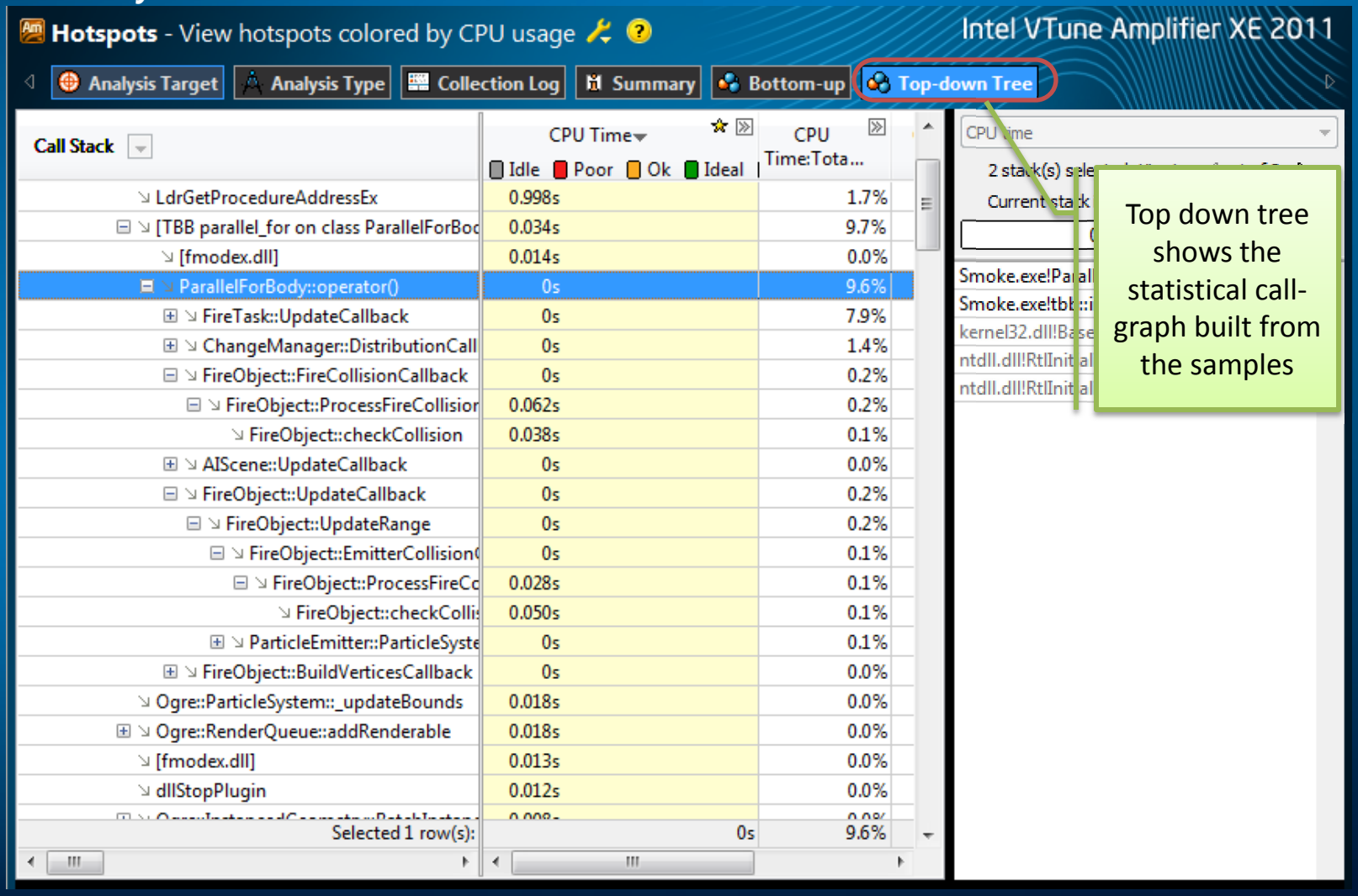
GUI Layout



Clicking on the Summary tab shows a high level summary of the run

VTune™ Amplifier XE

GUI Layout

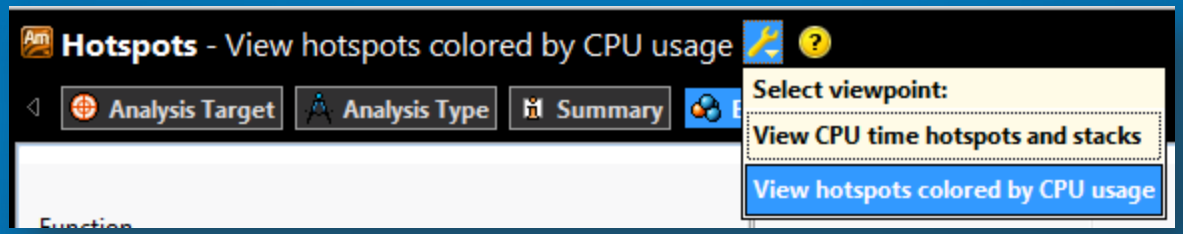


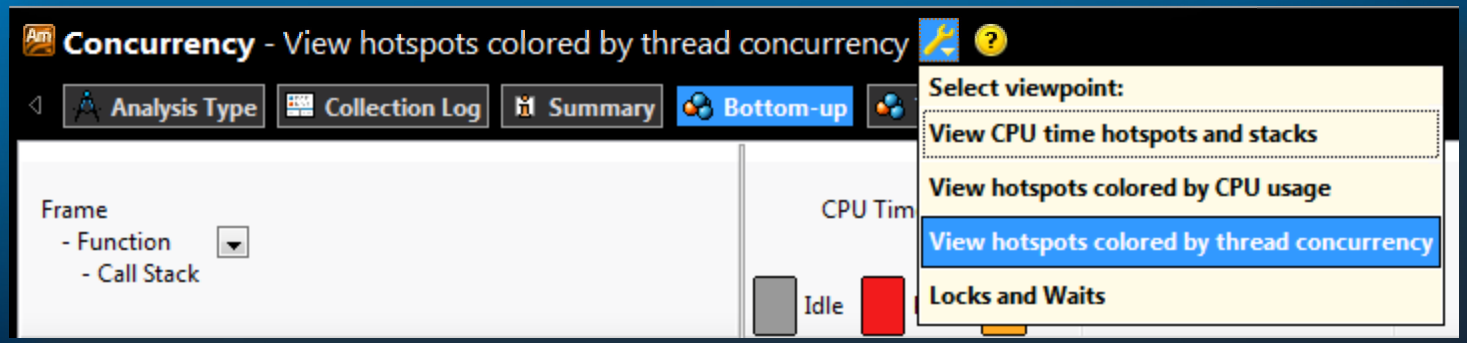
Key Result Analysis and GUI Concepts

VTune™ Amplifier XE

Key Concepts

- Viewpoints
  - Example of an analysis type having multiple viewpoints

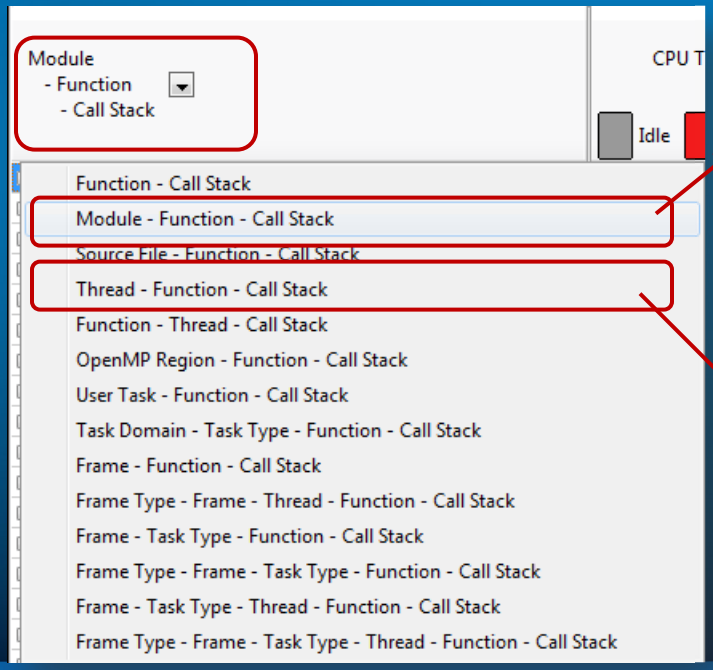


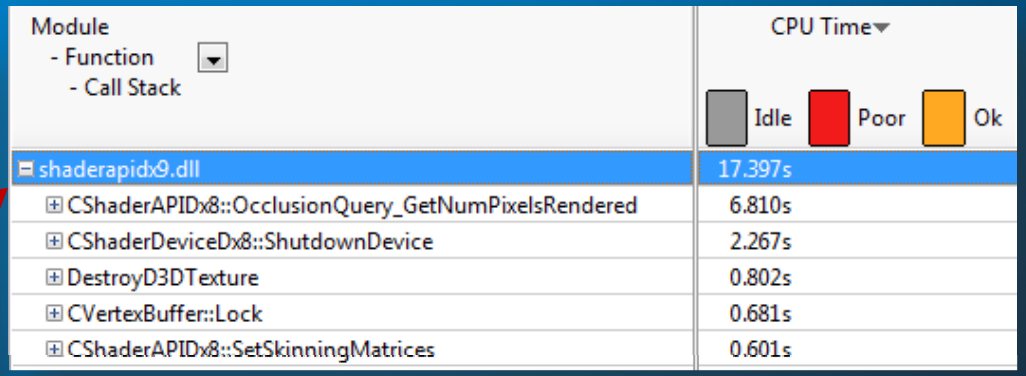


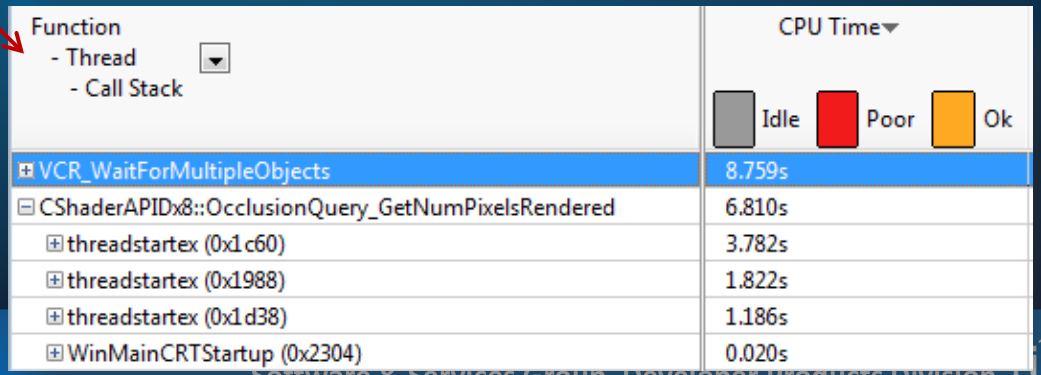
VTune™ Amplifier XE

Key Concepts

- Groupings
  - Each analysis type has many viewpoints
  - Each viewpoint has pre-defined groupings
  - Allows you to analyze the data in different hierarchies and granularities



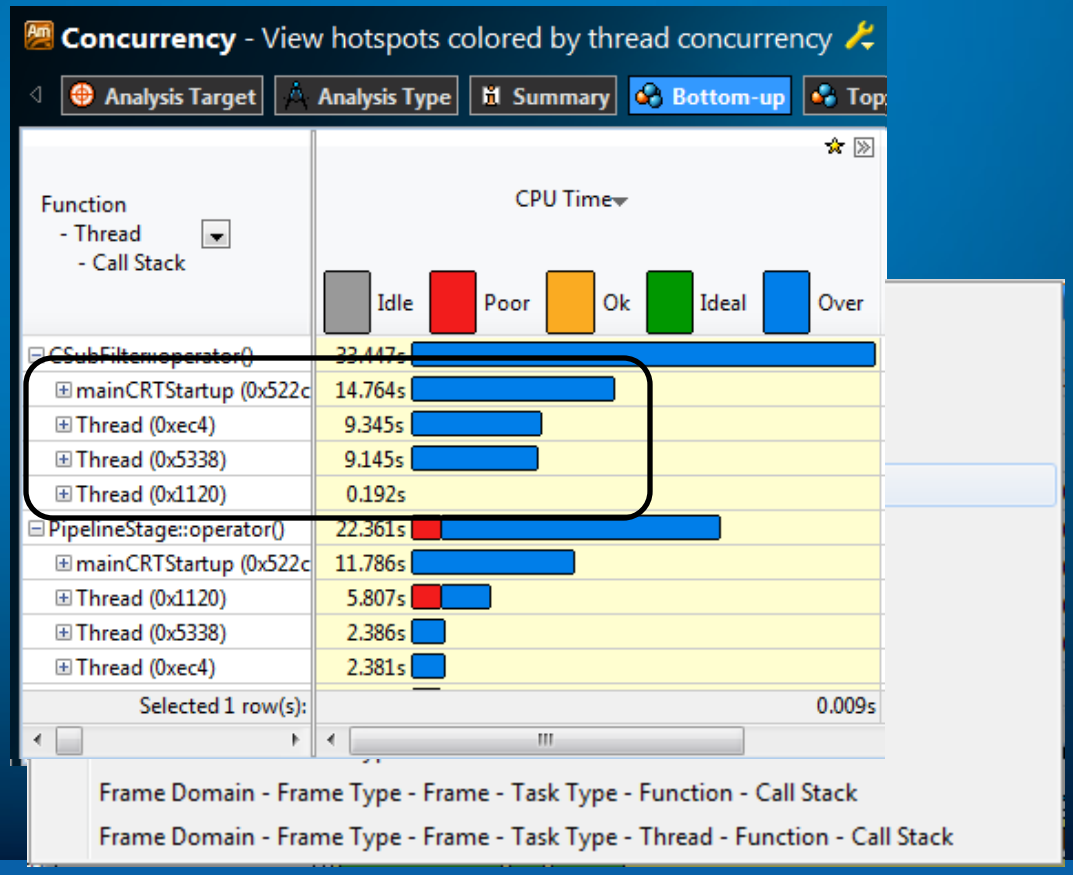




VTune™ Amplifier XE

Key Concepts

- For example, pre-defined groupings can be used to determine load imbalance



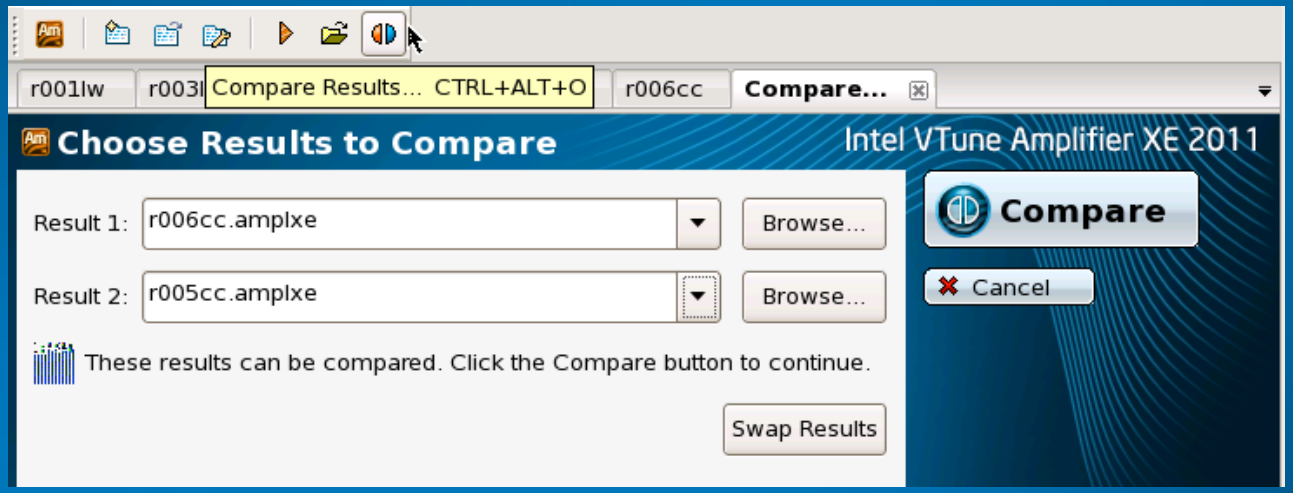
VTune™ Amplifier XE

Key Concepts

- VTune™ Amplifier XE allows comparison of two similar runs
- Extremely useful for
  - Benchmarking
  - Regression analysis
