Software

INTEL® VTUNE™ AMPLIFIER

PERFORMANCE PROFILER

Intel® Parallel Studio XE

Profiling, Analysis, and Architecture

Intel® Inspector
Memory and Threading Checking

Intel® VTune™ Amplifier
Performance Profiler

Intel® Advisor
Vectorization Optimization and Thread Prototyping

Cluster Tools

Intel® Cluster Checker
Cluster Diagnostic Expert System

Intel® Trace Analyzer and Collector
MPI Profiler

Performance Libraries

Intel® Data Analytics Acceleration Library
Optimized for Data Analytics & Machine Learning

Intel® Math Kernel Library
Optimized Routines for Science, Engineering, and Financial

Intel® MPI Library

Intel® Integrated Performance Primitives
Image, Signal, and Compression Routines

Intel® Threading Building Blocks
Task-Based Parallel C++ Template Library

Intel® C/C++ and Fortran Compilers

Intel® Distribution for Python
Performance Scripting

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Intel® Parallel Studio XE

Profiling, Analysis, and Architecture

- **Intel® Inspector** — Memory and Threading Checking
- **Intel® VTune™ Amplifier** — Performance Profiler
- **Intel® Advisor** — Vectorization Optimization and Thread Prototyping

Cluster Tools

- **Intel® Cluster Checker** — Cluster Diagnostic Expert System
- **Intel® Trace Analyzer and Collector** — MPI Profiler
- **Intel® MPI Library**

Performance Libraries

- **Intel® Data Analytics Acceleration Library** — Optimized for Data Analytics & Machine Learning
- **Intel® Math Kernel Library** — Optimized Routines for Science, Engineering, and Financial
- **Intel® Integrated Performance Primitives** — Image, Signal, and Compression Routines
- **Intel® Threading Building Blocks** — Task-Based Parallel C++ Template Library

Intel® C/C++ and Fortran Compilers

Intel® Distribution for Python — Performance Scripting

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Faster, Scalable Code, Faster

Intel® VTune™ Amplifier Performance Profiler

Accurate Data - Low Overhead

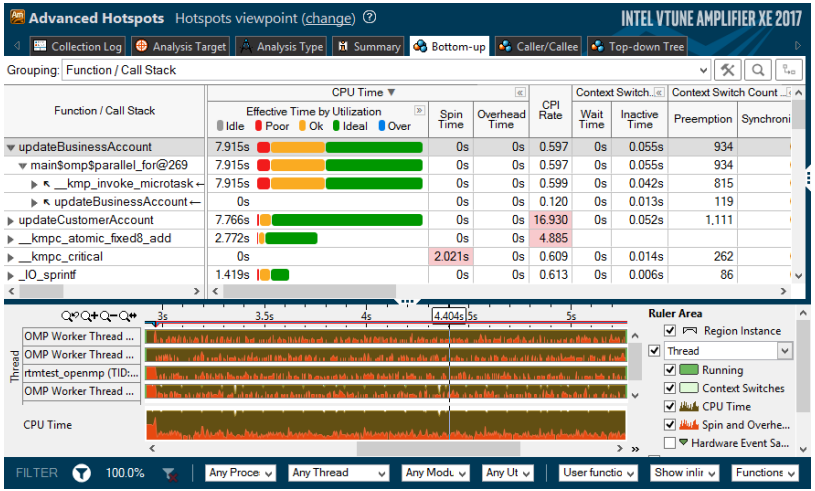
- CPU, GPU, FPU, threading, bandwidth…

Meaningful Analysis

- Threading, OpenMP region efficiency
- Memory access, storage device

Easy

- Data displayed on the source code
- Easy set-up, no special compiles



"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

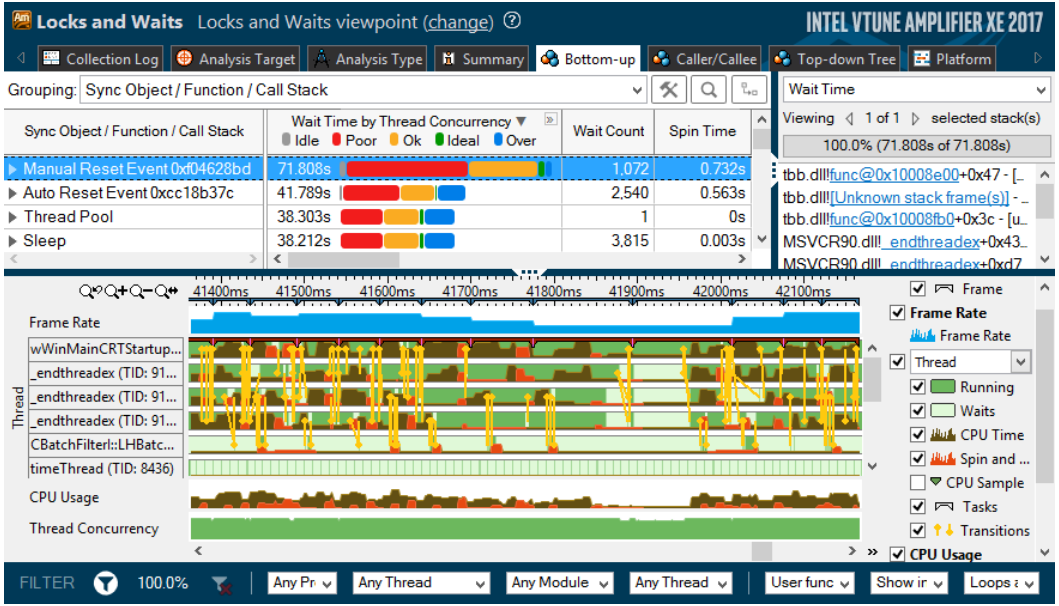
Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

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
- Advanced Features
- What's New
- Summary



Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

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

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

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** Memory-access, HPC Characterization, etc…
Any IA86 processor, any VM, no driver	Higher res., lower overhead, system wide

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

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Example: Hotspots Analysis

Summary View

General Exploration Hotspots viewpoint (change)

Collection Log | Analysis Target | Analysis Type | Summary | Bottom-up

Elapsed Time: 5.554s

CPU Time:	10.504s
Instructions Retired:	21,698,000,000
CPI Rate:	1.257
CPU Frequency Ratio:	1.041
Total Thread Count:	9
Paused Time:	0s

Top Hotspots

This section lists the most active functions in your application. Optimizing these hotspot functions typically results in improving overall application performance.

Function	Module	CPU Time
grid_intersect	3_tachyon_omp.exe	5.539s
sphere_intersect	3_tachyon_omp.exe	3.247s
func@0x1002e59d	libiomp5md.dll	0.148s
shader	3_tachyon_omp.exe	0.117s
KeDelayExecutionThread	ntoskrnl.exe	0.091s
[Others]	N/A*	1.361s

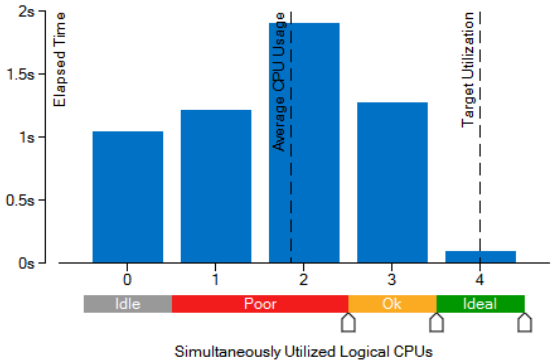
*N/A is applied to non-summable metrics.

