INTEL® VTUNE™ AMPLIFIER

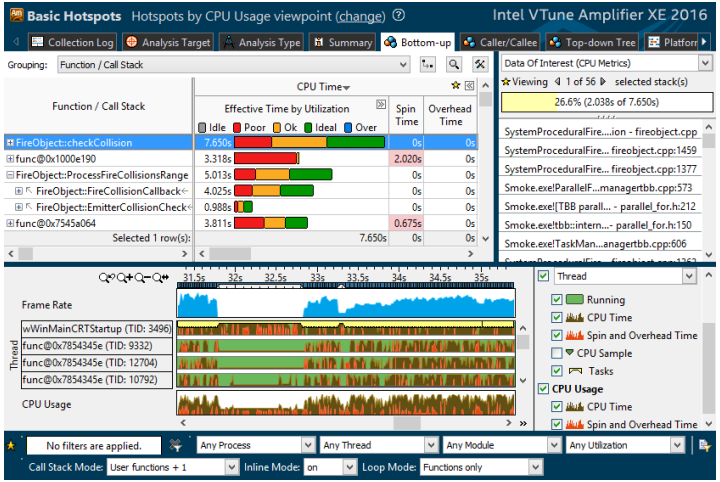
PERFORMANCE PROFILER

Faster, Scaleable Code, Faster

Intel® VTune™ Amplifier Performance Profiler

Get Faster Code Faster With Accurate Data & Meaningful Analysis

- Accurate CPU, GPU and threading data
- OpenMP region efficiency analysis
- Powerful data analysis & filtering
- Data displayed on the source code
- Easy set-up, no special compiles



For Windows* and Linux* From $899
(UI only now available on OS X*)

> "Last week, Intel® VTune™ Amplifier helped us find almost 3X performance improvement. This week it helped us improve the performance another 3X."

Claire Cates
Principal Developer
SAS Institute Inc.

http://intel.ly/vtune-amplifier-xe

Intel® VTune™ Amplifier

Faster, Scaleable Code, Faster

Get the Data You Need

- Hotspot (Statistical call tree), Call counts (Statistical)
- Thread Profiling – Concurrency and Lock & Waits Analysis
- Cache miss, Bandwidth analysis…[1]
- GPU Offload and OpenCL™ Kernel Tracing

Find Answers Fast

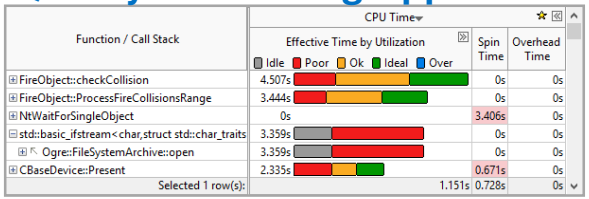
- View Results on the Source / Assembly
- OpenMP Scalability Analysis, Graphical Frame Analysis
- Filter Out Extraneous Data – Organize Data with Viewpoints
- Visualize Thread & Task Activity on the Timeline

Easy to Use

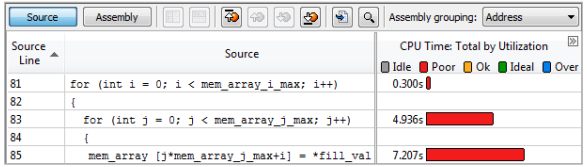
- No Special Compiles – C, C++, C#, Fortran, Java, ASM
- Visual Studio* Integration or Stand Alone
- Graphical Interface & Command Line
- Local & Remote Data Collection
- Analyze Windows* & Linux* data on OS X*[2]

[1] Events vary by processor. [2] No data collection on OS X*

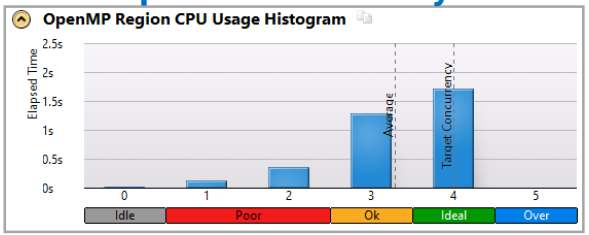
Quickly Find Tuning Opportunities



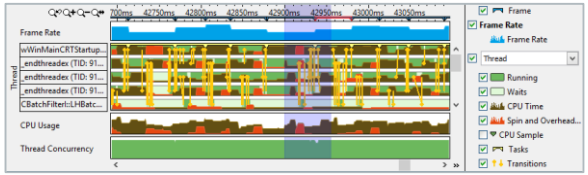
See Results On The Source Code



Tune OpenMP Scalability



Visualize & Filter Data



Good Tuning Data Gets Good Results

"We achieved a significant improvement (almost 2x) even on one core by optimizing the code based on the information provided by Intel® VTune™ Amplifier."

Alexey Andrianov,
R&D Director Deputy
Mechanical Analysis Division
Mentor Graphics Corporation

"The new VTune™ Amplifier brings even more capability to an already indispensable tool. The sampling based call stack hotspots is excellent and alone is worthy of the upgrade. We have also been impressed by how the concurrency and Locks and Waits analysis can even provide useful data on complex applications such as Premiere Pro."

Rich Gerber,
Engineering Manager
MediaCore
Adobe Systems Inc.

"Intel® VTune™ Amplifier analyzes complex code and helps us identify bottlenecks rapidly. By using it and other Intel® Software Development Tools, we were able to improve PIPESIM performance up to 10 times compared with the previous software version."

Details

Rodney Lessard
Senior Scientist
Schlumberger

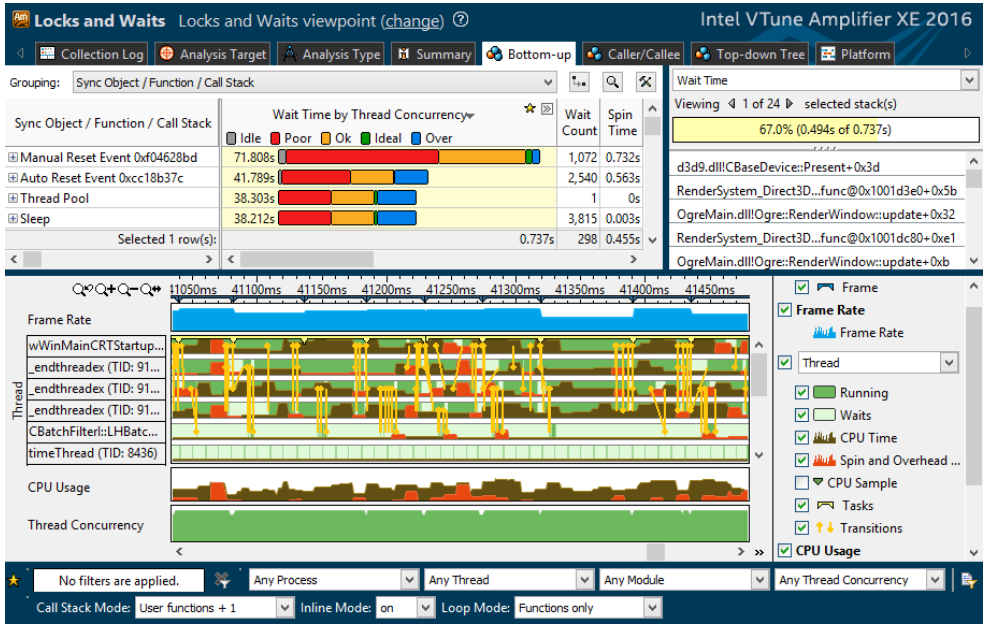
More Case Studies

Intel® VTune™ Amplifier

Tune Applications for Scalable Multicore Performance

Agenda

- Data Collection –
  Rich set of performance data
- Data Analysis -
  Find answers fast
- Flexible workflow –
  - User i/f and command line
  - Compare results
  - Remote collection
- New for 2016!
- Summary



Two Great Ways to Collect Data

Intel® VTune™ Amplifier

Software Collector	Hardware Collector
Uses OS interrupts	Uses the on chip Performance Monitoring Unit (PMU)
Collects from a single process tree	Collect system wide or from a single process tree.
~10ms default resolution	~1ms default resolution (finer granularity - finds small functions)
Either an Intel® or a compatible processor	Requires a genuine Intel® processor for collection
Call stacks show calling sequence	Optionally collect call stacks
Works in virtual environments	Works in a VM only when supported by the VM (e.g., vSphere*, KVM)
No driver required	Requires a driver - Easy to install on Windows - Linux requires root (or use default perf driver)

No special recompiles – C, C++, C#, Fortran, Java, Assembly

A Rich Set of Performance Data

Intel® VTune™ Amplifier

Software Collector	Hardware Collector
Basic Hotspots Which functions use the most time?	**Advanced Hotspots** Which functions use the most time? Where to inline? – Statistical call counts
Concurrency Tune parallelism. Colors show number of cores used.	**General Exploration** Where is the biggest opportunity? Cache misses? Branch mispredictions?
Locks and Waits Tune the #1 cause of slow threaded performance: – waiting with idle cores.	**Advanced Analysis** Dig deep to tune access contention, etc.
Any IA86 processor, any VM, no driver	Higher res., lower overhead, system wide