  - Testing

VTune™ Amplifier XE

Key Concepts



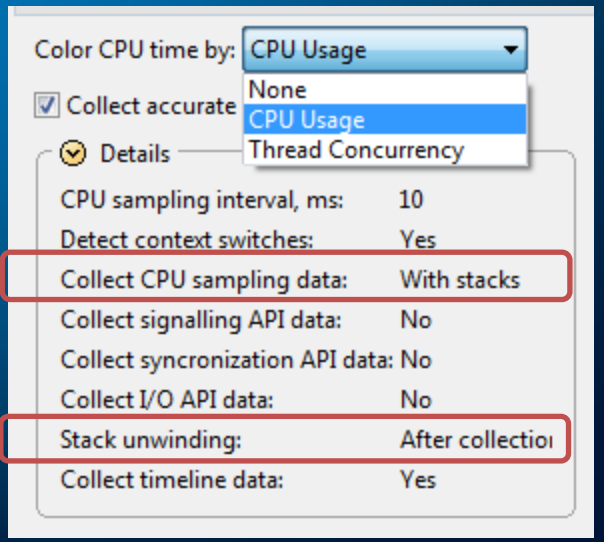
/Function	CPU Time:Difference by Utilization	CPU Time: r006cc by Utilization (Idle Poor Ok Ideal Over)	CPU Time: r005cc by Utilization (Idle Poor Ok Ideal Over)
grid_intersect	-3.122s	1.660s	4.782s
sphere_intersect	-2.388s	1.580s	3.968s
Raypnt	0s	0.060s	0.060s
pos2grid	-0.070s	0.050s	0.120s

Demo/Lab Activities
Analysis Types Revisited

Analysis Types

Hotspots

- For each sample, capture execution context, time passed since previous sample and thread CPU time
- Allows time spent in system calls to be attributed to user functions making the call
- Provides additional knobs:
  - Collect accurate CPU information requires “Administrator” privileges and uses ETW data to determine context switches
    - Timeline does not reflect CS
  - If “Thread Concurrency” is selected for coloring CPU time, then Pin Probes is used to capture wait data (slightly higher overhead)
  - The defaults for Hotspot analysis are configurable and can be done so by creating a custom analysis type inherited from Hotspots