Average Bandwidth

Package	Total, GB/sec	Read, GB/sec	Write, GB/sec
package_0	5.715	3.504	2.212

CPU Usage Histogram

This histogram displays a percentage of the wall time the specific number of CPUs were running simultaneously. Spin and Overhead time adds to the Idle CPU usage value.



Collection and Platform Info

This section provides information about this collection, including result set size and collection platform data.

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Example: Concurrency Analysis

Bottom-up View

Concurrency Hotspots by CPU Usage viewpoint (change)

Intel VTune Amplifier XE

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

Grouping: Function / Call Stack

Function / Call Stack	CPU Time by Utilization (Idle, Poor, Ok, Ideal, Over)	Wait Time by Utilization (Idle, Poor, Ok, Ideal, Over)	Ov. an.	Thre... Ove...	Module	Start Address	Function (Full)
grid_intersect	5.360s		0s	4.527s	3_tachyon_omp.exe	0x40c7f0	grid_intersect
sphere_intersect	3.542s		0s	2.914s	3_tachyon_omp.exe	0x40aca0	sphere_intersect
SwitchToThread	0.986s		0.986s	0.901s	KERNELBASE.dll	0x10047e49	SwitchToThread
__kmp_launch_thread	0.874s	2.104s	0.874s	0.008s	libiomp5md.dll	0x1004b0d0	__kmp_launch_thread
grid_bounds_intersect	0.297s		0s	0.215s	3_tachyon_omp.exe	0x40c4f0	grid_bounds_intersect
shader	0.106s		0s	0.066s	3_tachyon_omp.exe	0x406b50	shader(struct ray *)
GdipDrawImagePointRectI	0.098s		0s	0.098s	gdiplus.dll	0x10060336	GdipDrawImagePointRectI
pos2grid	0.090s		0s	0.074s	3_tachyon_omp.exe	0x40c410	pos2grid
Selected 1 row(s):	5.360s		0s	4.527s			

Data Of Interest (CPU Metrics)

Viewing 1 of 21 selected stack(s)

31.8% (1.703s of 5.360s)

3_tachyon_omp.exe!grid_intersect - grid...
3_tachyon_omp.exe!intersect_objects+0x...
3_tachyon_omp.exe!shader+0x357 - sha...
3_tachyon_omp.exe!trace+0x2f - trace_re...
3_tachyon_omp.exe!render_one_pixel+0...
3_tachyon_omp.exe!thread_trace+0x2c9...
libiomp5md.dll!__kmp_invoke_microta...
libiomp5md.dll![OpenMP dispatcher]+...

0.5s 1s 1.5s 2s 2.5s 3s 3.5s 4s 4.5s 5s 5.5s 6s

Thread: OMP Worker Thread #1, OMP Worker Thread #2, thread_video (TID: 5712, OMP Worker Thread #3, WinMainCRTStartup (TI, func@0x100097fe (TID:

CPU Usage

Thread Concurrency

Ruler Area: Region I... Thread: Running, Waits, CPU Time, Overhea..., CPU Sample, Transitions; CPU Usage: CPU Time, Overhea...

No filters are applied. | Any Process | Any Thread | Any Module | Utilization: Any Utilization | User functions + 1 | Inline Mode: on | Functions only

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Example: General Exploration

Bottom-up View

General Exploration General Exploration viewpoint (change) INTEL VTUNE AMPLIFIER XE 2017

Collection Log | Analysis Target | Analysis Type | Summary | Bottom-up | Event Count | Platform

Grouping: Function / Call Stack

Function / Call Stack	Clockticks	Instructions Retired	CPI Rate	Front-End Bound	Bad Speculation		Back-End Bound		Retiring	M
					Branch Mispredi...	Machine Clears	Memory Bound	Core Bound		
grid_intersect	14,293,700,000	11,109,300,000	1.287	7.1%	4.2%	0.4%	27.0%	42.3%	19.0%	3_tachyo
sphere_intersect	8,569,000,000	7,885,000,000	1.087	6.2%	0.0%	7.4%	18.2%	45.8%	22.4%	3_tachyo
func@0x1002e59d	374,300,000	311,600,000	1.201	5.1%	0.0%	15.2%	0.0%	28.9%	50.8%	libiomp5r
shader	271,700,000	195,700,000	1.388	10.5%	0.0%	0.0%	31.6%	47.4%	14.0%	3_tachyo
KeDelayExecutionThread	220,400,000	110,200,000	2.000	17.2%	0.0%	17.2%	0.0%	31.0%	34.5%	ntoskrnl.e
func@0x1401514f0	212,800,000	26,600,000	8.000	31.3%	0.0%	22.3%	0.0%	19.6%	26.8%	ntoskrnl.e
tri_intersect	195,700,000	210,900,000	0.928	4.9%	0.0%	9.7%	0.0%	61.2%	24.3%	3_tachyo
__kmp_x86_pause	144,400,000	43,700,000	3.304	19.7%	0.0%	0.0%	0.0%	34.2%	72.4%	libiomp5r
[wow64cpu.dll]	104,500,000	41,800,000	2.500	18.2%	0.0%	0.0%	0.0%	63.6%	54.5%	wow64cp
light_intersect	87,400,000	57,000,000	1.533	10.9%	0.0%	10.9%	0.0%	67.4%	10.9%	3_tachyo
func@0x10013010	81,700,000	247,000,000	0.331	0.0%	0.0%	0.0%	32.6%	32.6%	46.5%	gdiplus.d
VNorm	66,500,000	34,200,000	1.944	14.3%	0.0%	14.3%	28.6%	28.6%	14.3%	3_tachyo
func@0x10009c00	58,900,000	125,400,000	0.470	0.0%	0.0%	16.1%	0.0%	83.9%	0.0%	gdiplus.d
SleepEx	58,900,000	17,100,000	3.444	0.0%	0.0%	16.1%	0.0%	51.6%	32.3%	kernelbas
__libm_sse2_pow	55,100,000	26,600,000	2.071	17.2%	0.0%	17.2%	0.0%	65.5%	0.0%	libmmd.d
func@0x140159900	55,100,000	11,400,000	4.833	0.0%	0.0%	17.2%	0.0%	65.5%	17.2%	ntoskrnl.e

Find real CPU stalls due to cache misses, instruction fetch misses, branch misprediction, and a lot more.