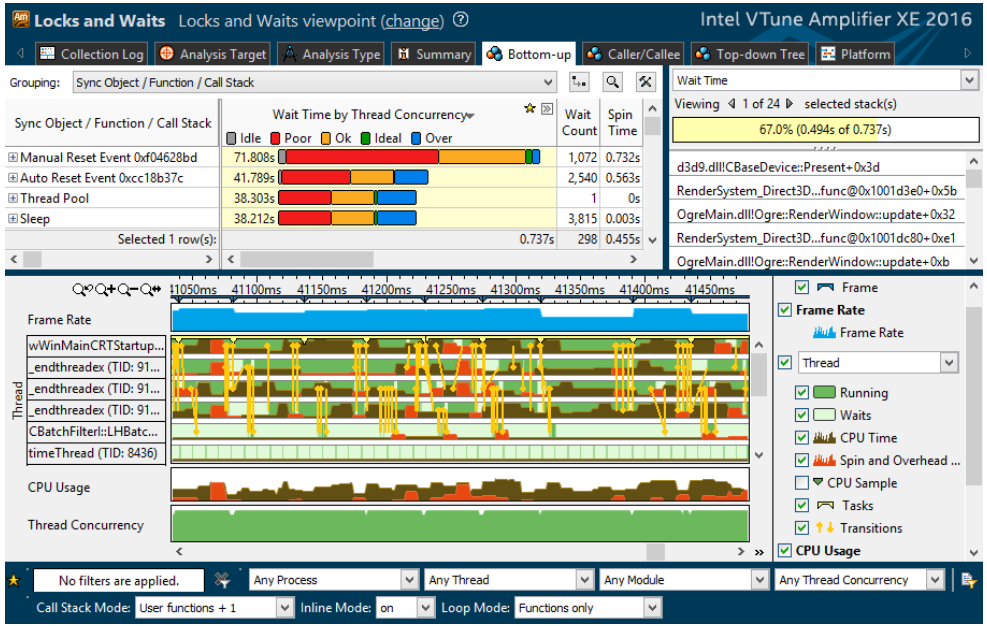
No special recompiles - C, C++, C#, Fortran, Java, Assembly

Intel® VTune™ Amplifier

Tune Applications for Scalable Multicore Performance

Agenda

- Data Collection –
  Rich set of performance data
- Data Analysis -
  Find answers fast
- Flexible workflow –
  - User i/f and command line
  - Compare results
  - Remote collection
- New for 2016!
- Summary



Find Answers Fast

Intel® VTune™ Amplifier

Adjust Data Grouping

Function - Call Stack
Module - Function - Call Stack
Source File - Function - Call Stack
Thread - Function - Call Stack

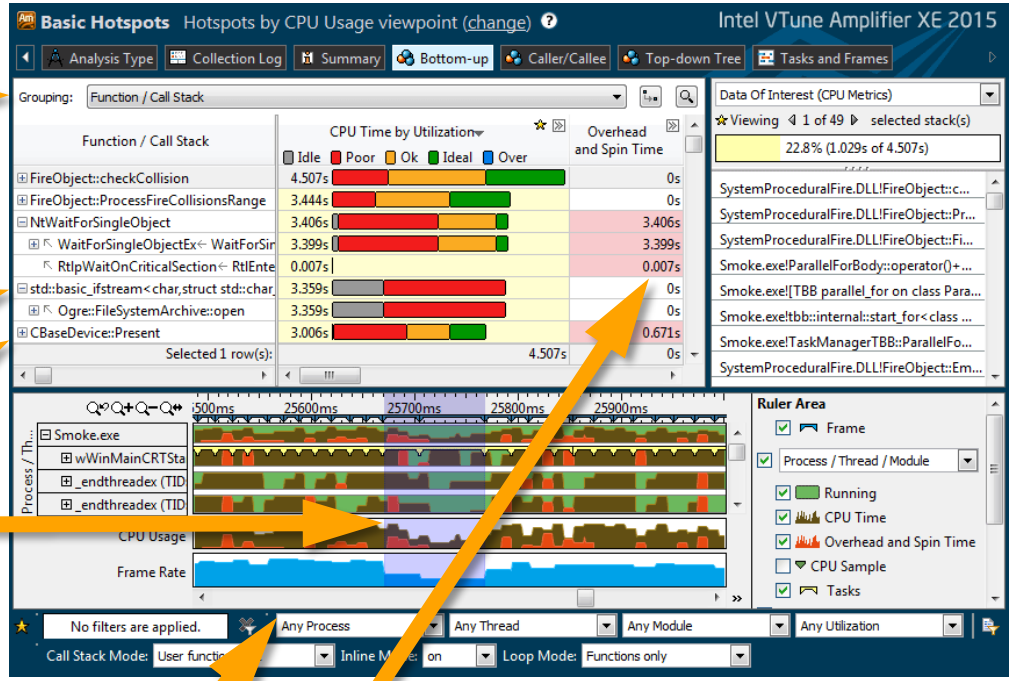
... (Partial list shown)

Double Click Function to View Source

Click [+] for Call Stack

Filter by Timeline Selection (or by Grid Selection)

Zoom In And Filter On Selection
Filter In by Selection
Remove All Filters



Filter by Process & Other Controls

Tuning Opportunities Shown in Pink. Hover for Tips

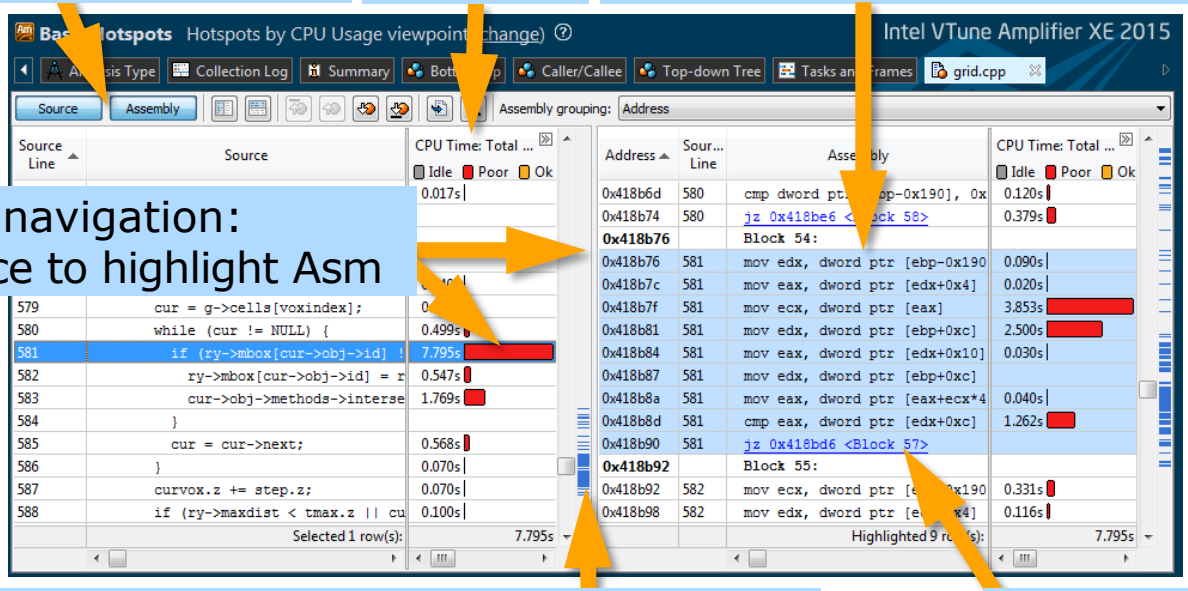
See Profile Data On Source / Asm

Double Click from Grid or Timeline

View Source / Asm or both

CPU Time

Right click for instruction reference manual



Quick Asm navigation:
Select source to highlight Asm

Scroll Bar “Heat Map” is an overview of hot spots

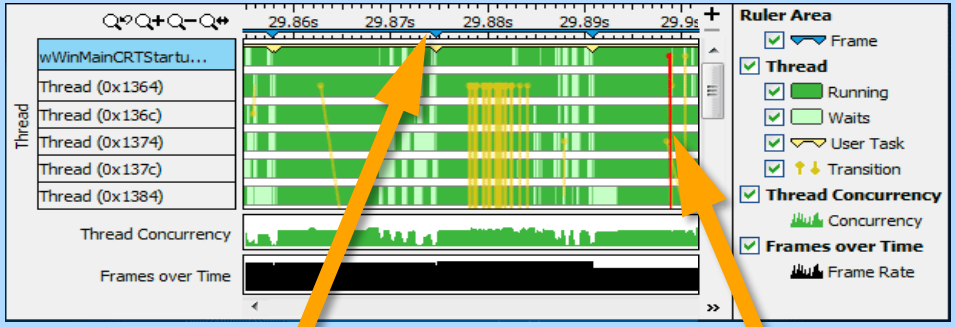
Click jump to scroll Asm

Timeline Visualizes Thread Behavior

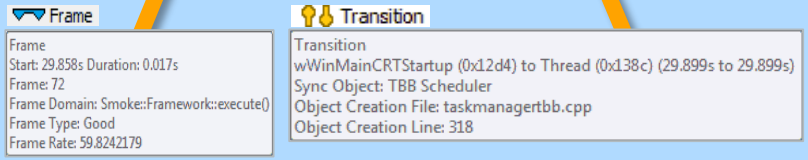
Intel® VTune™ Amplifier

Transitions

Locks & Waits

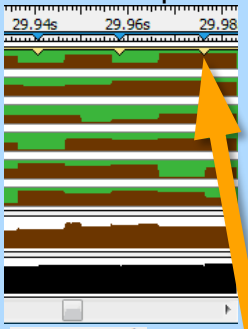


Hovers:

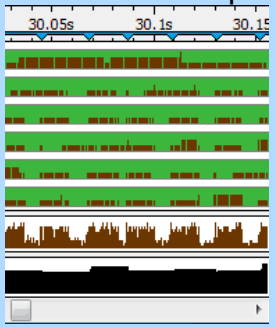


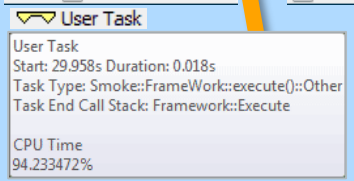
CPU Time

Basic Hotspots



Advanced Hotspots





Optional: Use API to mark frames and user tasks 

Optional: Add a mark during collection 

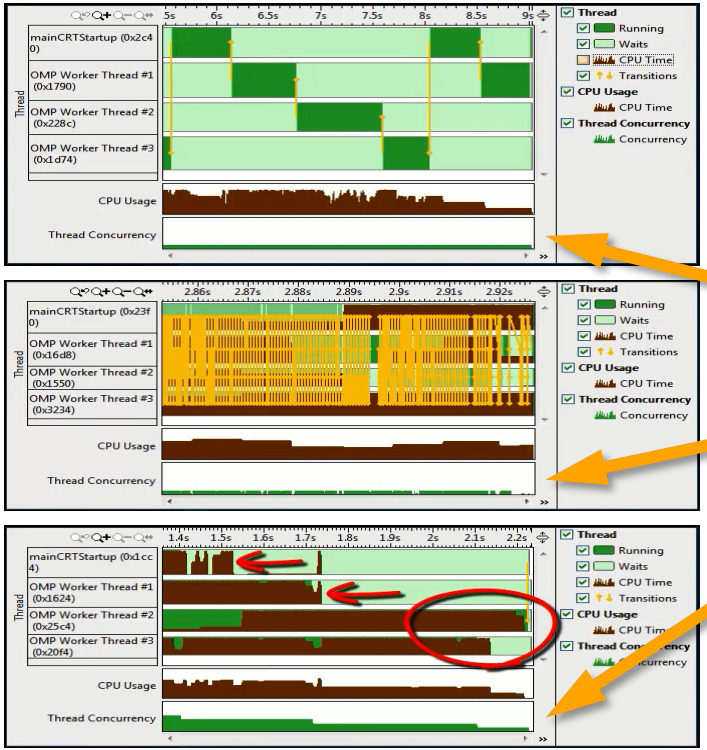
Visualize Parallel Performance Issues

Look for Common Patterns

Coarse Grain Locks

High Lock Contention

Load Imbalance

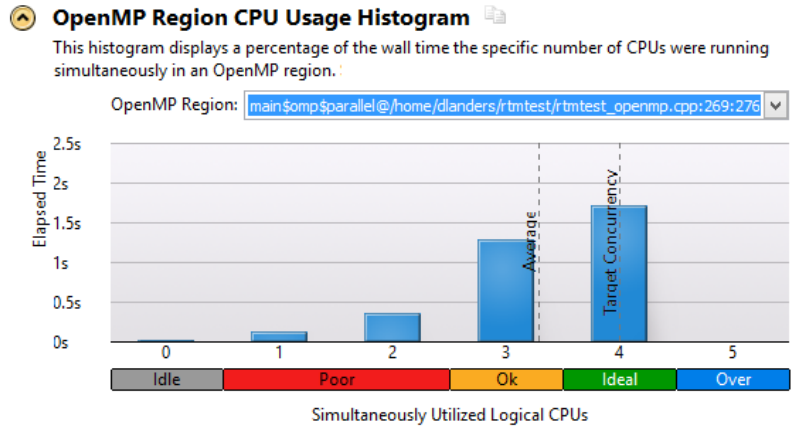


Low Concurrency

New for 2016! Faster Code Faster!

Intel® VTune™ Amplifier Performance Profiler

- Tune OpenMP Scalability Faster
- Easier Multi-Rank Analysis
- Easier OpenCL & GPU Analysis
- Memory Access Analysis & Better Bandwidth Analysis
- Faster, Easier to Use & Install
- Added VM Support
- Latest Processors & OSs



Get The Data You Need

Tune OpenMP for Efficiency and Scalability

Typical Questions:

Q: *"I put in pragmas, but why is my speed up far from linear?"*

A: Parallelization inefficiency

Q: *"I ran my app on a system with more cores but why does it run less efficiently than on the system with fewer cores?"*

A: Scalability issues

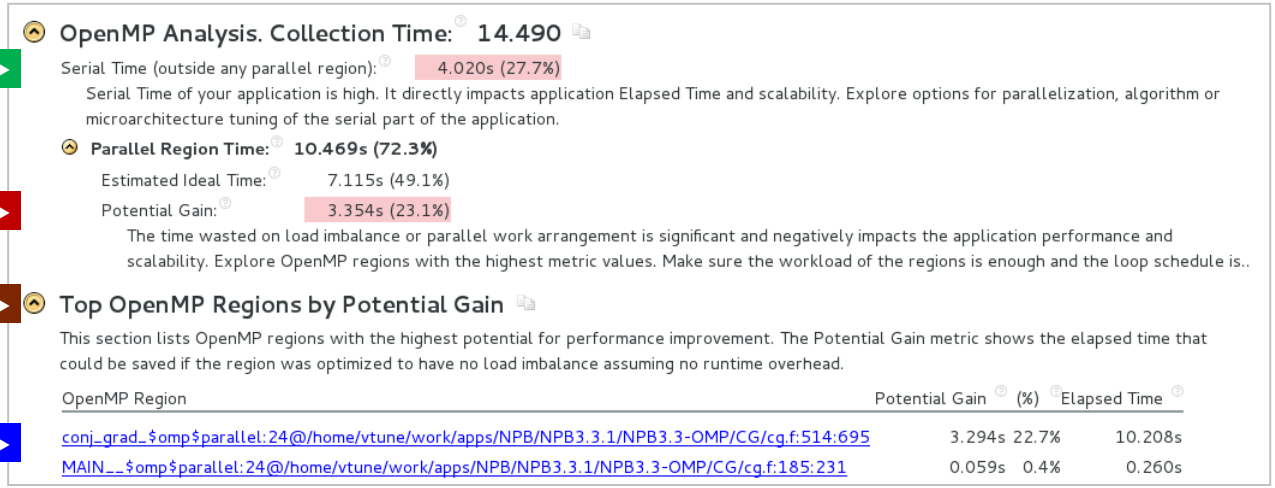
Data Needed:

1) Is the serial time of my application significant enough to prevent scaling?
2) How much gain can be achieved by tuning OpenMP?
3) Which OpenMP regions / loops / barriers will benefit most from tuning?
4) What are the inefficiencies with each region?

Tune OpenMP for Efficiency and Scalability

New!

Fast Answers: Is My OpenMP Scalable? How Much Faster Could It Be?



The summary view shown above gives fast answers to four important OpenMP tuning questions:

1) Is the serial time of my application significant enough to prevent scaling?
2) How much performance can be gained by tuning OpenMP?
3) Which OpenMP regions / loops / barriers will benefit most from tuning?
4) What are the inefficiencies with each region? (click the link to see details)

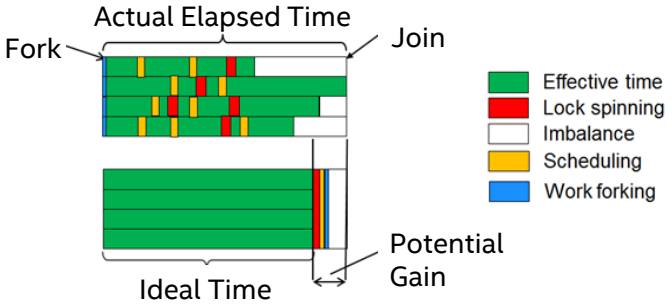
New!