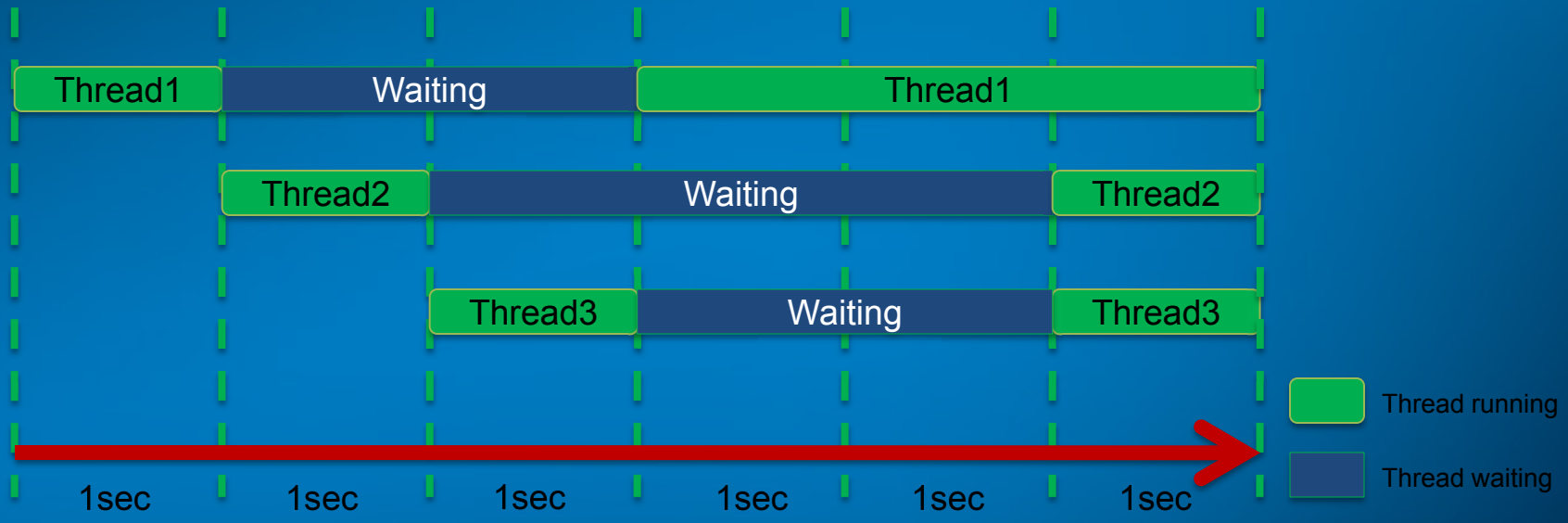
VTune™ Amplifier XE

Terminology used

- CPU Time
  - The amount of time a thread spends executing on a logical processor
  - For multi-threaded applications, the CPU time is aggregated over all threads for the given level of granularity
- Wait Time
  - The amount of time that a given thread waited for some event to occur, such as: synchronization waits and I/O waits

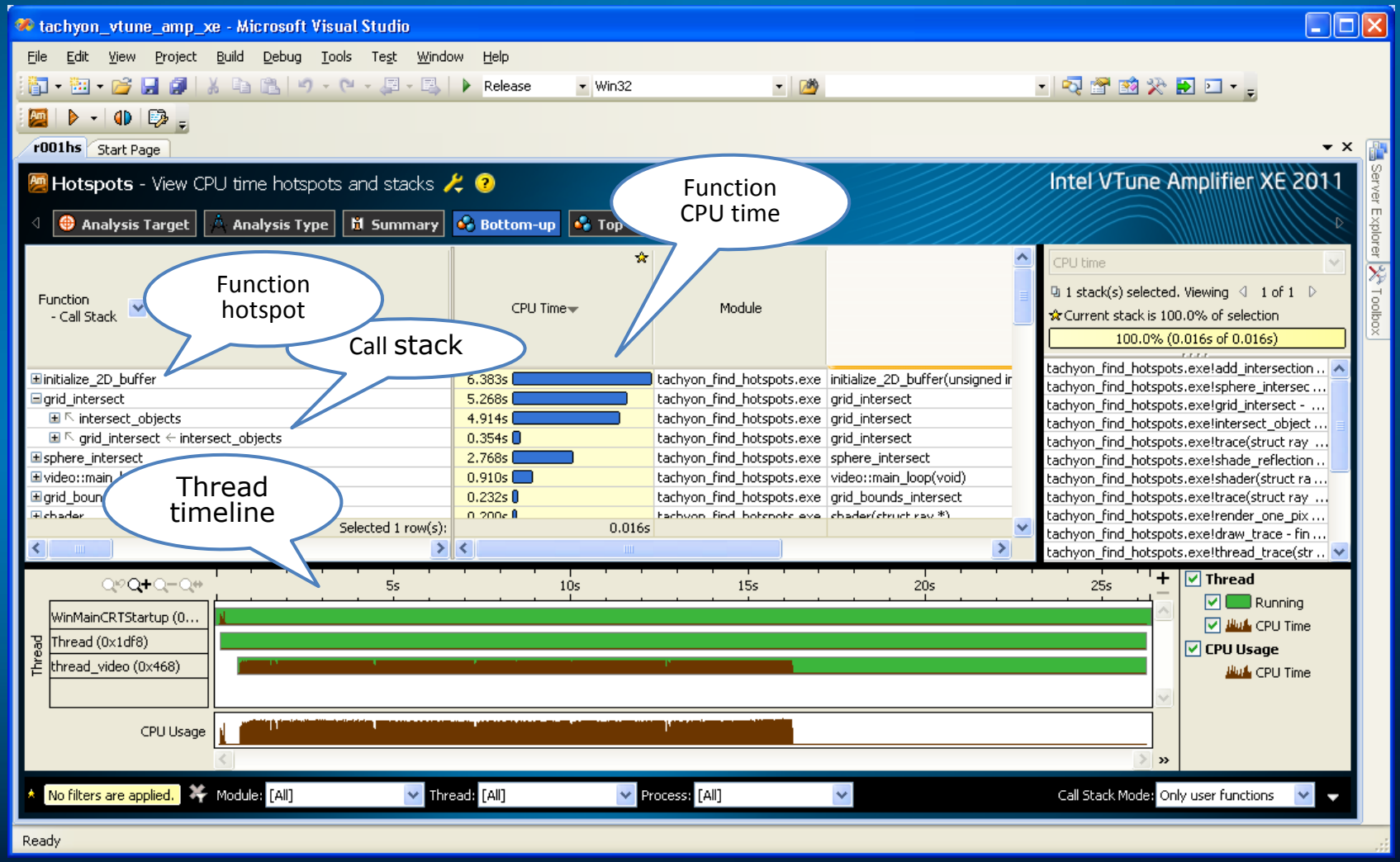
VTune™ Amplifier XE

Terminology used

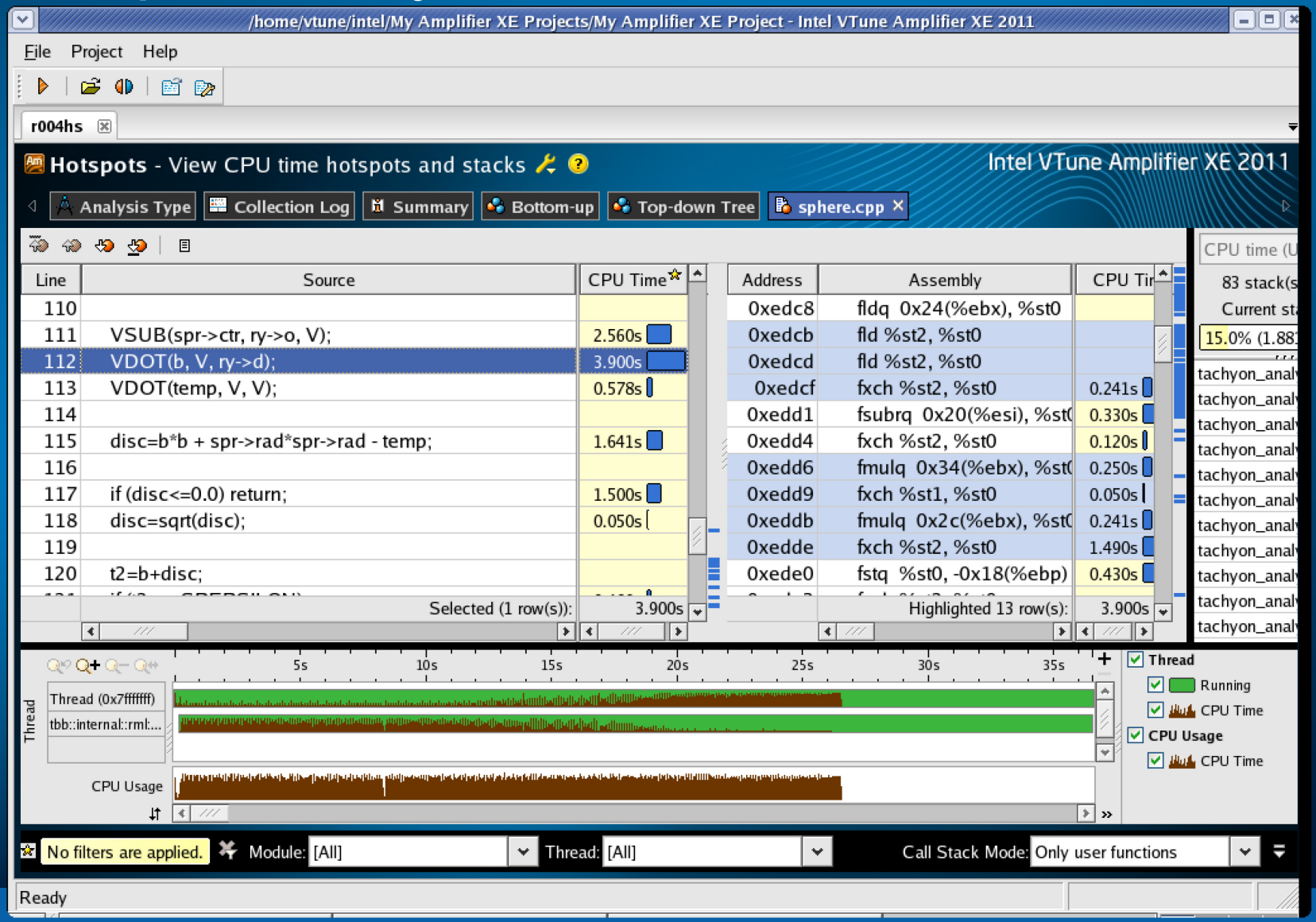


- **Elapsed Time**: 6 seconds
- **CPU Time**: T1 (4s) + T2 (2s) + T3 (2s) = 8 seconds
- **Wait Time**: T1(2s) + T2(3s) + T3 (2s) = 7 seconds