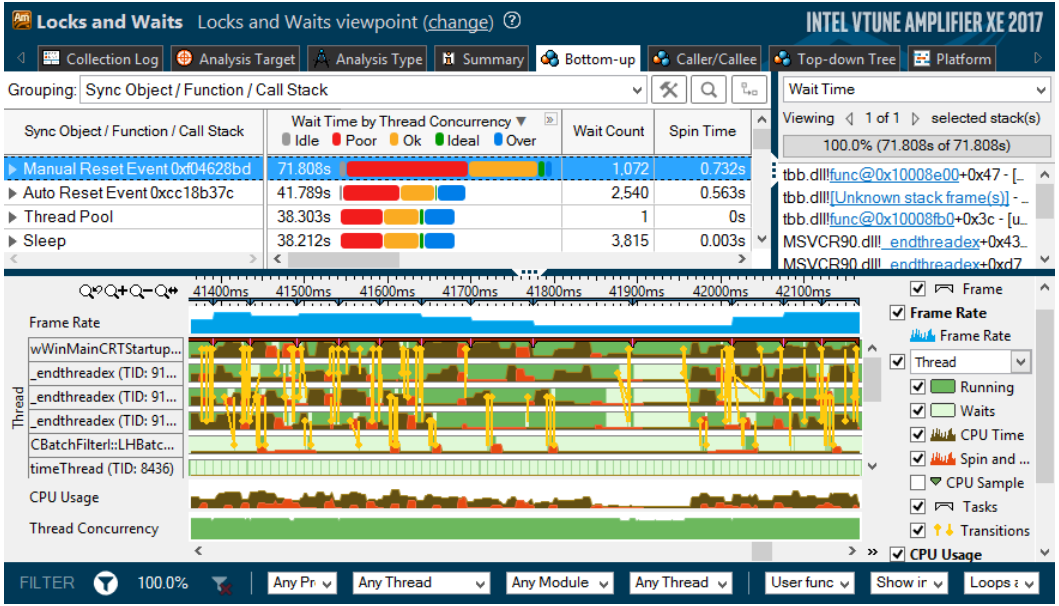
Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

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
- Advanced Features
- What's New
- Summary



Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

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

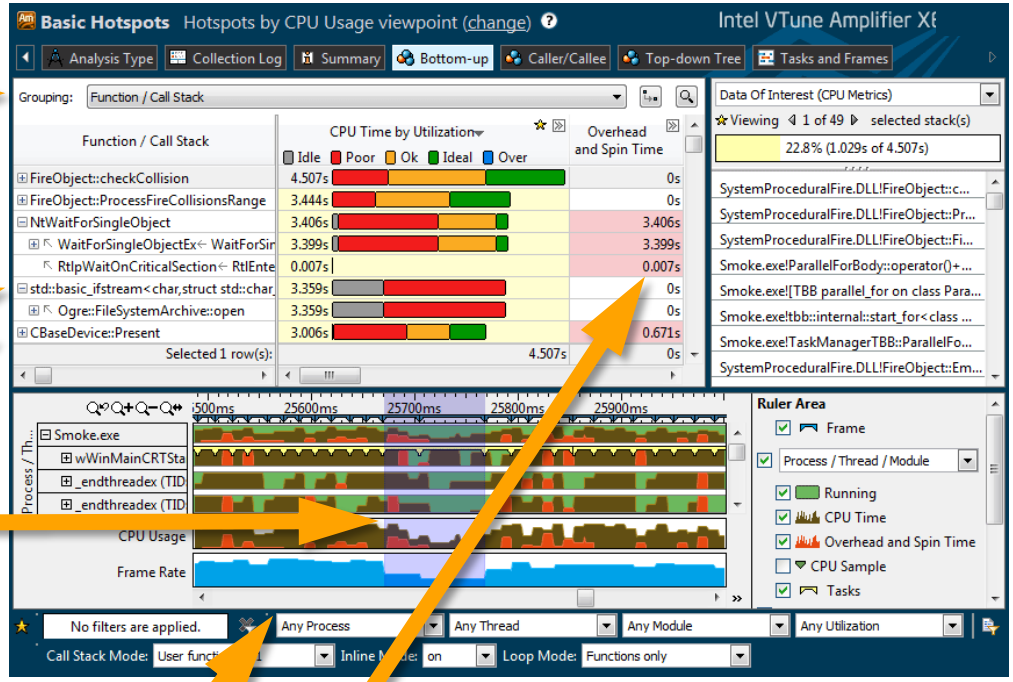
... (Partial list shown)

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

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

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

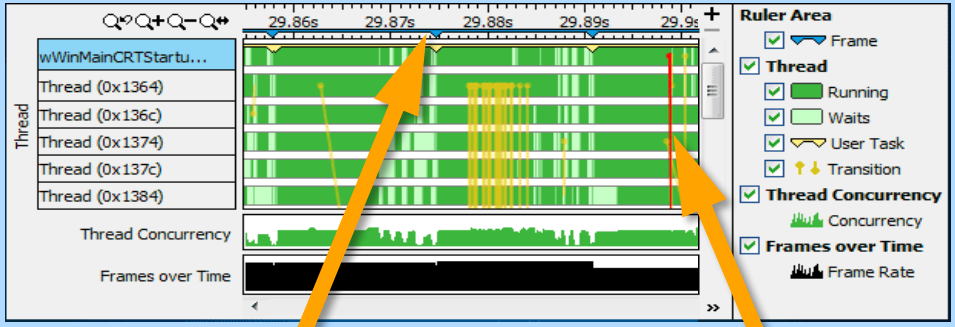
Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Timeline Visualizes Thread Behavior

Intel® VTune™ Amplifier

Transitions

Locks & Waits



Hovers:

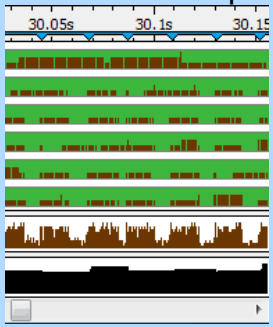
Frame
Frame
Start: 29.858s Duration: 0.017s
Frame: 72
Frame Domain: Smoke::Framework::execute()
Frame Type: Good
Frame Rate: 59.8242179

Transition
Transition
wWinMainCRTStartup (0x12d4) to Thread (0x138c) (29.899s to 29.899s)
Sync Object: TBB Scheduler
Object Creation File: taskmanagertbb.cpp
Object Creation Line: 318

CPU Time

Basic Hotspots

Advanced Hotspots



User Task
User Task
Start: 29.958s Duration: 0.018s
Task Type: Smoke::FrameWork::execute()::Other
Task End Call Stack: Framework::Execute