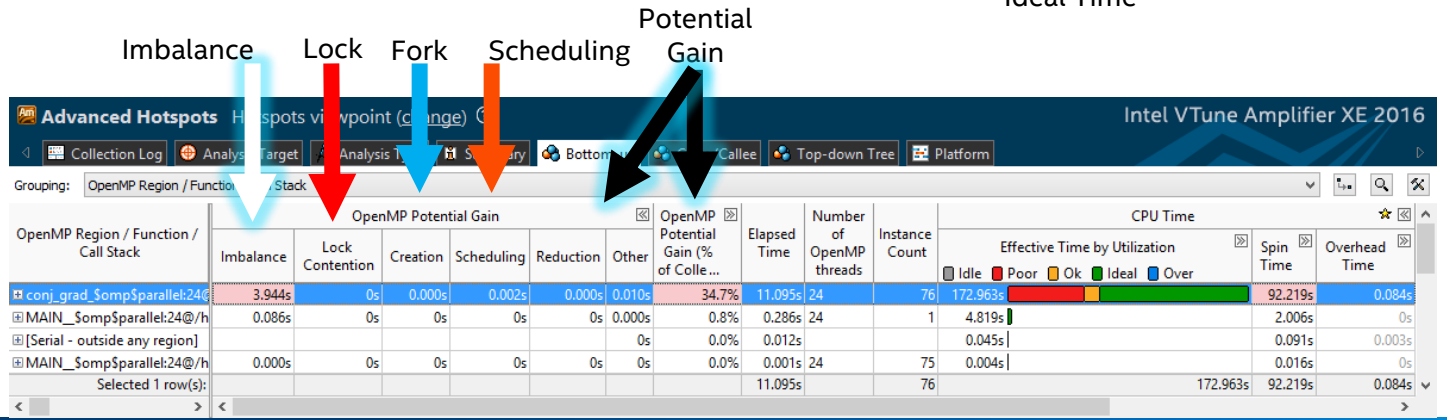
Tune OpenMP for Efficiency and Scalability

See the wall clock impact of inefficiencies, identify their cause

Focus On What's Important

- What region is inefficient?
- Is the potential gain worth it?
- Why is it inefficient?
  Imbalance? Scheduling? Lock spinning?
- Intel® Xeon Phi systems supported

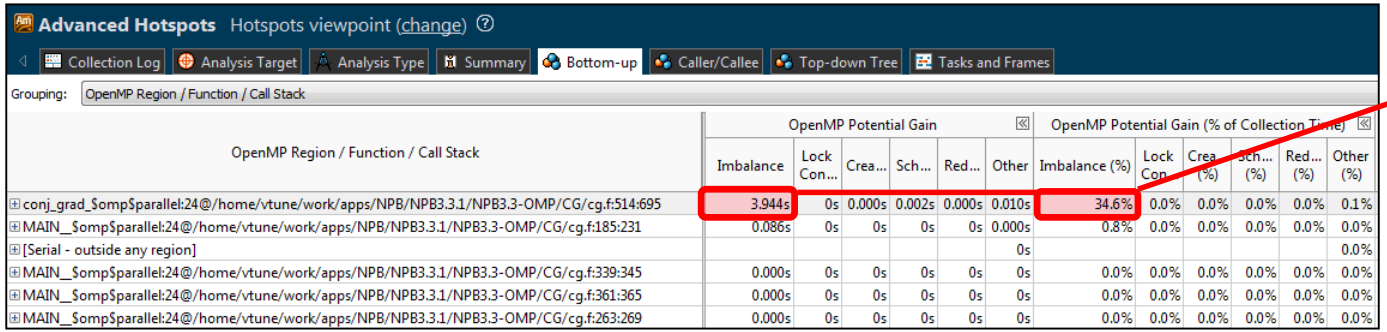




New!

What is Hindering Parallel Performance?

VTune™ Amplifier Identifies Parallel Region Inefficiencies



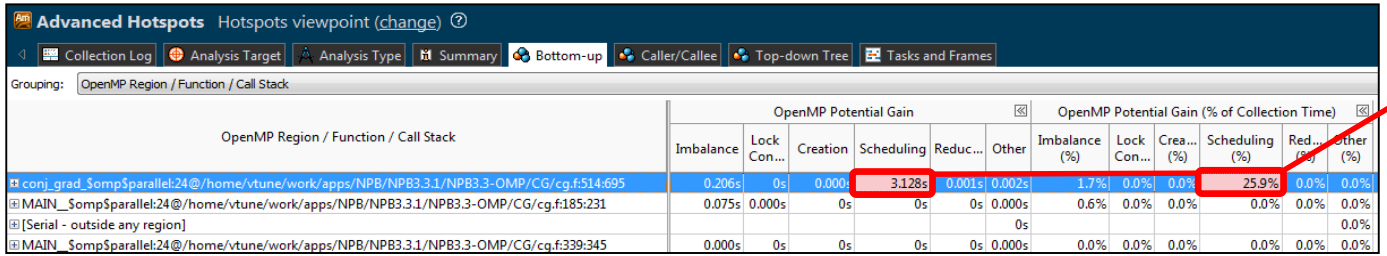
Advanced Hotspots Hotspots viewpoint (change)

Collection Log | Analysis Target | Analysis Type | Summary | Bottom-up | Caller/Callee | Top-down Tree | Tasks and Frames

Grouping: OpenMP Region / Function / Call Stack

OpenMP Region / Function / Call Stack	OpenMP Potential Gain						OpenMP Potential Gain (% of Collection Time)					
	Imbalance	Lock Con...	Crea...	Sch...	Red...	Other	Imbalance (%)	Lock Con...	Crea... (%)	Sch... (%)	Red... (%)	Other (%)
conj_grad_ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:514:695	3.944s	0s	0.000s	0.002s	0.000s	0.010s	34.6%	0.0%	0.0%	0.0%	0.0%	0.1%
MAIN__ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:185:231	0.086s	0s	0s	0s	0s	0.000s	0.8%	0.0%	0.0%	0.0%	0.0%	0.0%
[Serial - outside any region]						0s						0.0%
MAIN__ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:339:345	0.000s	0s	0s	0s	0s	0s	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
MAIN__ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:361:365	0.000s	0s	0s	0s	0s	0s	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
MAIN__ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:263:269	0.000s	0s	0s	0s	0s	0s	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Imbalance



Advanced Hotspots Hotspots viewpoint (change)

Collection Log | Analysis Target | Analysis Type | Summary | Bottom-up | Caller/Callee | Top-down Tree | Tasks and Frames

Grouping: OpenMP Region / Function / Call Stack

OpenMP Region / Function / Call Stack	OpenMP Potential Gain						OpenMP Potential Gain (% of Collection Time)					
	Imbalance	Lock Con...	Creation	Scheduling	Reduc...	Other	Imbalance (%)	Lock Con...	Crea... (%)	Scheduling (%)	Red... (%)	Other (%)
conj_grad_ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:514:695	0.206s	0s	0.000s	3.128s	0.001s	0.002s	1.7%	0.0%	0.0%	25.9%	0.0%	0.0%
MAIN__ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:185:231	0.075s	0.000s	0s	0s	0s	0.000s	0.6%	0.0%	0.0%	0.0%	0.0%	0.0%
[Serial - outside any region]						0s						0.0%
MAIN__ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:339:345	0.000s	0s	0s	0s	0s	0.000s	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

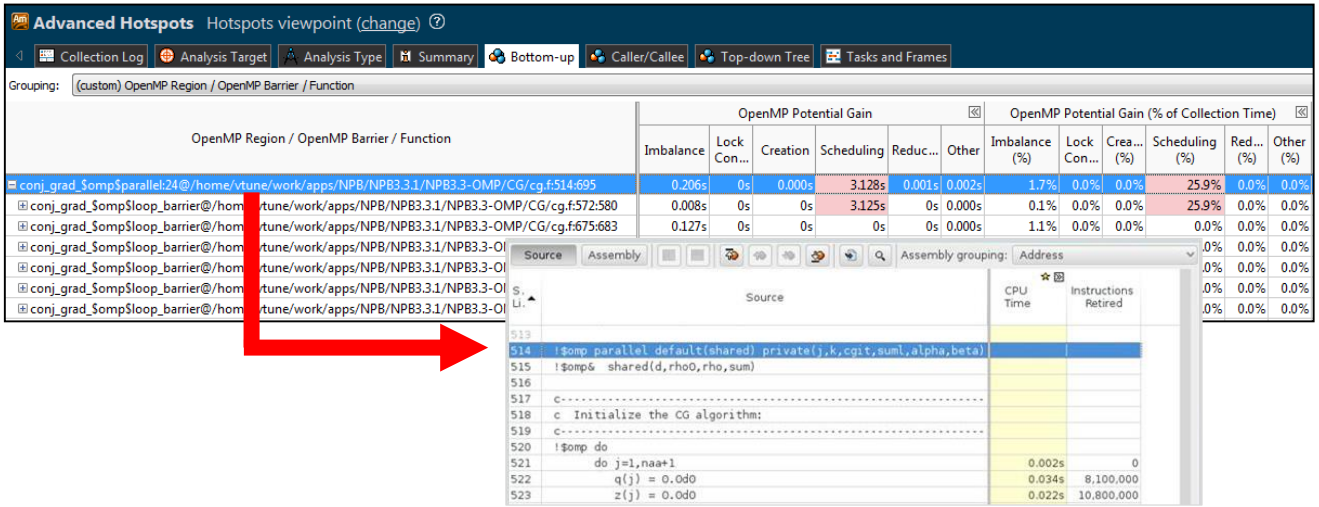
Likely culprit: Dynamic scheduling overhead

New!

Jump to Parallel Region Source Code

Find Answers Faster

- View data specific to the region at the source code level
- Tip: use '-parallel-source-info=2' compiler option to embed source file name in region name (enables drill down to source file)



New!