Hotspots analysis



Hotspots analysis – Source View



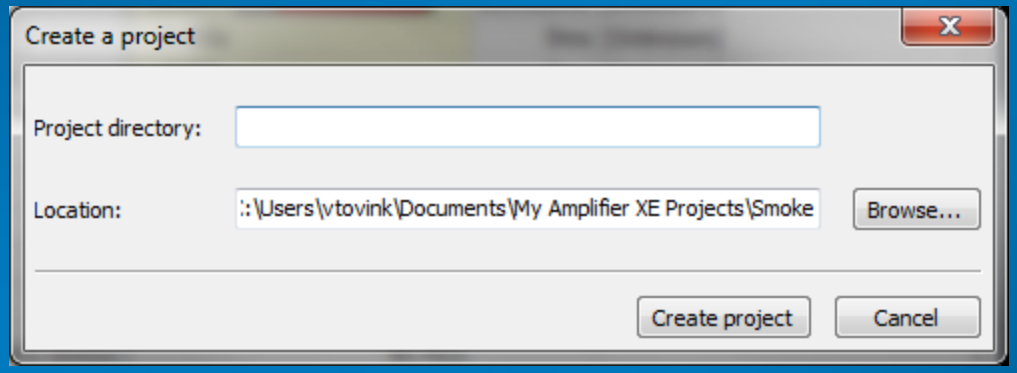
Before starting demo/lab activities let's quickly check how to use VTune™ Amplifier XE

Creating a New Project

- Project management support provided is minimal
  - Allows users to create new projects
  - Each project is defined by an executable of interest (or system wide for EBS)
  - Associated command line arguments that describe the workload, if applicable

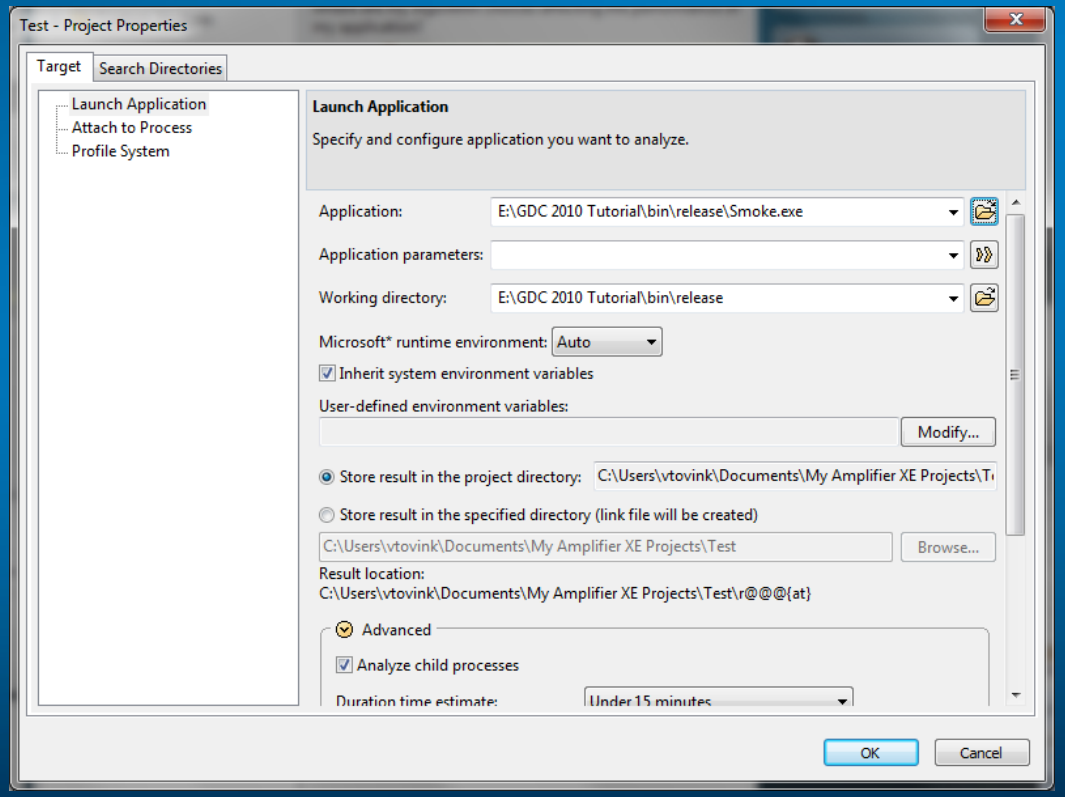
Steps for creating a new Project

- Select the “New Project” menu attribute



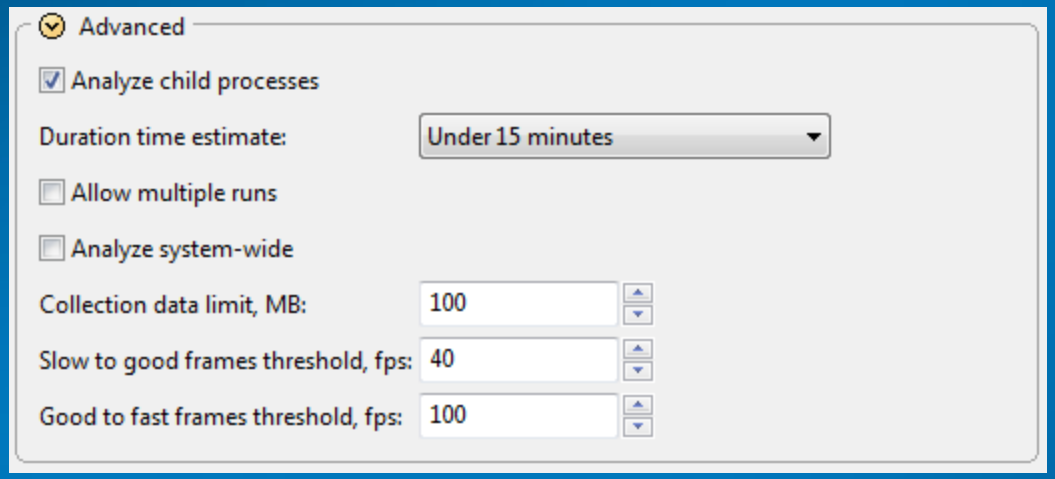
- This opens a dialog window for you to create a *new* project directory
  - Browse to the directory under which you choose to place the new project
  - Give a name for the project in the “Project Directory” field
  - Press the “Create Project” button

Associating an Application and Workload to the Project

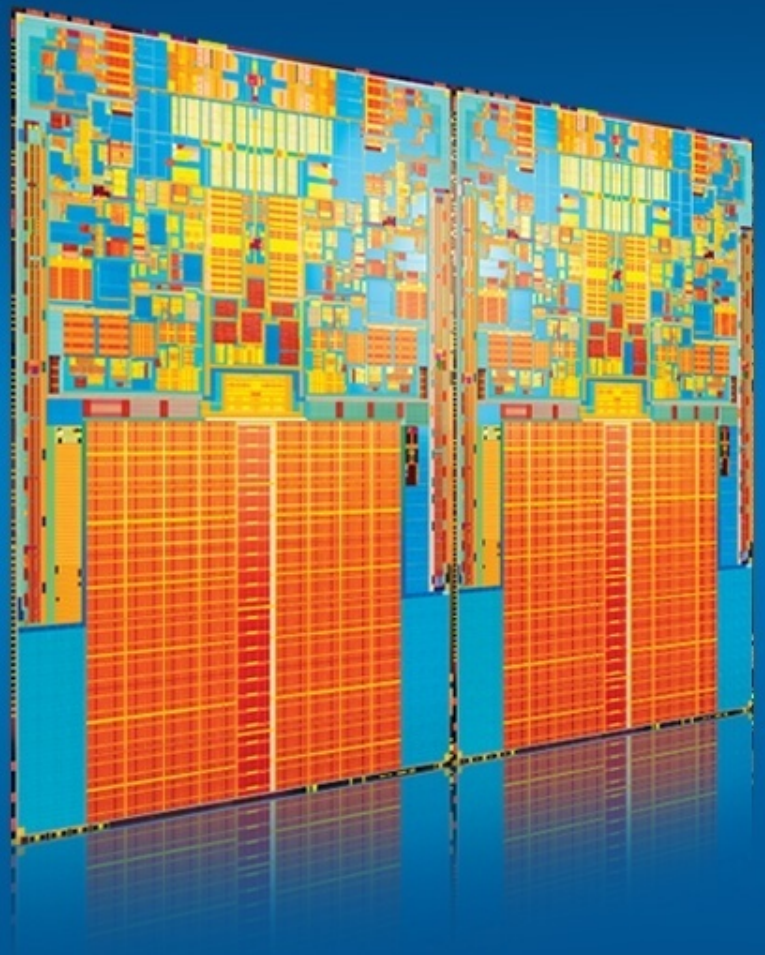


- Select the application
- Specify the command line arguments
- Select the working directory, if different from the parent directory of the application binary
- Select the “Ok” button