CPU Time
94.233472%

Optional: Use API to mark frames and user tasks

Optional: Add a mark during collection

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

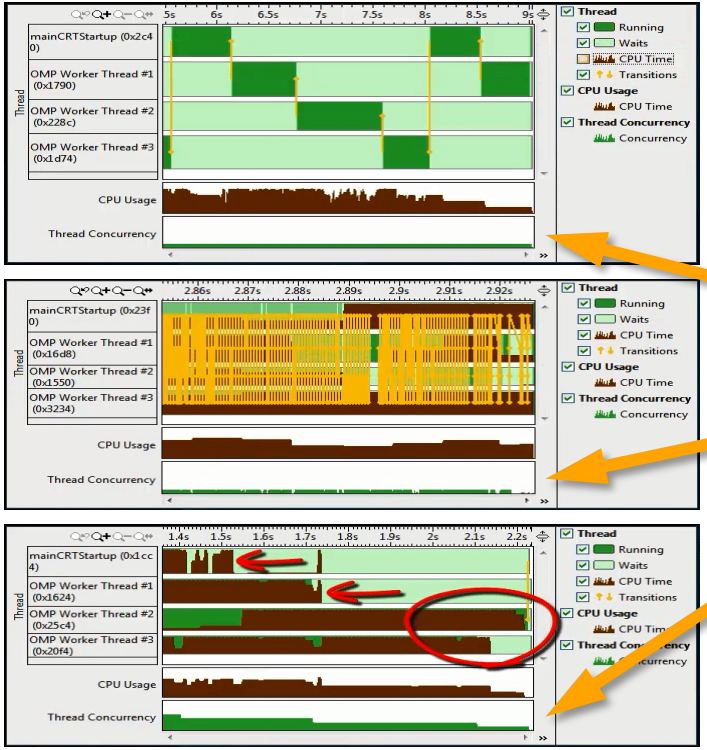
Visualize Parallel Performance Issues

Look for Common Patterns

Coarse Grain Locks

High Lock Contention

Load Imbalance



Low Concurrency

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Tune OpenMP for Efficiency and Scalability

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

OpenMP Analysis. Collection Time: 14.490

1) ▶ Serial Time (outside any parallel region): 4.020s (27.7%)

Serial Time of your application is high. It directly impacts application Elapsed Time and scalability. Explore options for parallelization, algorithm or microarchitecture tuning of the serial part of the application.

Parallel Region Time: 10.469s (72.3%)

Estimated Ideal Time: 7.115s (49.1%)

2) ▶ Potential Gain: 3.354s (23.1%)

The time wasted on load imbalance or parallel work arrangement is significant and negatively impacts the application performance and scalability. Explore OpenMP regions with the highest metric values. Make sure the workload of the regions is enough and the loop schedule is..

3) ▶ Top OpenMP Regions by Potential Gain

This section lists OpenMP regions with the highest potential for performance improvement. The Potential Gain metric shows the elapsed time that could be saved if the region was optimized to have no load imbalance assuming no runtime overhead.

OpenMP Region	Potential Gain	(%)	Elapsed Time
4) ▶ conj_grad_ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:514:695	3.294s	22.7%	10.208s
MAIN__ompparallel:24@/home/vtune/work/apps/NPB/NPB3.3.1/NPB3.3-OMP/CG/cg.f:185:231	0.059s	0.4%	0.260s

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)

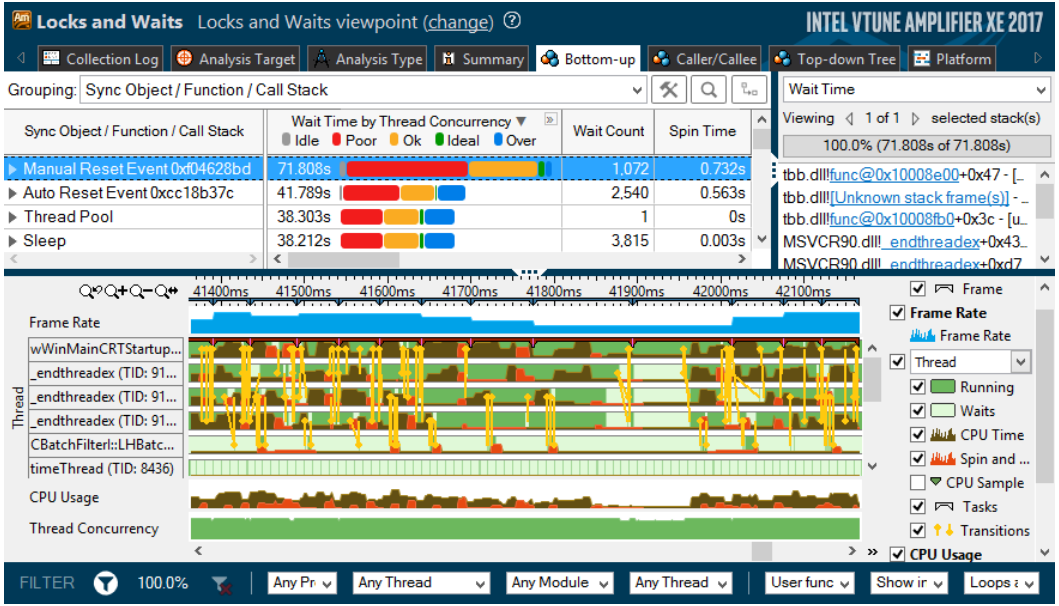
Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

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
- Advanced Features
- What's New
- Summary



Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Command Line Interface

Automate analysis

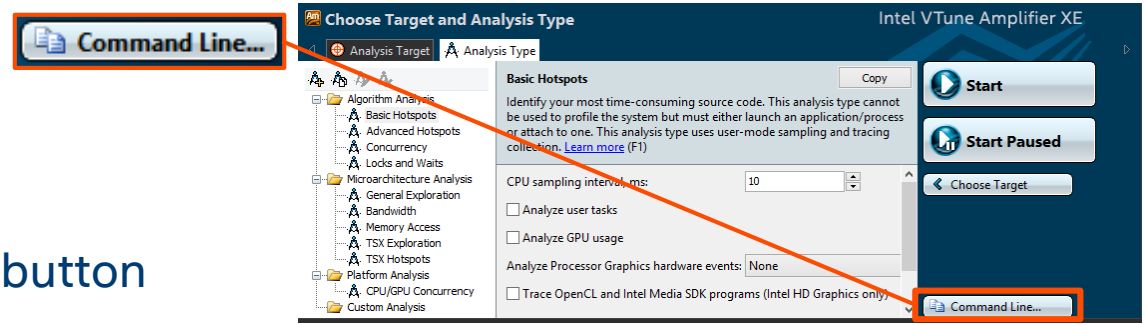
amplxe-cl is the command line:

- **Windows:** `C:\Program Files (x86)\Intel\VTune Amplifier XE \bin[32|64]\amplxe-cl.exe`
- **Linux:** `/opt/intel/vtune_amplifier_xe/bin[32|64]/amplxe-cl`

Help: `amplxe-cl -help`

Use UI to setup

1) Configure analysis in UI
2) Press “Command Line…” button
3) Copy & paste command



Great for regression analysis – send results file to developer
Command line results can also be opened in the UI

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

MPI Analysis

Command line:

```
> mpirun -n 16 -ppn 4 -l amplxe-cl -collect advanced-hotspots -trace-mpi -result-dir my_result -- my_app.a
```

Or use gtool:

```
> mpirun -gtool "amplxe-cl -collect memory-access -result-dir my_result:7,5" my_app.a
```

Each process data is presented for each node they were running on:

```
my_result.host_name1 (rank 0-3)
my_result.host_name2 (rank 4-7)
my_result.host_name3 (rank 8-11)
my_result.host_name4 (rank 12-15)
```

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Interactive Remote Data Collection