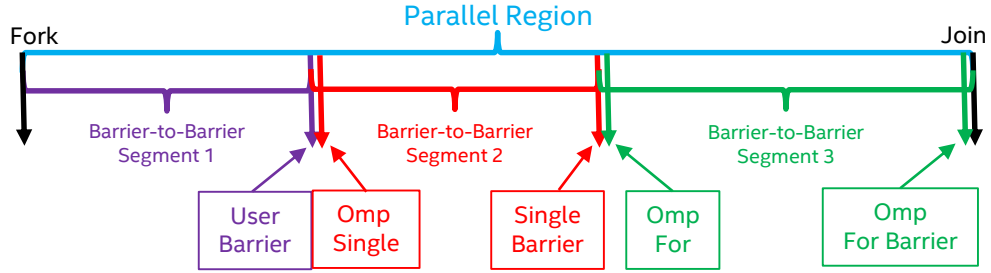
Tune OpenMP for Efficiency and Scalability

See inside each parallel region – Understand the cause of inefficiency

Detailed Barrier to Barrier Analysis

- Tune each segment separately
- Easier to see tuning opportunities

```
#pragma omp parallel
{
   ...
#pragma omp barrier
#pragma omp single
      {
         ...
      {
#pragma omp for
      {
```



Grouping: OpenMP Region / OpenMP Barrier-to-Barrier Segment / Function / Call Stack

OpenMP Region / OpenMP Barrier-to-Barrier Segment / Function / Call Stack	OpenMP Potential Gain						OpenMP Potential Gain (% of Collection ...
	Imbalance	Lock Contention	Creation	Scheduling	Reduction	Other	
3.3-OMP/CG/cg.f:514:695	3.944s	0s	0.000s	0.002s	0.000s	0.010s	34.7%
\PB3.3.1/NPB3.3-OMP/CG/cg.f:580	3.725s	0s	0s	0.000s	0s	0.008s	32.8%
\PB3.3.1/NPB3.3-OMP/CG/cg.f:683	0.149s	0s	0s	0s	0s	0.000s	1.3%
\PB3.3.1/NPB3.3-OMP/CG/cg.f:664	0.014s	0s	0s	0.000s	0s	0.000s	0.1%

New!

Tune OpenMP for Efficiency and Scalability

More precision where you need it most

Profile Small Region Instances

- Working thread imbalance is a major performance issue
- Tune using precise imbalance measurement (trace-based)

OpenMP Region / OpenMP Barrier-to-Barrier Segment / Function / Call Stack	OpenMP Potential Gain						OpenMP Potenti... Gain (% of Coll...	Spin Time			
	Imbalance	Lock Contention	Creation	Scheduling	Reduction	Other		Imbalance or Serial Spinning...	Lock ...	Com...	Other
z_solve_ompparallel:24@/home/vtune/work/apps/NPB/	1.244s	0s	0s	0s	0s	0.001s	6.3%	0.007s	0s	0s	0.007s
y_solve_ompparallel:24@/home/vtune/work/apps/NPB/	1.186s	0s	0s	0s	0s	0.001s	6.0%	0.005s	0s	0s	0.013s
x_solve_ompparallel:24@/home/vtune/work/apps/NPB/	1.153s	0.000s	0s	0.001s	0s	0.001s	5.9%	0.005s	0.003s	0s	0.005s
compute_rhs_ompparallel:24@/home/vtune/work/apps/	1.019s	0.000s	0s	0.000s	0.000s	0.009s	5.2%	0.067s	0.001s	0s	0.100s
add_ompparallel:24@/home/vtune/work/apps/NPB/NPB	0.098s	0.000s	0s	0s	0s	0.002s	0.5%	0.002s	0.003s	0s	0.015s
initialize_ompparallel:24@/home/vtune/work/apps/NPB	0.011s	0.000s	0s	0s	0s	0.003s	0.1%	0.001s	0.008s	0s	0s
exact_rhs_ompparallel:24@/home/vtune/work/apps/NP	0.003s	0.000s	0s	0.000s	0s	0.001s	0.0%	0.001s	0.001s	0s	0s

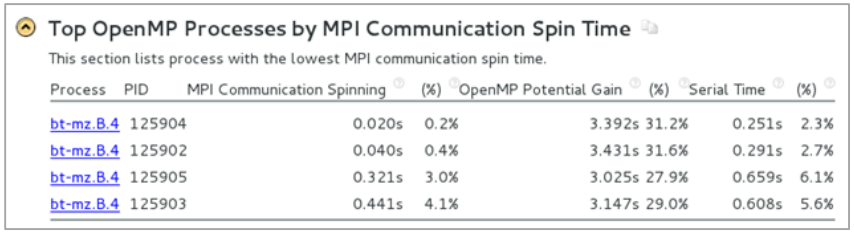
Precise imbalance - significant

almost 0 because KMP_BLOCKTME=0

New for 2016! Faster Code Faster!

Intel® VTune™ Amplifier Performance Profiler

- Tune OpenMP Scalability Faster
- **Easier Multi-Rank Analysis**
- Easier OpenCL & GPU Analysis
- Memory Access Analysis & Better Bandwidth Analysis
- Faster, Easier to Use & Install
- Added VM Support
- Latest Processors & OSs



Top OpenMP Processes by MPI Communication Spin Time

This section lists process with the lowest MPI communication spin time.

Process	PID	MPI Communication Spinning	(%)	OpenMP Potential Gain	(%)	Serial Time	(%)
bt-mz.B.4	125904	0.020s	0.2%	3.392s	31.2%	0.251s	2.3%
bt-mz.B.4	125902	0.040s	0.4%	3.431s	31.6%	0.291s	2.7%
bt-mz.B.4	125905	0.321s	3.0%	3.025s	27.9%	0.659s	6.1%
bt-mz.B.4	125903	0.441s	4.1%	3.147s	29.0%	0.608s	5.6%

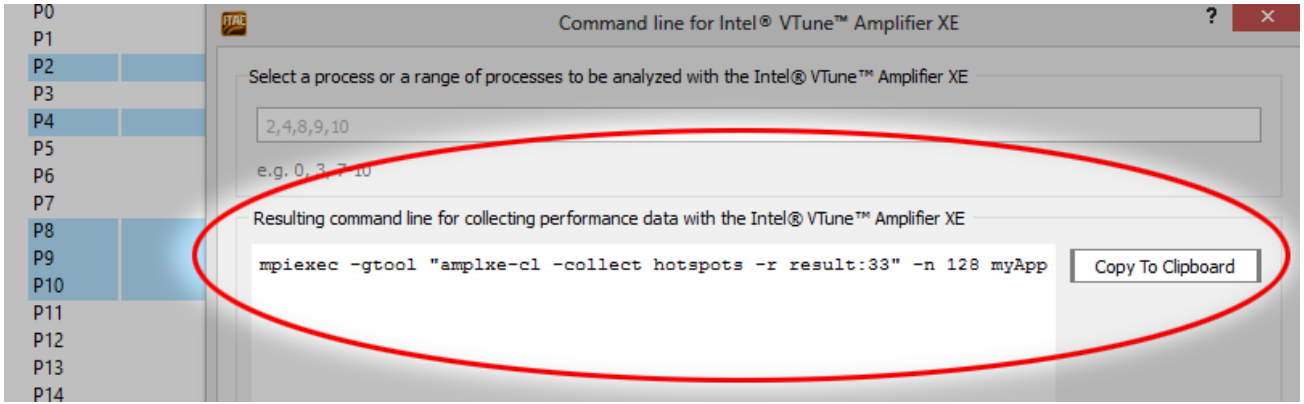
New!

Easier Multi-Rank Analysis of MPI + OpenMP

Tune hybrid parallelism using ITAC + VTune Amplifier

Intel Trace Analyzer & Collector (ITAC) - MPI Analyzer and Profiler

1) ITAC finds ranks with low MPI communication spin time
   These will benefit most from better OpenMP performance
2) Select these ranks for OpenMP profiling in VTune Amplifier



New!

Easier Multi-Rank Analysis of MPI + OpenMP

Tune hybrid parallelism using ITAC + VTune Amplifier

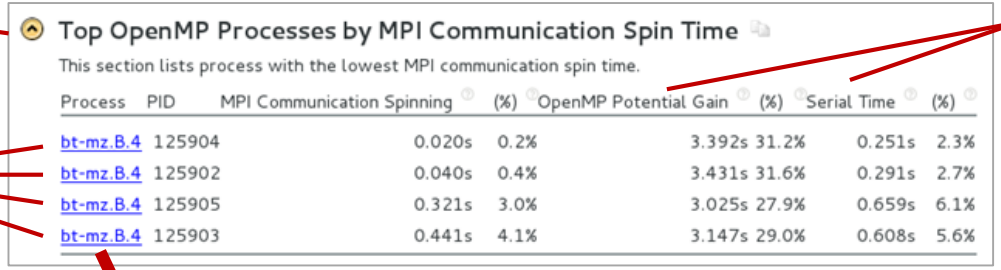
Tune OpenMP performance of high impact ranks in VTune Amplifier

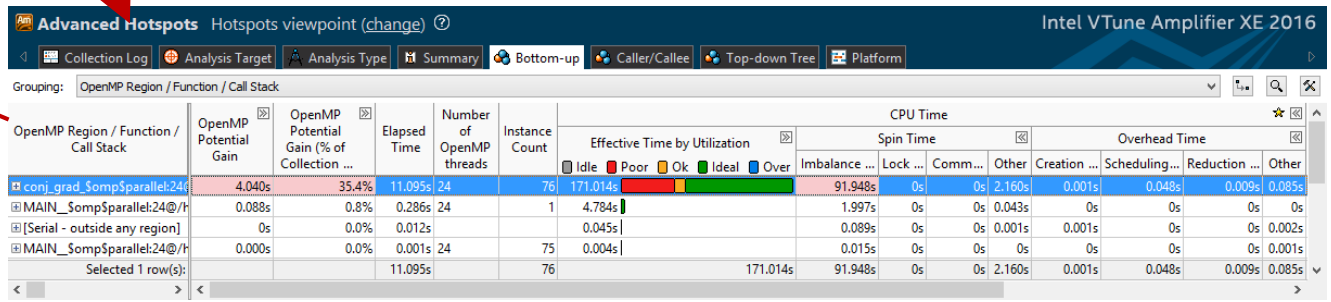
Ranks sorted by OpenMP tuning impact on overall performance