Advanced Project Configuration



- At the bottom of the “project Properties” dialog, other knobs are provided
  - User can specify if child process have to followed and analyzed
  - In the case of EBS, checking the “Allow multiple runs” will disable event multiplexing
  - There is also a data collection limit, which defaults to 100MB



Lab 1

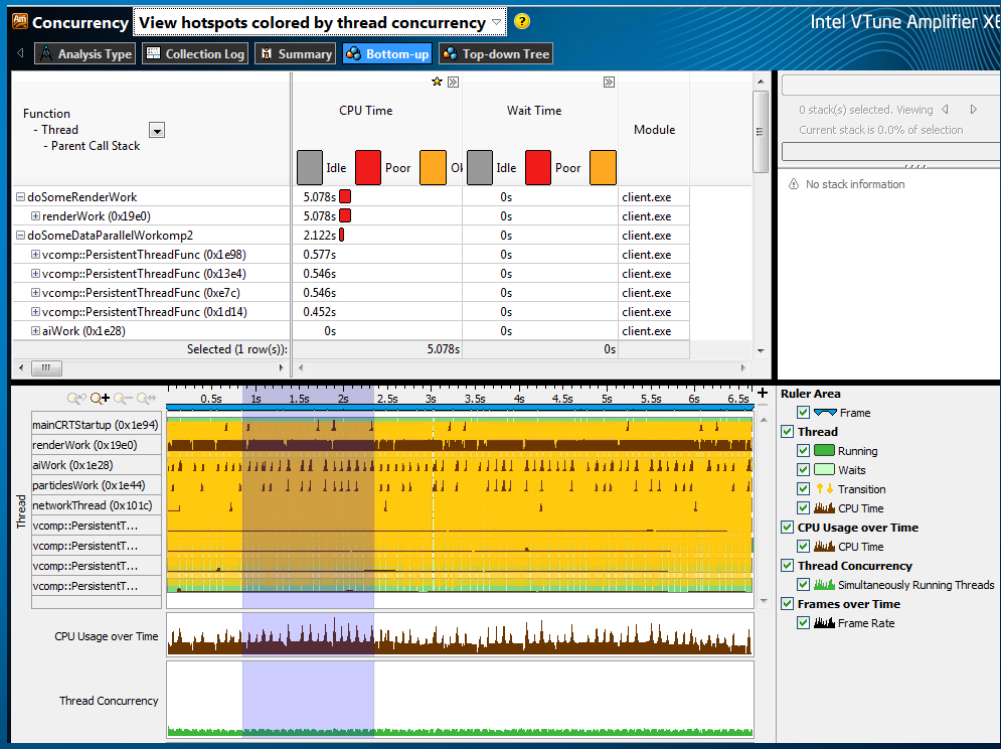
Finding Hotspots

intel Software

Intel VTune™ Amplifier XE

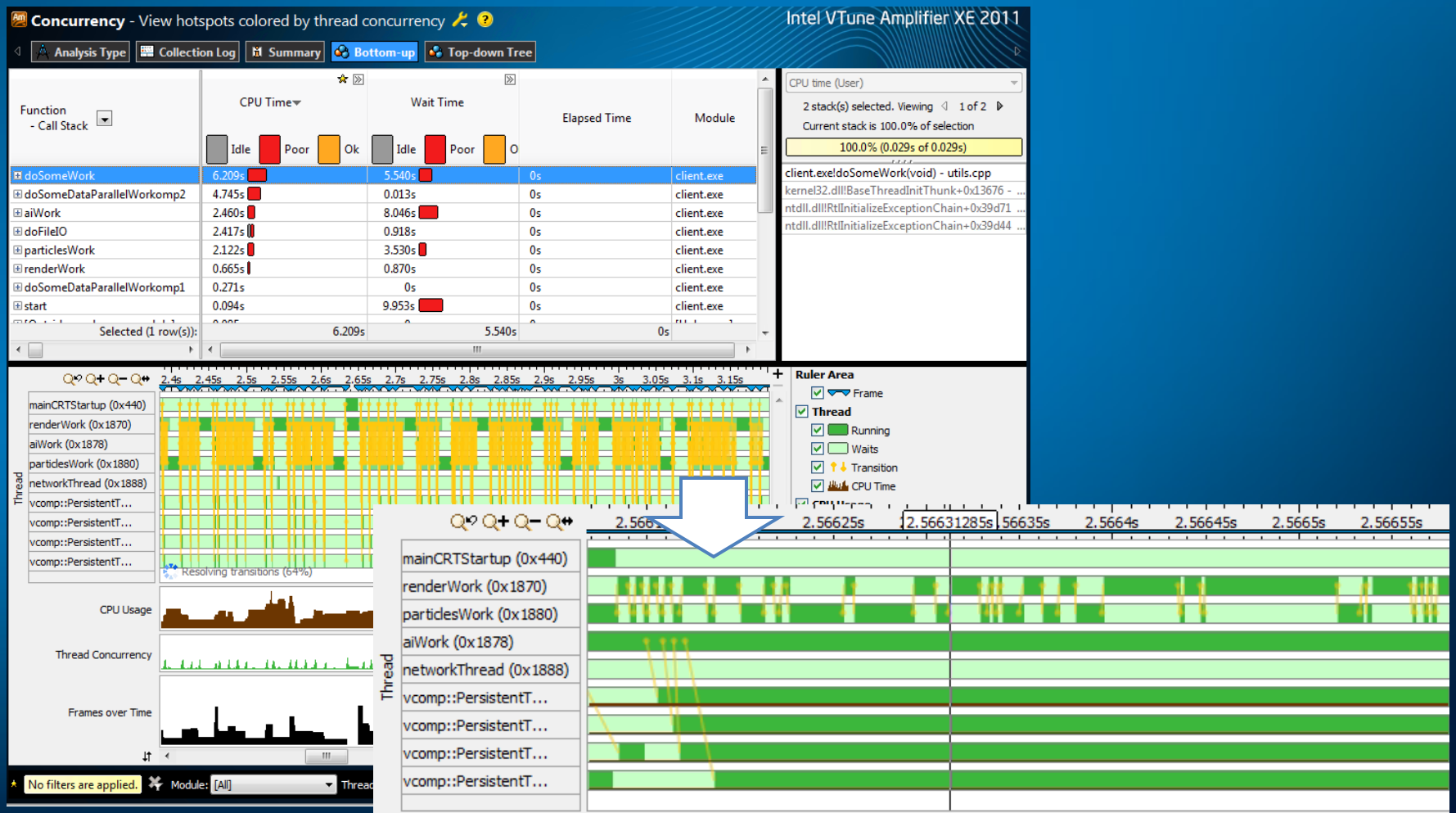
Concurrency Analysis

- For Concurrency analysis,
  - Stack sampling is done just like in Hotspots analysis type
  - Wait functions are instrumented (e.g. WaitForSingleObject, EnterCriticalSection)
  - Signal functions are instrumented (e.g. SetEvent, LeaveCriticalSection)
  - I/O functions are instrumented (e.g. ReadFile, socket)
  - Has the ability to also display utilization based on CPU Usage
  - Has newer metrics such as “overhead” and “spin” time