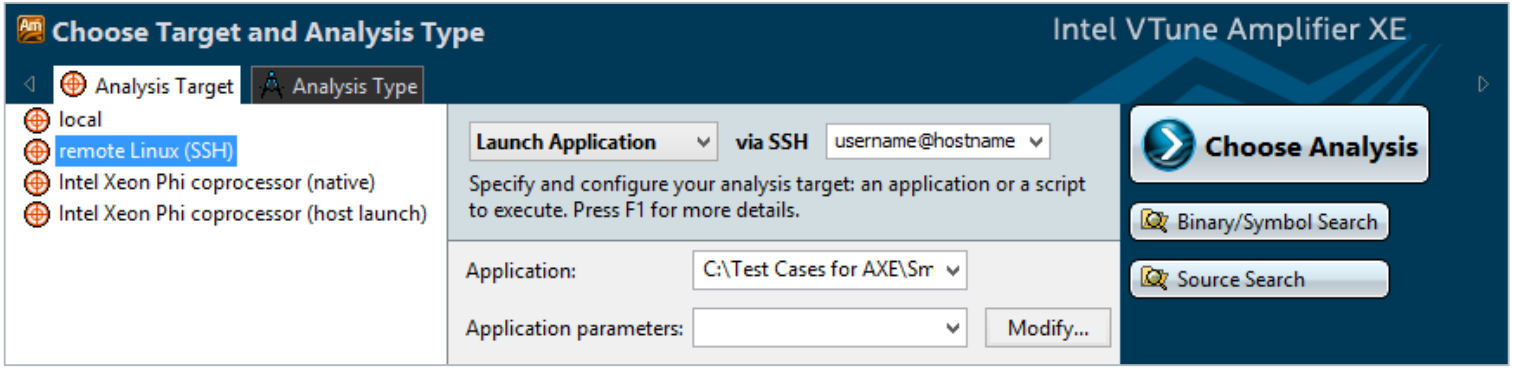
Performance analysis of remote systems just got a lot easier

Interactive analysis

1) Configure SSH to a remote Linux* target
2) Choose and run analysis with the UI

Command line analysis

1) Run command line remotely on Windows* or Linux* target
2) Copy results back to host and open in UI



Conveniently use your local UI to analyze remote systems

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Compare Results Quickly - Sort By Difference

Intel® VTune™ Amplifier

Quickly identify cause of regressions.

- Run a command line analysis daily
- Identify the function responsible so you know who to alert

Compare 2 optimizations – What improved?

Compare 2 systems – What didn't speed up as much?

Grouping: Function / Call Stack

Function / Call Stack	CPU Time:Difference	Module	CPU Time:r007hs	CPU Time:r006hs
FireObject::checkCollision	4.850s	SystemProceduralFire.DLL	6.281s	1.431s
FireObject::ProcessFireCollisionsRange	4.644s	SystemProceduralFire.DLL	5.643s	0.999s
dllStopPlugin	3.765s	RenderSystem_Direct3D9.DLL	9.184s	5.419s

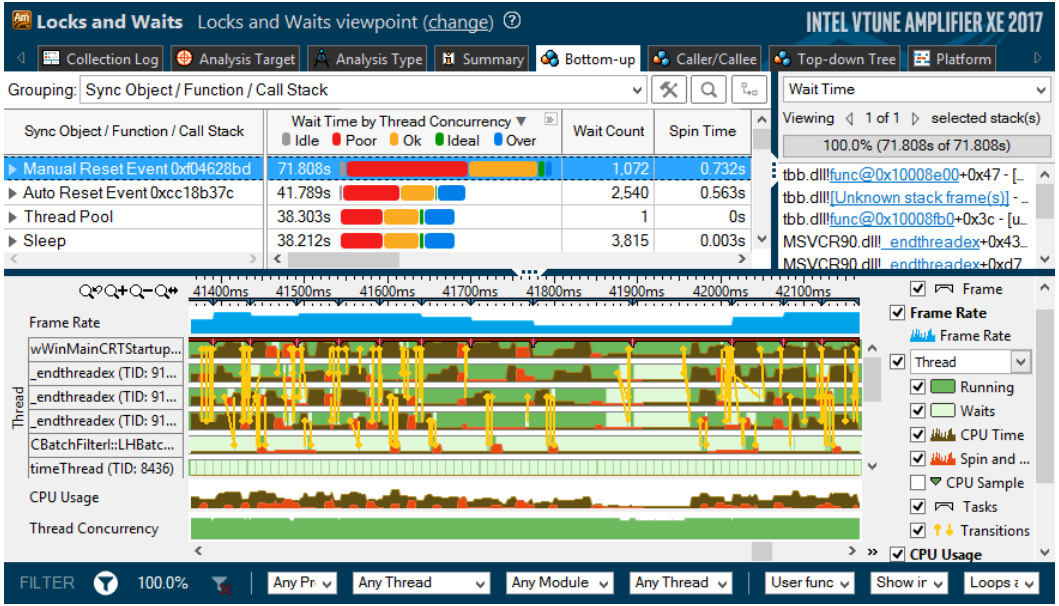
Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

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
- Advanced Features
- What's New
- Summary



Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Java Analysis

- Multiple simultaneous JVMs
- Sampling is fast / unobtrusive
- Mixed Java / C++ / Fortran
- See results on the Java source

Memory Access - Hardware Issues

Analysis Target | Analysis Type | Collection Log | Summary | Bottom-up | Top-down

Grouping: Function / Call Stack

Function / Call Stack	Hardware Event Count		CPI Rate	LLC Miss
	CPU_CLK_U ...	INST_RETIRED.ANY		
jnt::scimark2::SparseCompRow::matmult	13,430,000,000	19,360,000,000	0.694	
jnt::scimark2::FFT::transform_internal	11,520,000,000	5,362,000,000	2.148	
jnt::scimark2::SOR::execute	10,998,000,000	16,324,000,000	0.674	
jnt::scimark2::LU::factor	9,422,000,000	15,072,000,000	0.625	
jnt::scimark2::Random::nextDouble	8,426,000,000	5,056,000,000		
jnt::scimark2::FFT::bitreverse	1,362,000,000	368,000,000		
jnt::scimark2::MonteCarlo::integrate	902,000,000	236,000,000		
Selected 1 row(s):	13,430,000,000	19,360,000,000		

Threads: Thread (0x2b8), Thread (0x1b48), Thread (0x13e4), Thread (0x1228)

Hardware Events

Basic Hotspots Hotspots by CPU Usage viewpoint (change)

Collection Log | Analysis Target | Analysis Type | Summary | Bottom-up | Caller/Callee | Top-down Tre

Source | Assembly | Assembly grouping: Address

Source Line	Source	CPU Time: Total	CPU Time: Self
30	{		
31	return consume_time(token) ;		
32	}		
33			
34	JNIEXPORT void JNICALL Java_MixedCall_CallBackJavaFunc		
35	(JNIEnv * JNIEnv, jobject obj, jint token)		
36	{	67.6%	11.810s
37	jclass cls = (*JNIEnv)->GetObjectClass(JNIEnv, obj);		
38	jmethodID mid = (*JNIEnv)->GetMethodID(JNIEnv, cls, "CallJavaFunc",		
39	(*JNIEnv)->CallIntMethod(JNIEnv, obj, mid, token);		
40	}		