Process names link to OpenMP metrics

Detailed OpenMP metrics

Per-rank OpenMP Potential Gain and Serial Time metrics

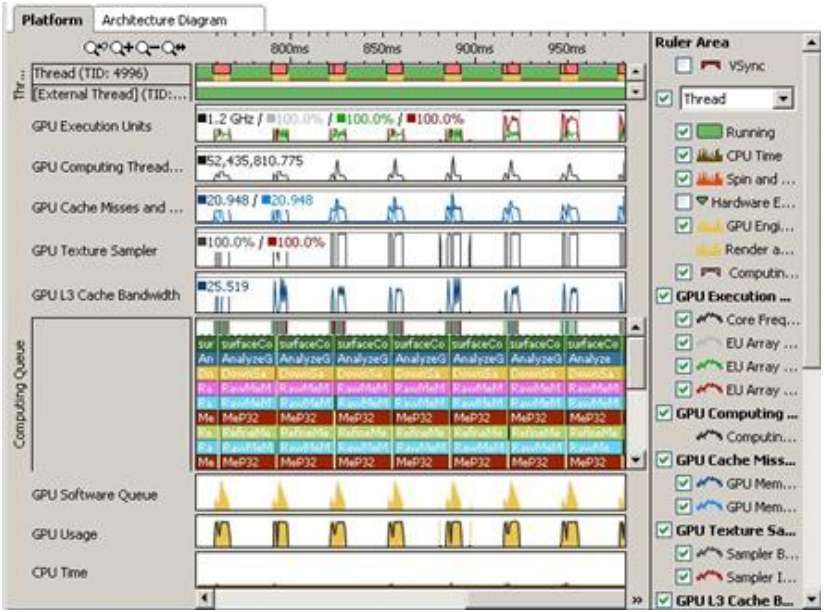




New for 2016! Faster Code Faster!

Intel® VTune™ Amplifier Performance Profiler

- Tune OpenMP Scalability Faster
- Easier Multi-Rank Analysis
- Easier OpenCL & GPU Analysis
- Memory Access Analysis & Better Bandwidth Analysis
- Faster, Easier to Use & Install
- Added VM Support
- Latest Processors & OSs

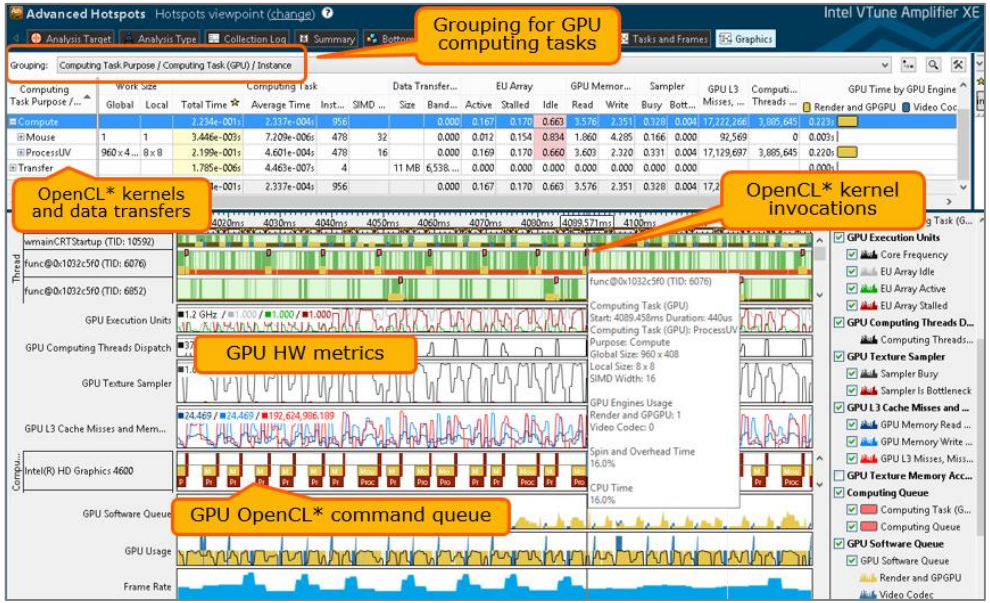


GPU Profiling for Linux* and Windows*

New!

Hardware and OpenCL* Metrics

- Intel® Integrated Graphics hardware metrics
- Details OpenCL™ activity on the GPU
- Correlated with CPU processes and threads
- Now on Linux* and Windows*



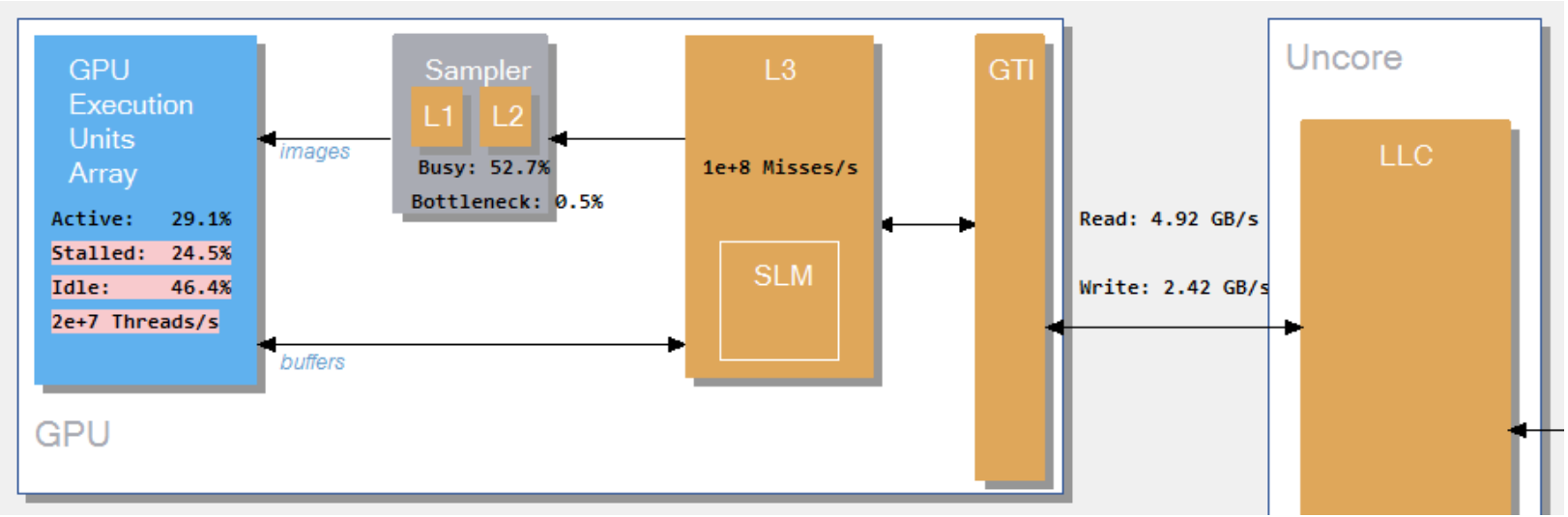
* See this Getting Started article

New!

Easier OpenCL & GPU Profiling

New! Intel® VTune™ Amplifier 2016

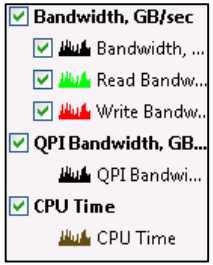
- OpenCL and GPU analysis for Linux*
- GPU Architecture Diagram – easy interpretation of GPU hardware metrics
- Intel Media SDK analysis
- Extended counter set for GPU on the 5th Generation Intel® Core™ processors



New for 2016! Faster Code Faster!

Intel® VTune™ Amplifier Performance Profiler

- Tune OpenMP Scalability Faster
- Easier Multi-Rank Analysis
- Easier OpenCL & GPU Analysis
- **Memory Access Analysis & Better Bandwidth Analysis**
- Faster, Easier to Use & Install
- Added VM Support
- Latest Processors & OSs



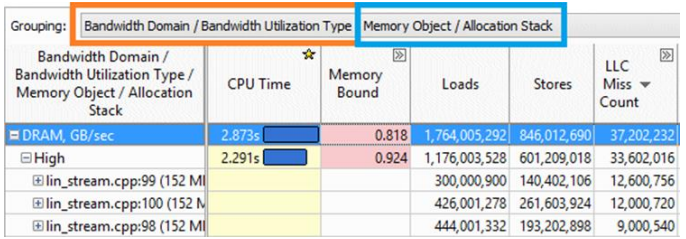
New!

Memory Access Analysis

New! Intel® VTune™ Amplifier 2016

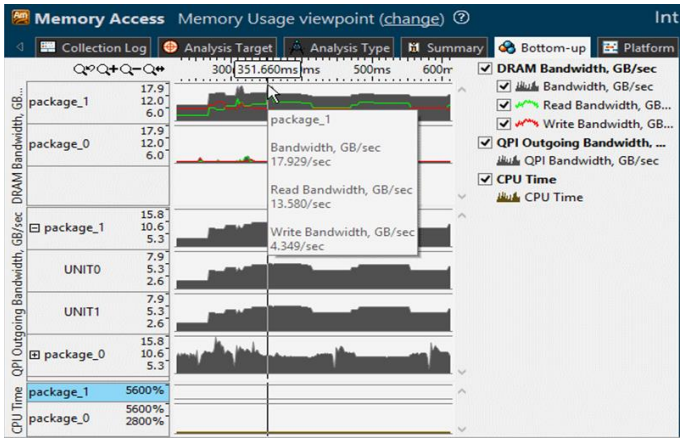
Tune data structures for better performance

- Attribute cache misses to data structures



Bandwidth Analysis for Non-Uniform Memory

- See Read & Write contributions to Total Bandwidth
- Easier tuning of multi-socket bandwidth



Seeing total bandwidth can suggest data blocking opportunities to change a bandwidth bound app into a compute bound app.