Intel VTune™ Amplifier XE

Parallelism/Concurrency Analysis



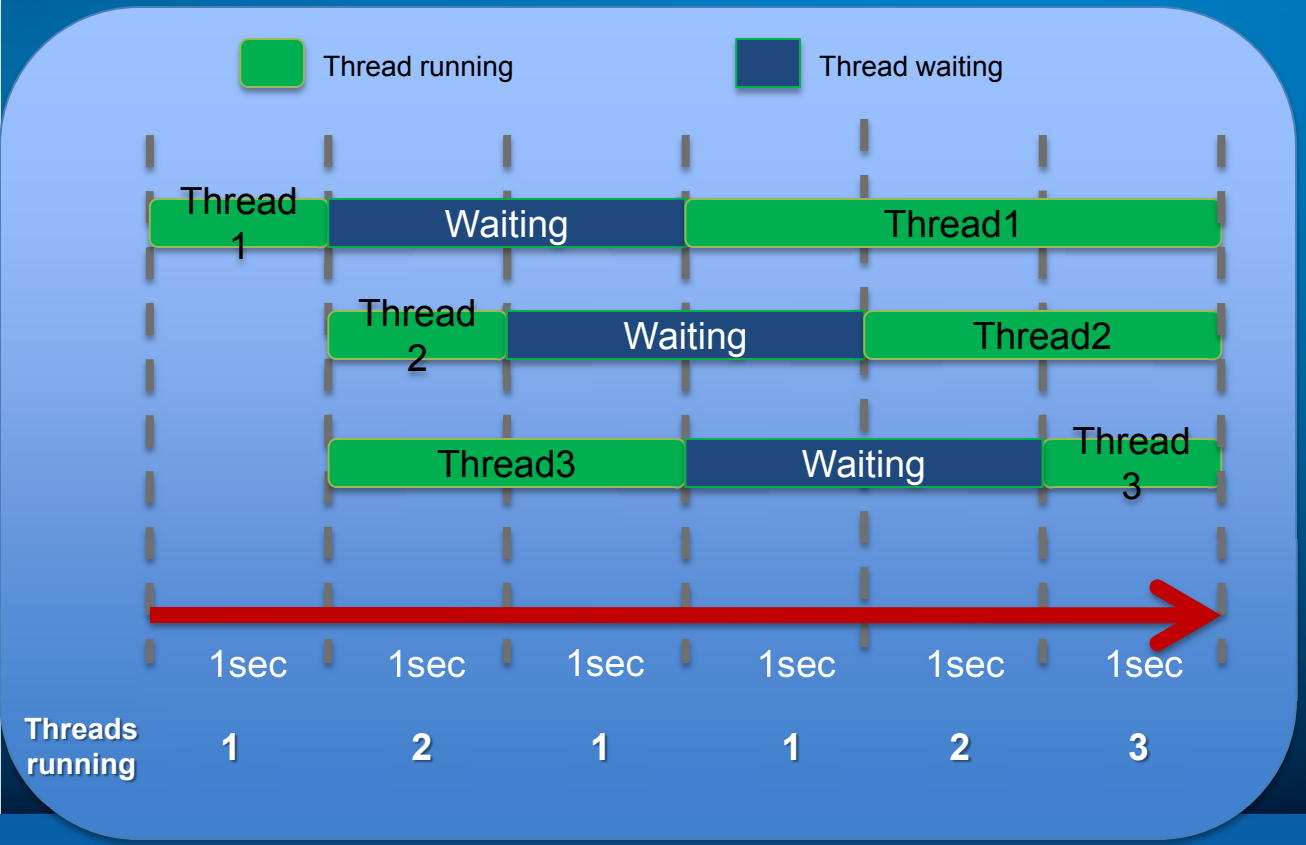
Software & Services Group, Developer Products Division

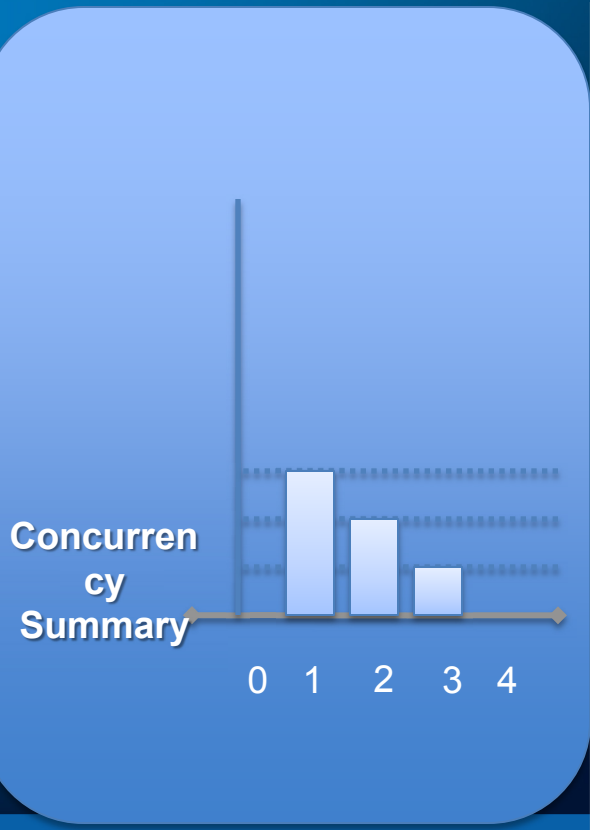
intel Software

VTune™ Amplifier XE

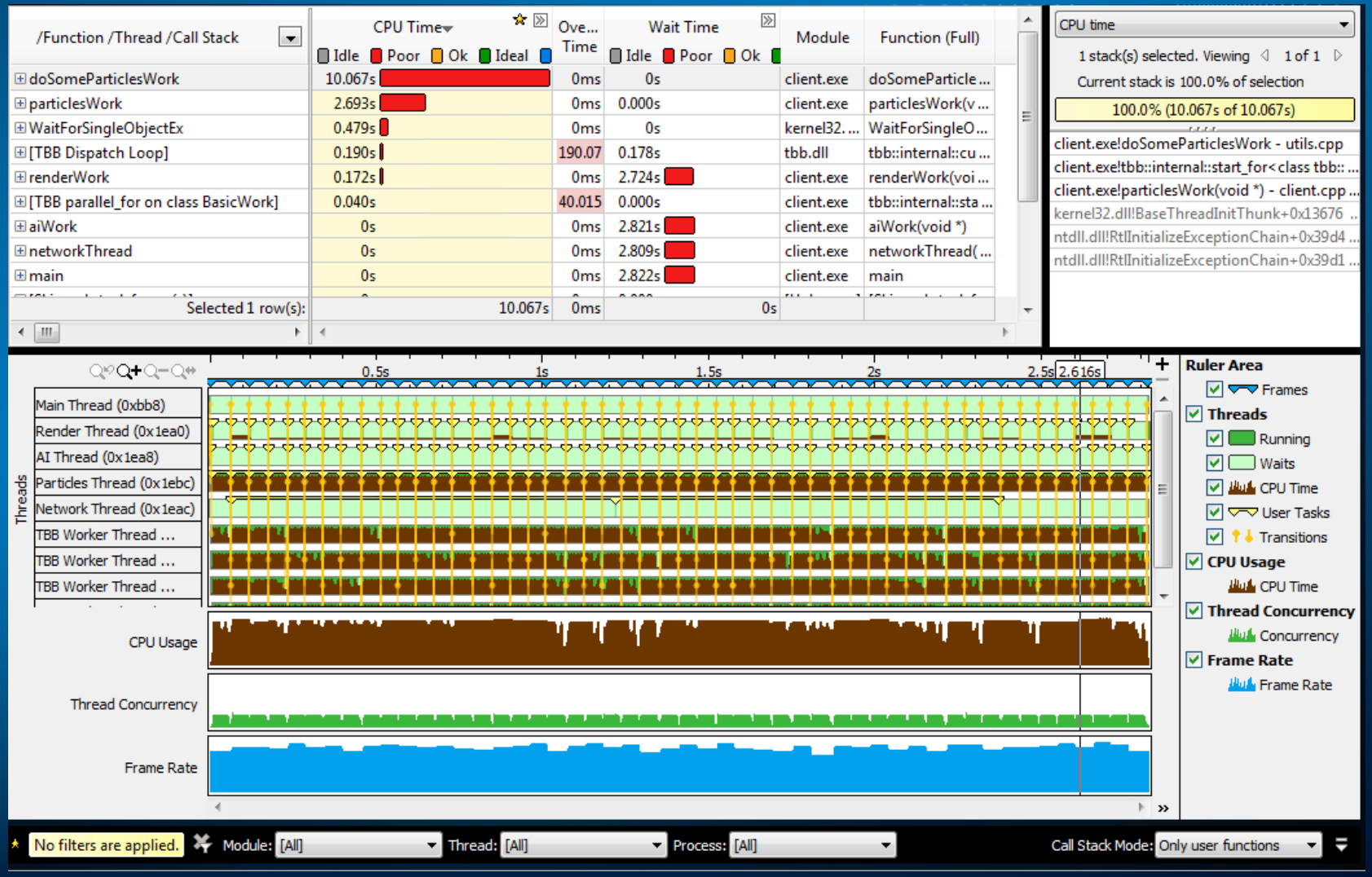
Terminology used

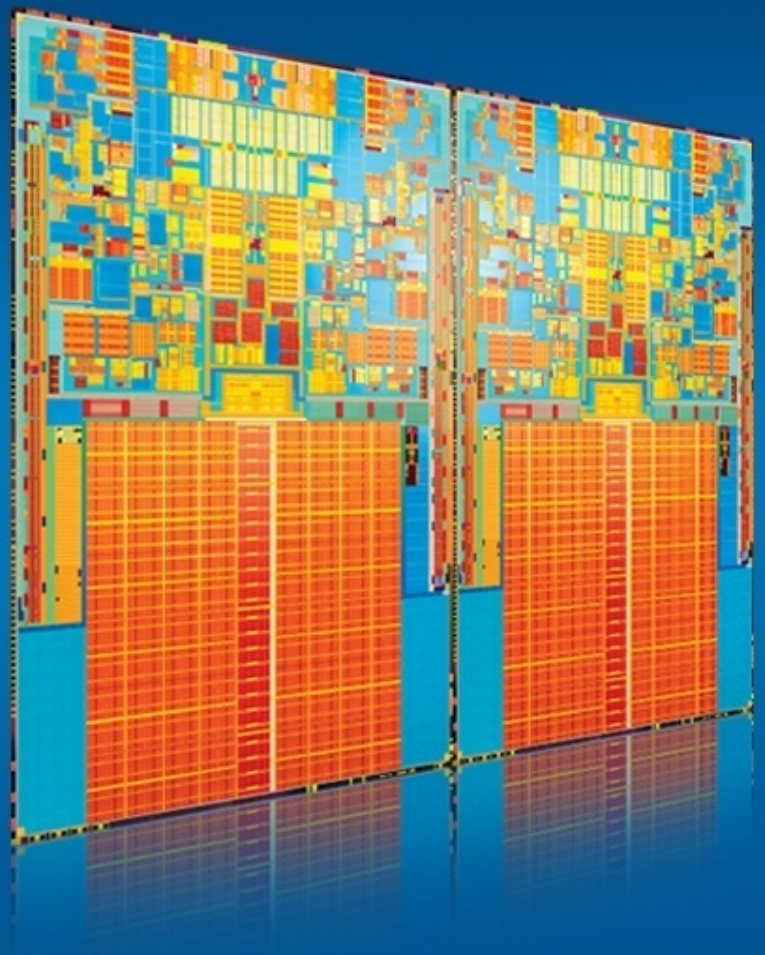
- Concurrency
  - Is a measurement of the number of active threads