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Optimize Memory Access

Memory Access Analysis - Intel® VTune™ Amplifier 2017

Tune data structures for performance

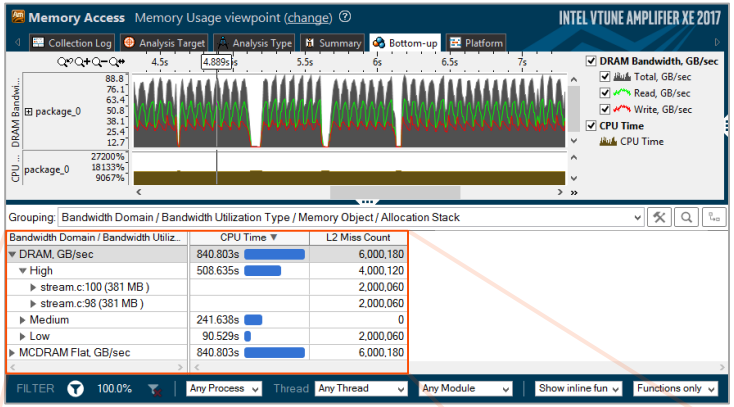
- Attribute cache misses to data structures (not just the code causing the miss)
- Support for custom memory allocators

Optimize NUMA latency & scalability

- True & false sharing optimization
- Auto detect max system bandwidth
- Easier tuning of inter-socket bandwidth

Easier install, Latest processors

- No special drivers required on Linux*
- Intel® Xeon Phi™ processor MCDRAM (high bandwidth memory) analysis



Bandwidth Domain / Bandwidth Utiliz...	CPU Time ▼	L2 Miss Count
▼ DRAM, GB/sec	840.803s	6,000,180
▼ High	508.635s	4,000,120
▶ stream.c:100 (381 MB)		2,000,060
▶ stream.c:98 (381 MB)		2,000,060
▶ Medium	241.638s	0
▶ Low	90.529s	2,000,060
▶ MCDRAM Flat, GB/sec	840.803s	6,000,180

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Storage Device Analysis (HDD, SATA or NVMe SSD)

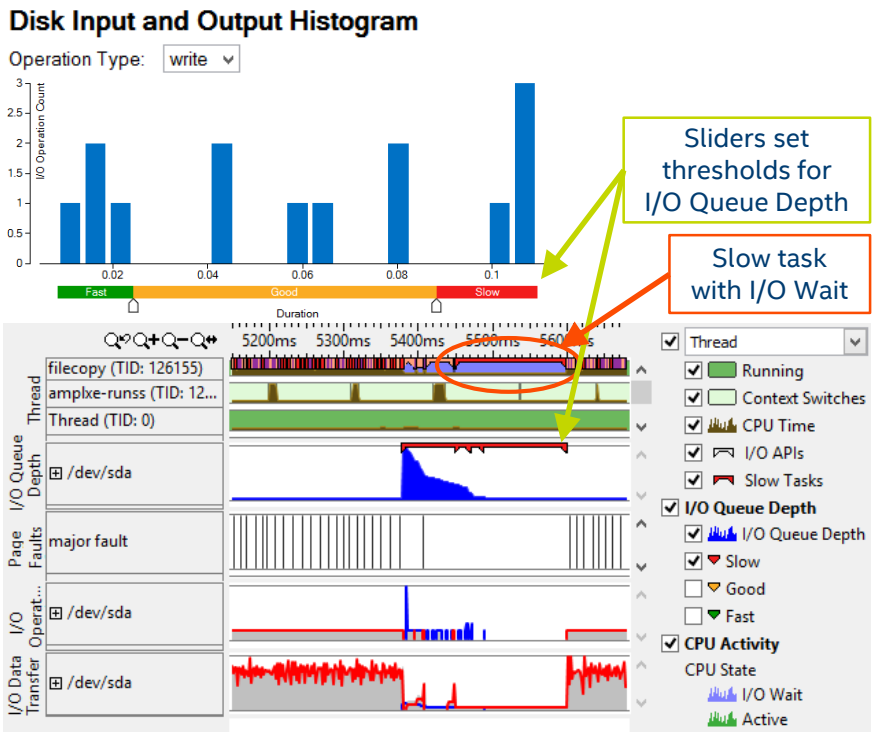
Intel® VTune™ Amplifier

Are You I/O Bound or CPU Bound?

- Explore imbalance between I/O operations (async & sync) and compute
- Storage accesses mapped to the source code
- See when CPU is waiting for I/O
- Measure bus bandwidth to storage

Latency analysis

- Tune storage accesses with latency histogram
- Distribution of I/O over multiple devices



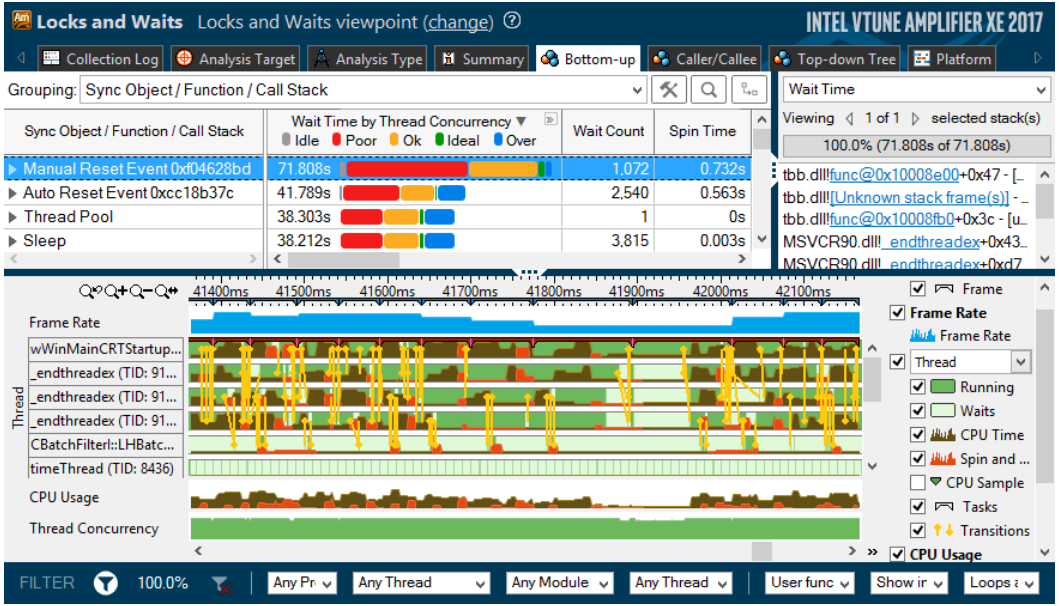
Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

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
- Advanced Features
- What's New
- Summary



Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

BETTER: MPI, SNAPSHOT, CLOUD, PYTHON*, AND MORE

INTEL® VTUNE™ AMPLIFIER 2018 BETA

Latest Hardware

- Support for Intel® Xeon Phi™ processor (codenamed Knights Landing)

Better Metrics Easier Tuning

- Better Application Snapshot: Merges MPI + Application data
- Enhanced metrics for MPI applications: Imbalance & critical path rank

Container Support

- Profile inside Docker* & Mesos* containers
- Attach to Running Java* services & daemons

Enhanced Profiling

- Locks & waits analysis for mixed Python* and native code
- Memory consumption analysis for Python, C/C++

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

APPLICATION PERFORMANCE SNAPSHOT

EASY PERFORMANCE SNAPSHOT FOR THREADED MPI APPLICATIONS

Quick & easy performance overview

- Does the app need performance tuning?