New for 2016! Faster Code Faster!

Intel® VTune™ Amplifier Performance Profiler

- Tune OpenMP Scalability Faster
- Easier Multi-Rank Analysis
- Easier OpenCL & GPU Analysis
- Memory Access Analysis & Better Bandwidth Analysis
- Faster, Easier to Use & Install
- Added VM Support
- Latest Processors & OSs

New!

Faster, More Responsive Interface

More data pre-processing for faster interactive performance

	Initial Release 2015 Mid-Size Data	Initial Release 2015 Large Data	Initial Release 2016 Both Sizes
Open Summary	10-40 seconds	1-2 minutes	~4-5 seconds
Open Timeline	10-40 seconds	2-5 minutes	~1-3 seconds
Zoom Timeline	5-10 seconds	1-2 minutes	~ 1 second
Grid Node Expand	5-15 seconds	>1 minute	<1 – 2 seconds
Finalization			Usually <2x slower Sometimes faster

- Larger data sets will see the greatest speedup.
- Small data sets may see little improvement.
- Some optimizations already released in 2015 updates.

Tip: For the best performance avoid VNC's slow graphics. Run the UI locally. Import data from the remote target.

Data based on typical results obtained during internal testing, your results may differ.
Results vary with different systems, applications and data sizes.

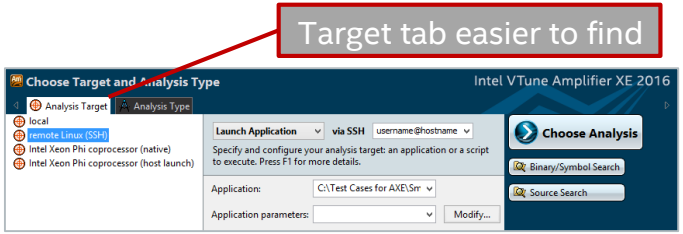
New!

Many Little Things to Make Life Easier

Easier install, setup, visualization and analysis with VMs

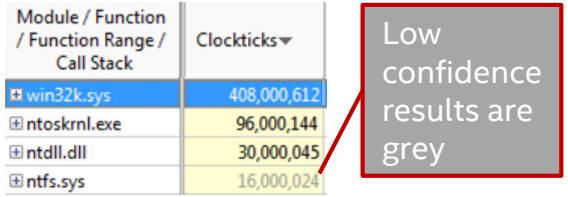
Easier Install & Setup

- Easier Linux* install:
  Driverless EBS with stacks (uses perf)
- Simplified setup of project & target –
  New “Target” tab is easy to find & use



Easier Analysis

- Confidence indication added to General Exploration analysis
- “Super-tiny” timeline view – visualize process/thread behavioral patterns



Easier Profiling When Running a VM

- New: EBS profiling under KVM*
- Existing: EBS profiling under VMware*[1]
- Existing: Software collectors work on all VMs

[1] With select VMware products. Contact VMware for details.

New!

New for 2016! Faster Code Faster!

Intel® VTune™ Amplifier Performance Profiler

Tune OpenMP Scalability Faster

- Pre-instrumented run time: actionable data without time consuming setup.
- OpenMP regions sorted by potential gain and cause of inefficiency (imbalance, etc.)

MPI Hybrid Analysis Made Easy

- Integration with MPI profiling makes rank selection and setup fast and easy.

Easier OpenCL & GPU Analysis

- Metrics on GPU diagram. Find issues faster
- Intel® Media SDK analysis
- Linux* (new!) and Windows* OSs

Easier to Use & Install

- Faster, more responsive user interface
- Confidence indicators flag metrics based on a very small sample size
- Driverless hardware sampling with stacks

Memory Access Analysis & Better Bandwidth Analysis

- Tune data structures for performance
- Easier optimization of multi-socket systems

More VM Support

- New: EBS profiling under KVM
- Existing: EBS profiling under VMware*[1]

[1] With select VMware products. Contact VMware for details.

INTEL® VTUNE™ AMPLIFIER

PERFORMANCE PROFILER

BACKUP - WHAT'S NEW?

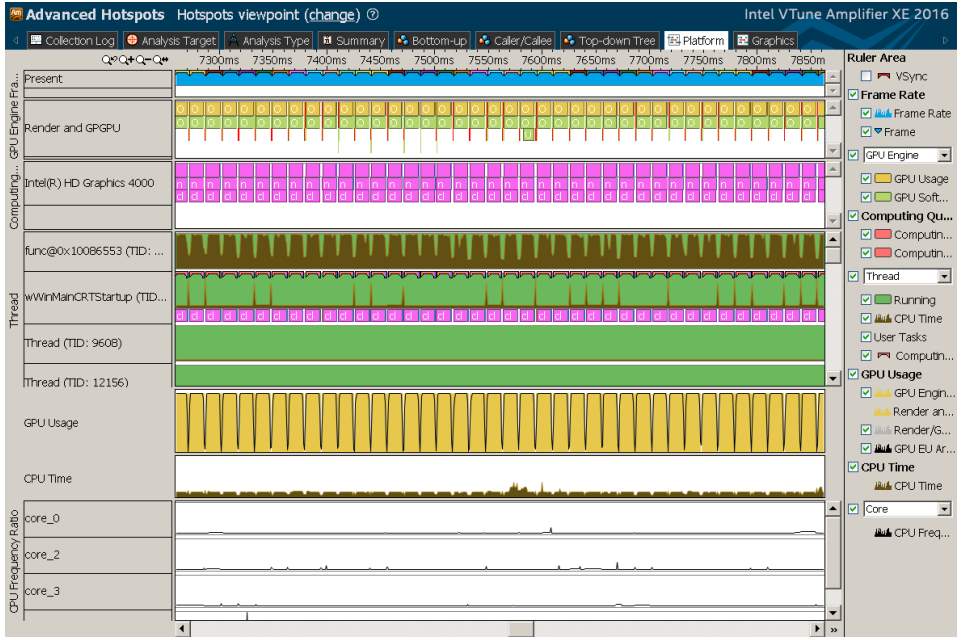
Pick and choose what you need for your audience.

Platform Tab Replaces Tasks and Frames

New Information Available

Includes:

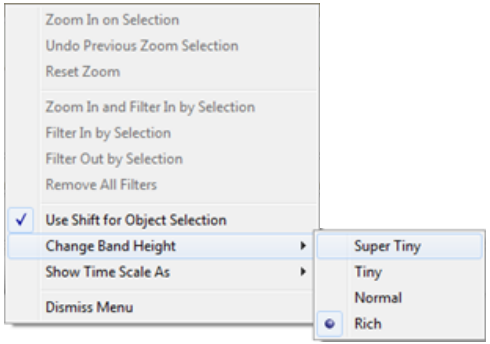
- GPU Usage/Queue
- Bandwidth
- CPU Freq ratio
- depends on collection options

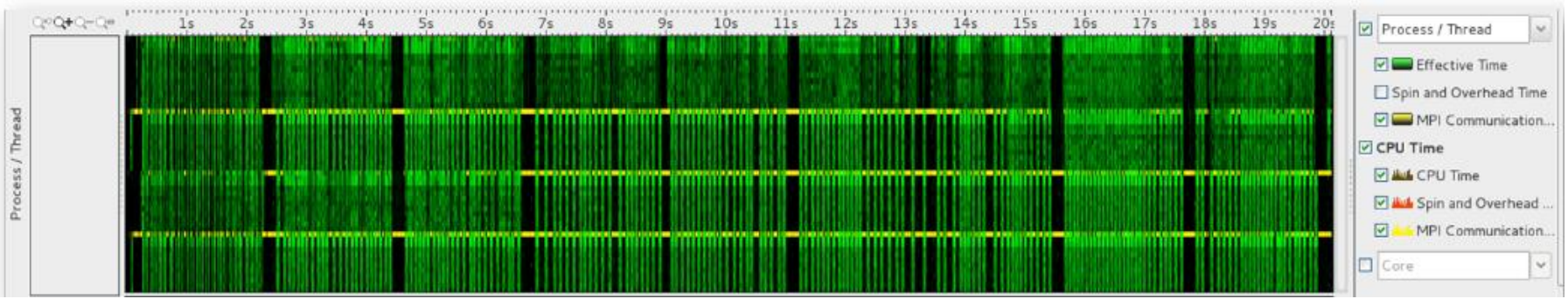


More “Big Picture” Support

“Super Tiny” bird’s-eye view

- Helps recognize application phases and behavioral patterns
- Use context menu to select





Better Linux* Support

Linux ‘perf’ supported for users without root access

- Linux* operating systems based on kernel 2.6.32 or higher
- That export CPU PMU programming details over /sys/bus/event_source/devices/cpu/format file system

VTune Amplifier hardware-based sampling driver adds:

- Stacks†
- Uncore events
- Multiple, precise events
- New events for the latest processors, even on older operating systems

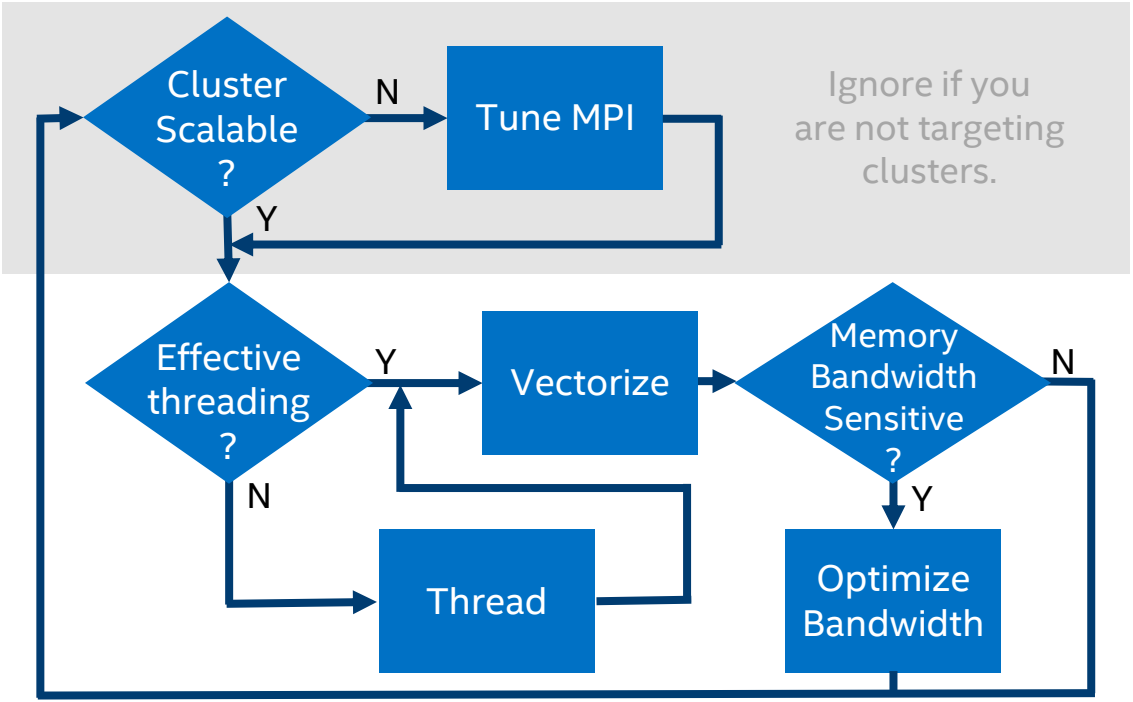
† Newer Linux releases include support for stacks-collection with PMU events

WHICH TOOL SHOULD I USE?

INTEL® VTUNE™ AMPLIFIER? INTEL® ADVISOR? ITAC?

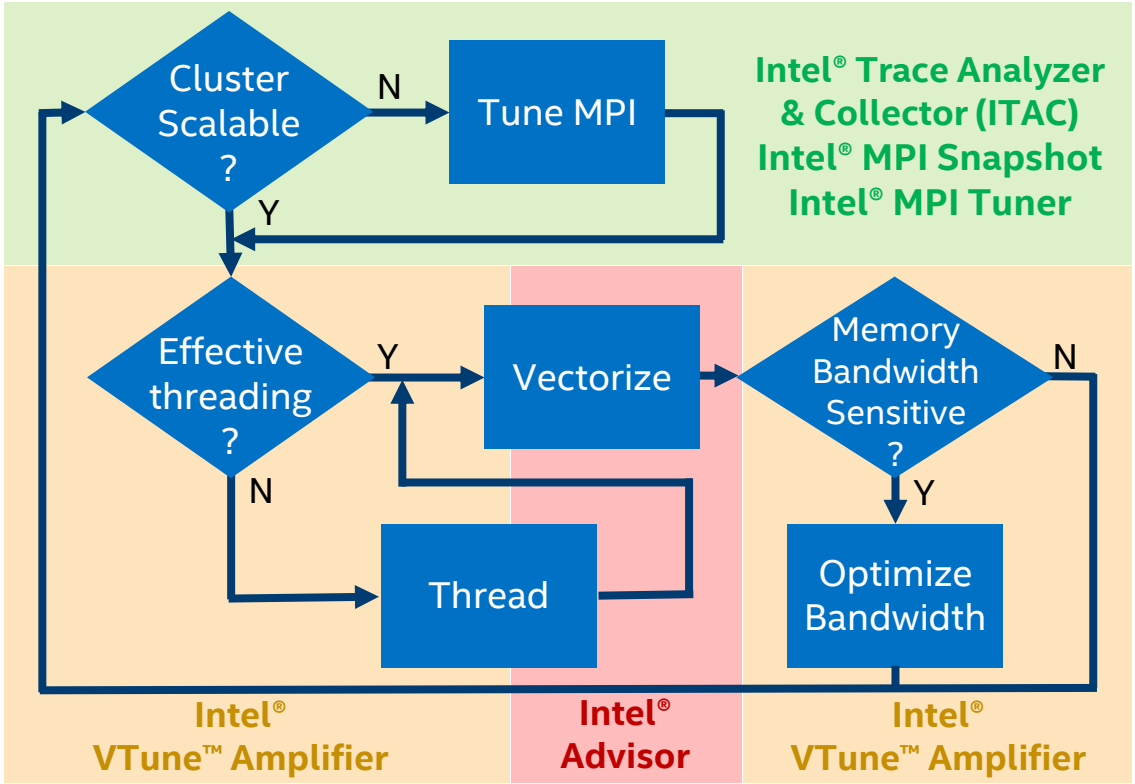
Optimizing Performance On Parallel Hardware

It's an iterative process…



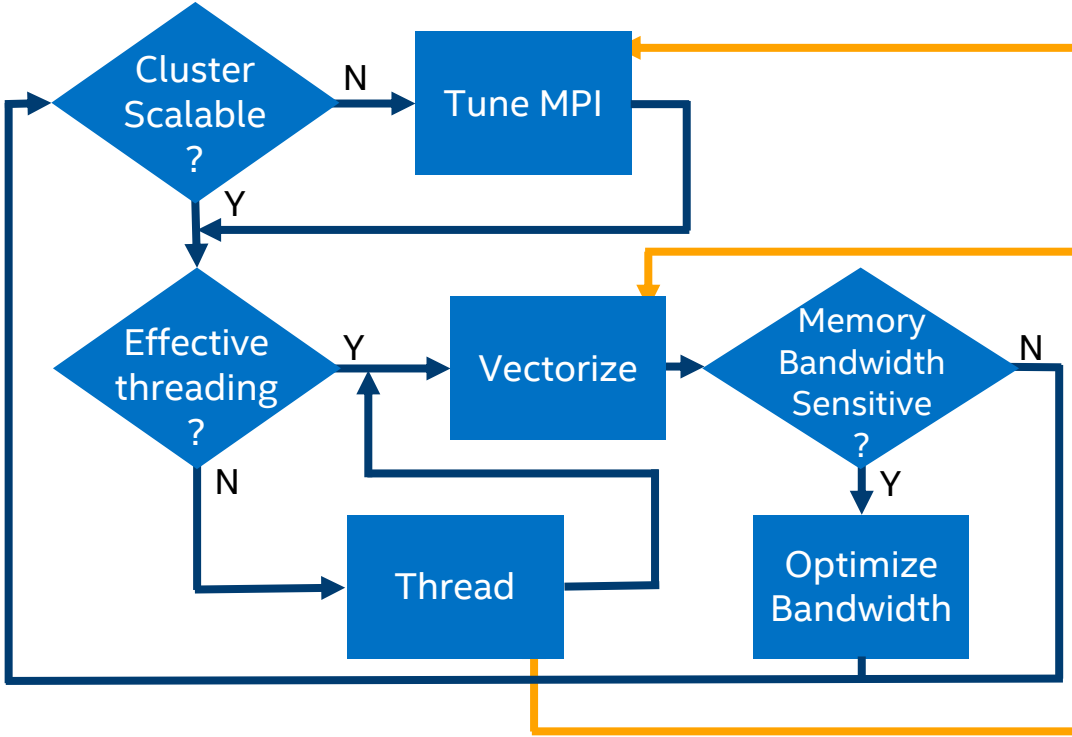
Performance Analysis Tools for Diagnosis

Intel® Parallel Studio XE



Tools for High Performance Implementation

Intel® Parallel Studio XE

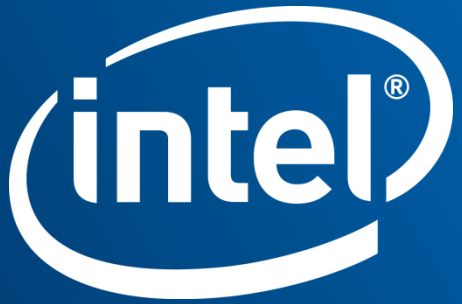


Intel® MPI Library
Intel® MPI Benchmarks

Intel® Compiler

Intel® Math Kernel Library
Intel® IPP – Media & Data Library
Intel® Data Analytics Library
Intel® Cilk™ Plus
Intel® OpenMP*

Intel® TBB – Threading Library


intel®

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.

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.

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