Sample Concurrency View





Lab 2

Analyzing Concurrency Issues

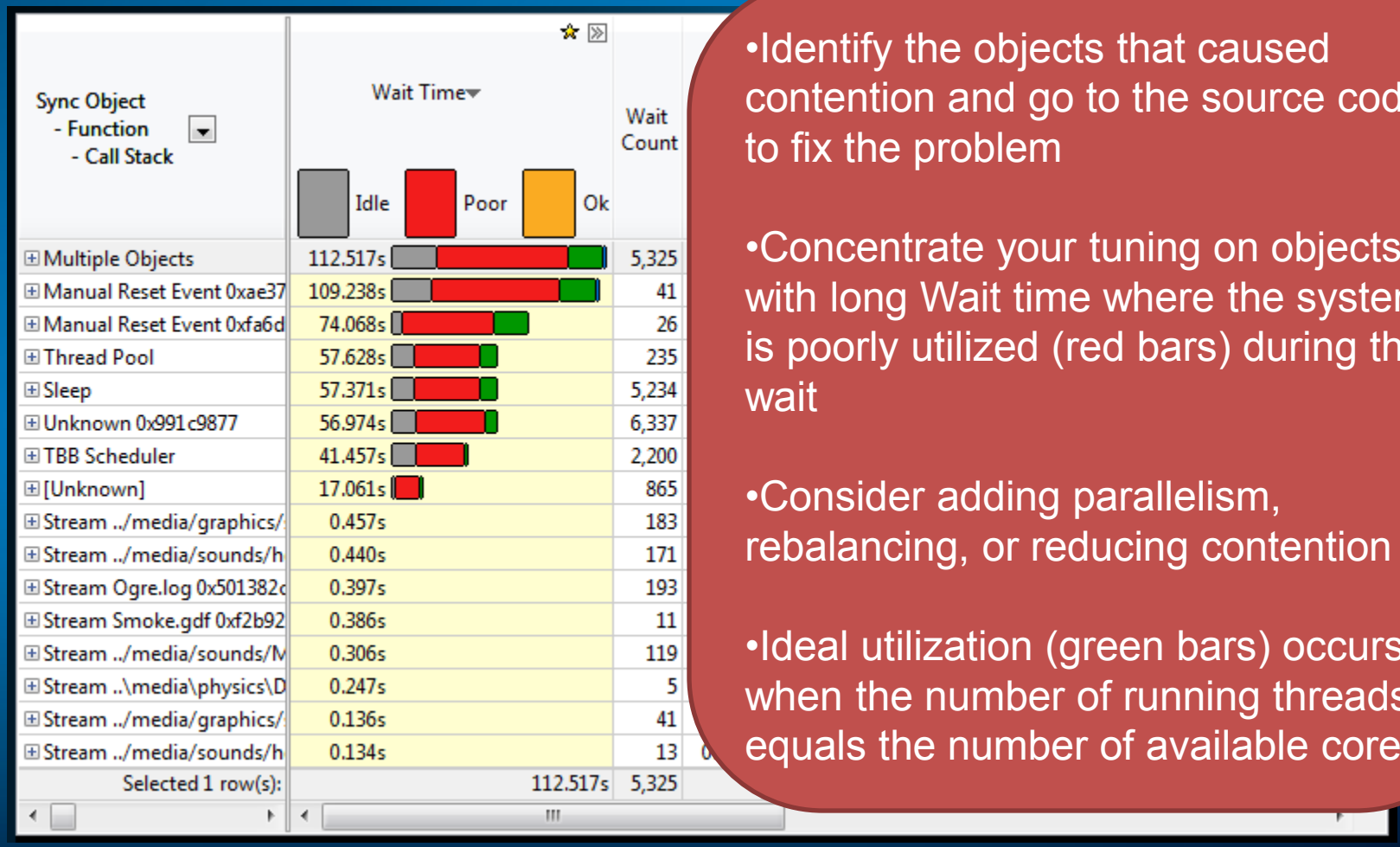
intel Software

Analysis Types
Lock and Waits

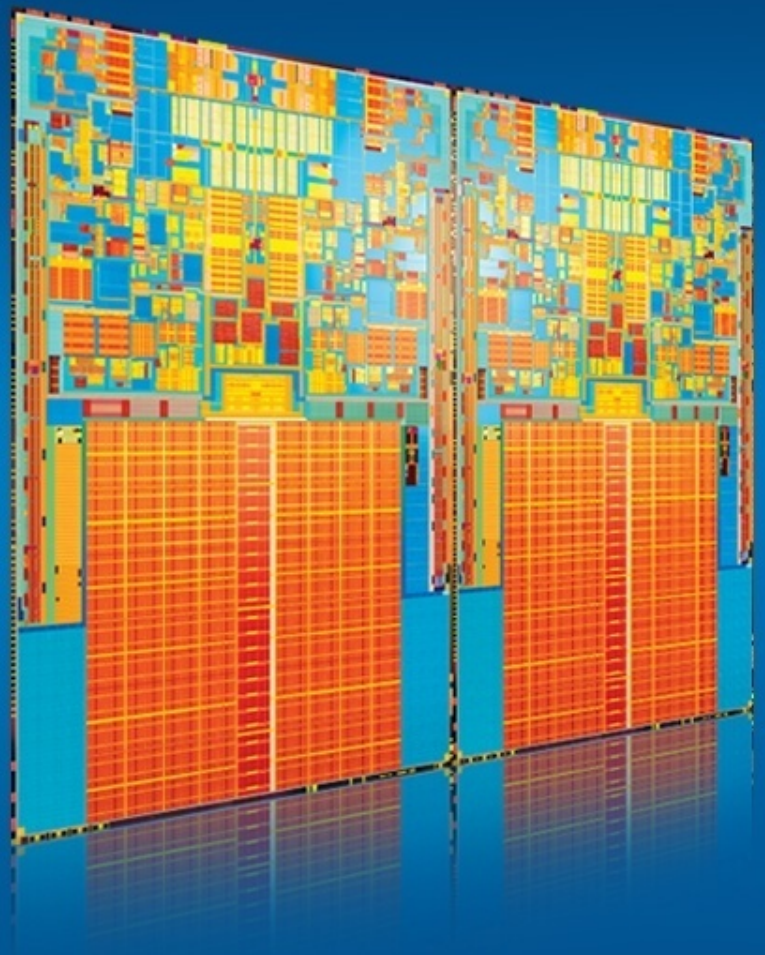
- Shows how an application is utilizing available CPU cores
- Helps identify the cause of ineffective utilization,
  - threads waiting too long on synchronization objects (locks),
  - I/O related issues
  - Timers while CPU cores are underutilized
- Overheads are higher than Hotspots analysis type

Analysis Types

Lock and Waits



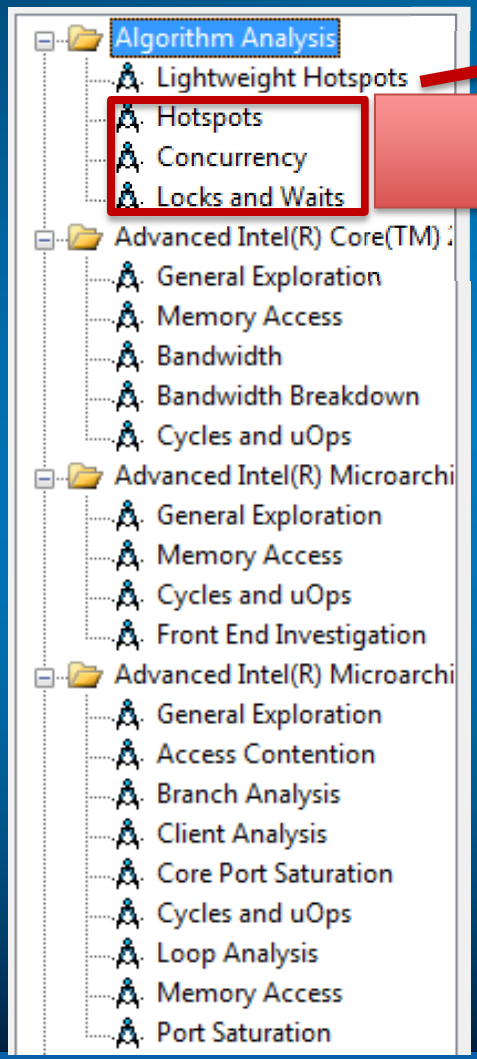
- Identify the objects that caused contention and go to the source code to fix the problem
- Concentrate your tuning on objects with long Wait time where the system is poorly utilized (red bars) during the wait
- Consider adding parallelism, rebalancing, or reducing contention
- Ideal utilization (green bars) occurs when the number of running threads equals the number of available cores



Lab 3
Locks and Waits Analysis

intel Software

VTune™ Amplifier XE
Pre-defined Analysis Types



Lightweight Hotspot analysis based on the underlying architecture

User mode stack sampling
Threading, IO, Signaling API instrumentation

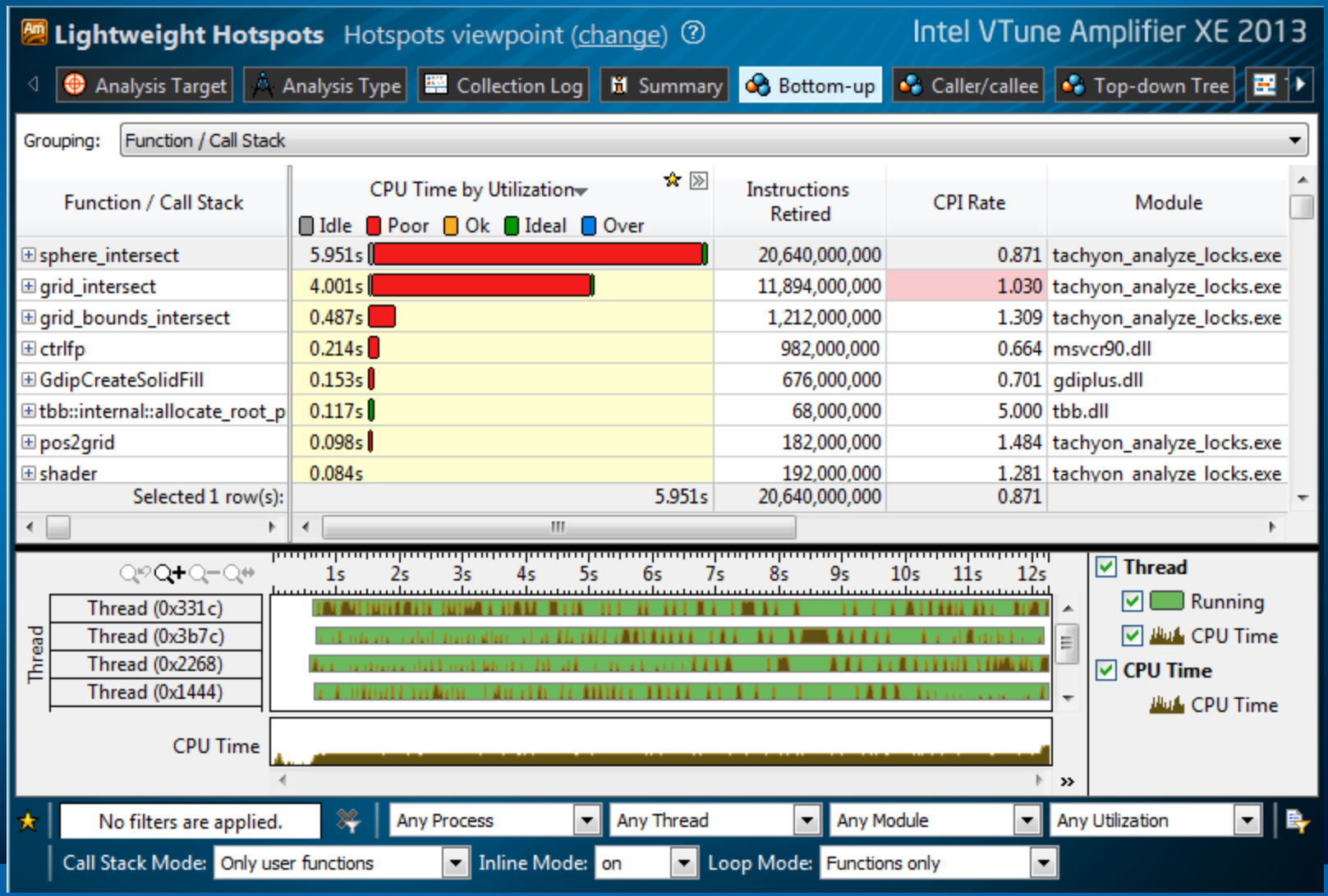
Analysis Types
Lightweight Hotspots

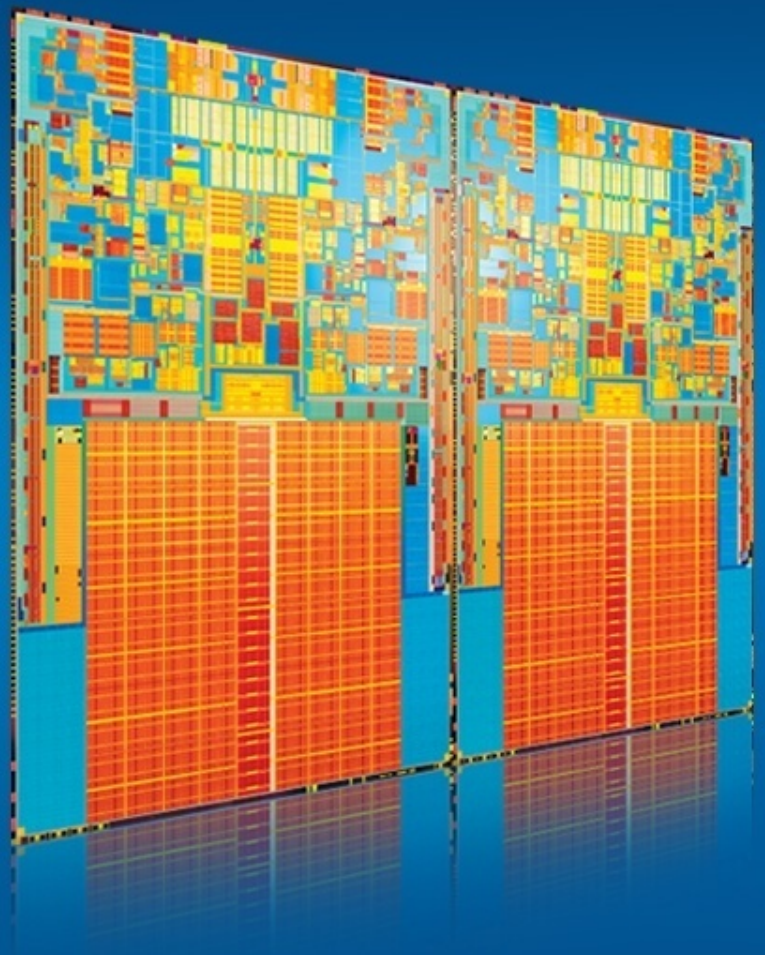
- New for Amplifier XE
- Similar to Hotspot Analysis
  - Sampling is performed with the SEP collector
  - Driver is required
- Stack walking is not performed
  - Only hotspots are reported
- Samples are taken more frequently, but may have less accurate timing information
- Analysis may be performed for a single application or for the entire system

Lightweight Hotspots vs. Hotspots

- Lightweight Hotspots
  - Uses Hardware EBS
  - Lower overhead
  - System wide profiling
  - Faster finalization
  - Requires a driver/privileged mode
- Hotspots
  - Uses software sampling collector
  - Stackwalk is done post process, hence slower finalization times
  - Provides stack information and a statistical call graph

Lightweight Hotspots





Lab 4
Lightweight Hotspots

intel Software

Intel® VTune™ Amplifier XE

Command Line Interface (CLI)

- **Command Line Interface use cases:**
  - Test code changes for performance regressions
  - Automate execution of performance analyses

- **Command Line Interface features:**
  - Fine-grained control of all analysis types and options
  - Text-based analysis reports
  - Analysis results can be opened in the graphical user interface

Intel® VTune™ Amplifier XE

Command Line Interface

```
$ amplxe-cl -help
Intel(R) VTune(TM) Amplifier Command Line Tool
Copyright (C) 2009-2013 Intel Corporation. All rights reserved.

Usage: amplxe-cl <-action> [-action-option] [-global-option] [[--] target [target options]]

 To view the results in the IDE, double-click the <resultname>.amplxe
 file located in the result directory.

 Type amplxe-cl -help <action>' for help on a specific action.

 Available actions:

   collect
   collect-with
   command
   finalize
   help
   import
   report
   version
```

Intel® VTune™ Amplifier XE

Command Line Interface Examples

- Display a list of available analysis types and preset configuration levels

```
amplxe-cl -help collect
```

- Run Hot Spot analysis on target *myApp* and store result in default-named directory, such as *r000hs*

```
amplxe-cl -collect hotspots myApp
```

- Run the Parallelism analysis, store the result in directory *r001par*

```
amplxe-cl -collect concurrency -result-dir r001par myApp
```

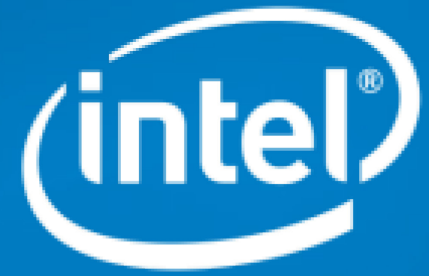
Intel® VTune™ Amplifier XE

Command Line Interface - reporting

```
Examples
Generate the 'hotspots' report for the result directory 'r000hs'.
    amplxe-cl -report hotspots -r r000hs
Group the data by module.
    amplxe-cl -report hotspots -r r000hs -group-by module

Filter the output.
     amplxe-cl -report hotspots -r r000hs -filter module=libexample.so

Use '-help report <report name>' for more information about each report.

Available reports
    callstacks   Display CPU or wait time for callstacks.
    gprof-cc     Display CPU or wait time in the gprof-like format.
    hotspots     Display CPU time.
    hw-events    Display hardware events.
    perf         Display CPU or wait time for each module.
    perf-detail  Display CPU time for each function or synchronization object.
    pmu-events
    sfdump       Specialized report to display hardware events.
    summary      Display data about overall performance.
    top-down     Display a call tree for your target application and provide CPU and wait time for each
function.
    wait-time    Display wait time.
```

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



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

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.

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