MPI and non-MPI Apps

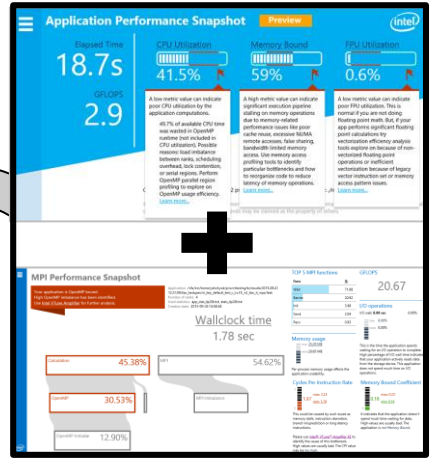
- Distributed MPI with or without threading
- Shared memory applications

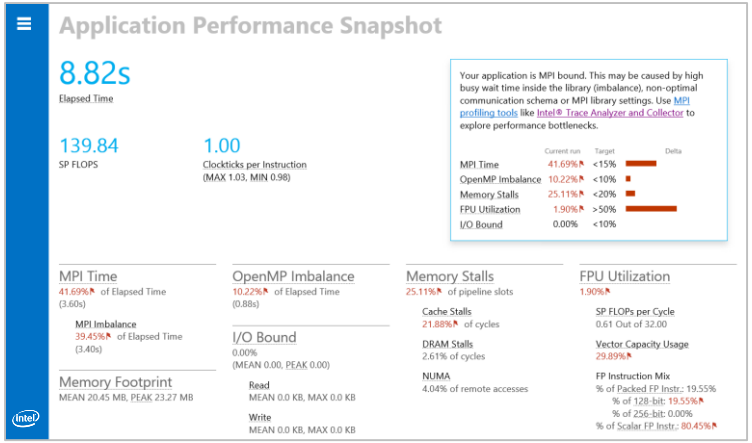
Popular MPI implementations supported

- Intel® MPI
- MPICH and Cray MPI

Richer metrics on computation efficiency

- CPU (processor stalls, memory access)
- FPU (vectorization metrics)





*Linux only

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Optimize Private Cloud-Based Applications

Profile Enterprise Applications

- Native C, C++, Fortran*
- Attach to running Java* services (e.g., Mail)
- Profile Java daemons without restart

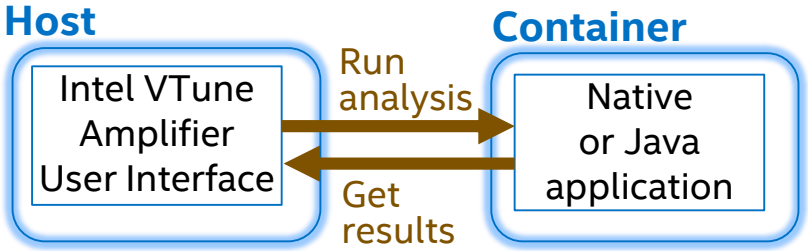
Accurate, Low-Overhead Data Collection

- Advanced hotspots and hardware events
- Memory analysis
- Accurate stack information for Java and HHVM*

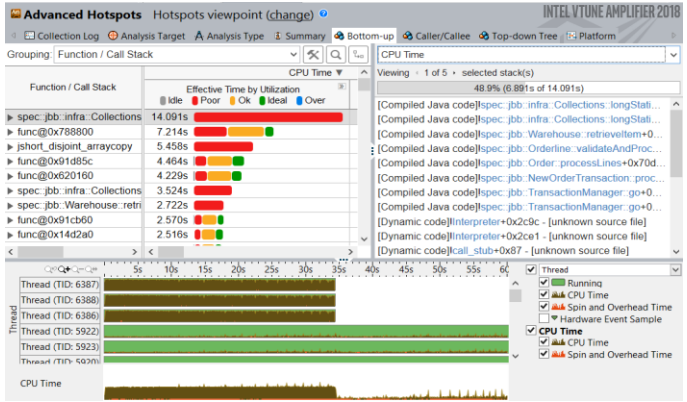
Popular Containers Supported

- Docker*
- Mesos*

Software collectors (e.g., locks & waits) and Python* profiling are not currently available for containers.



- No container configuration required
- Detection of the container is automatic



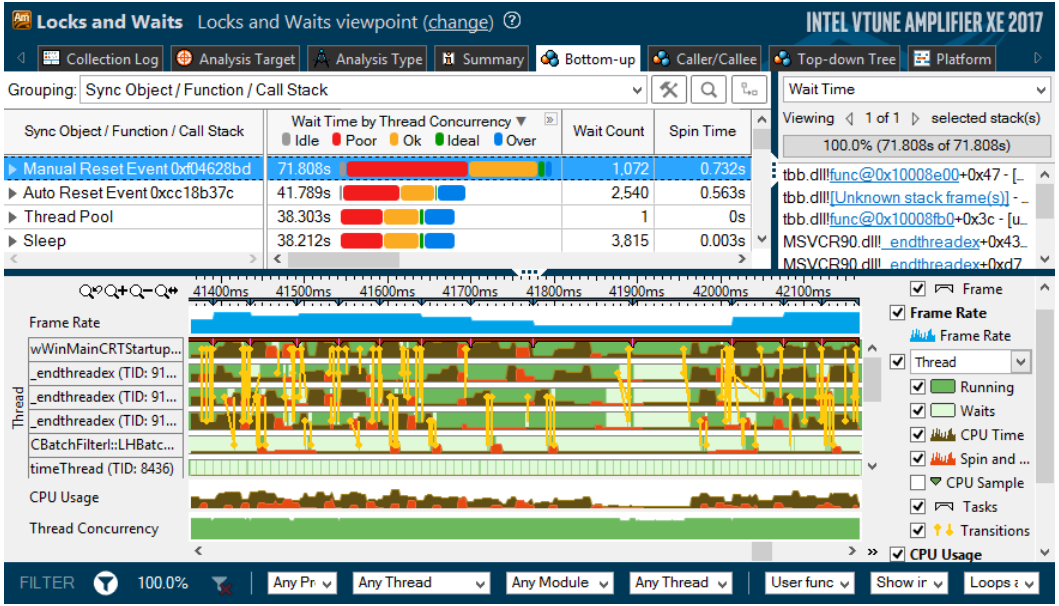
Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

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
- Advanced Features
- What's New
- Summary



Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Intel® VTune™ Amplifier

Faster, Scalable 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

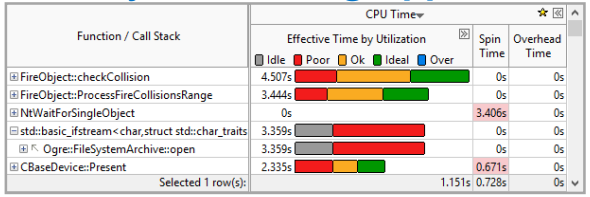
Find Answers Fast

- 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

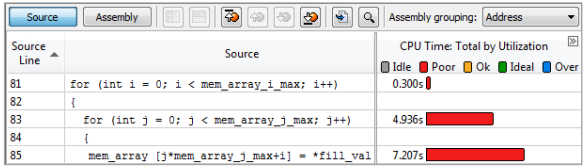
- No Special Compiles – C, C++, C#, Fortran, Java, ASM
- Visual Studio* Integration or Stand Alone
- Local & Remote Data Collection, Command Line
- Analyze Windows* & Linux* data on OS X*[2]

Quickly Find Tuning Opportunities



Function / Call Stack	CPU Time: Effective Time by Utilization (Idle, Poor, Ok, Ideal, Over)	Spin Time	Overhead Time
FireObject::checkCollision	4.507s	0s	0s
FireObject::ProcessFireCollisionsRange	3.444s	0s	0s
NtWaitForSingleObject	0s	3.406s	0s
std::basic_ifstream<char,struct std::char_traits	3.359s	0s	0s
Ogre::FileSystemArchive::open	3.359s	0s	0s
CBaseDevice::Present	2.335s	0.671s	0s
Selected 1 row(s):	1.151s	0.728s	0s

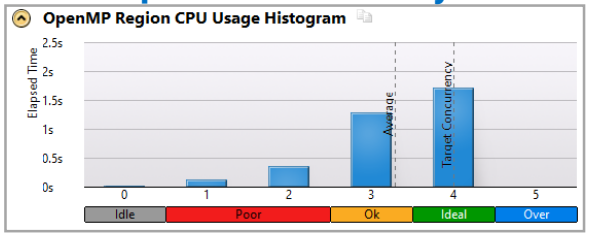
See Results On The Source Code



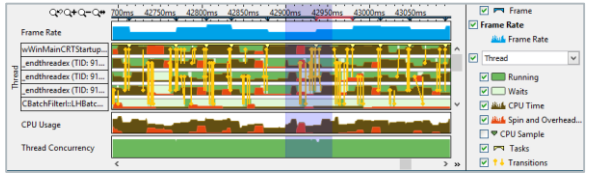
Source | Assembly | Assembly grouping: Address

Source Line	Source	CPU Time: Total by Utilization (Idle, Poor, Ok, Ideal, Over)
81	for (int i = 0; i < mem_array_i_max; i++)	0.300s
82	{	
83	for (int j = 0; j < mem_array_j_max; j++)	4.936s
84	{	
85	mem_array [j*mem_array_j_max+i] = *fill_val	7.207s

Tune OpenMP Scalability



Visualize & Filter Data



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

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Intel® Parallel Studio XE

Profiling, Analysis, and Architecture

Intel® Inspector
Memory and Threading Checking

Intel® VTune™ Amplifier
Performance Profiler

Intel® Advisor
Vectorization Optimization and Thread Prototyping

Cluster Tools

Intel® Cluster Checker
Cluster Diagnostic Expert System

Intel® Trace Analyzer and Collector
MPI Profiler

Performance Libraries

Intel® Data Analytics Acceleration Library
Optimized for Data Analytics & Machine Learning

Intel® Math Kernel Library
Optimized Routines for Science, Engineering, and Financial

Intel® MPI Library

Intel® Integrated Performance Primitives
Image, Signal, and Compression Routines

Intel® Threading Building Blocks
Task-Based Parallel C++ Template Library

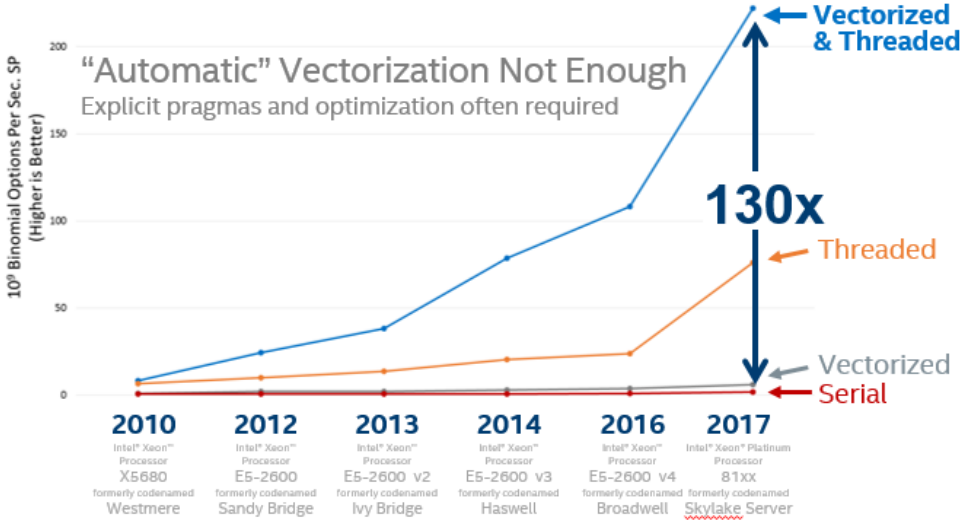
Intel® C/C++ and Fortran Compilers

Intel® Distribution for Python
Performance Scripting

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

Intel® Advisor – Modernize Your Code

Optimize Vectorization & Prototype Threading



The Difference Is Growing with Each New Generation of Hardware

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. For more information go to http://www.intel.com/performance. Configurations for 2007-2016 Benchmarks at the end of this presentation

Modern Performant Code

- Vectorized (uses AVX, AVX-512)
- Efficient memory access
- Threaded

Intel Advisor

- Adds & optimizes vectorization
- Analyzes memory patterns
- Quickly prototypes threading

New for 2018! (partial list)

- Roofline analysis
- Faster data collection
- More recommendations

Learn more: http://intel.ly/advisor-xe

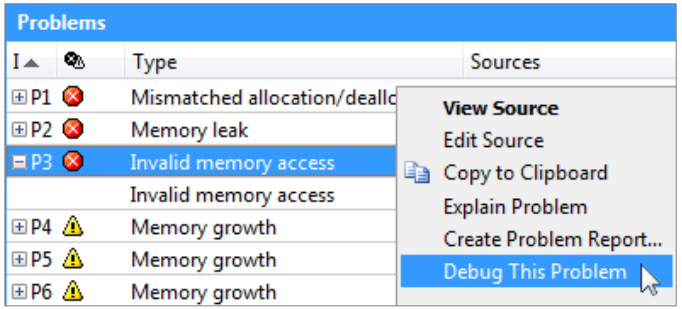
Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

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, & SSSE3 instruction sets & 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

Intel® Inspector – Memory & Thread Debugger

Find & Debug Memory Leaks, Corruption, Data Races, Deadlocks & more

Debugger Breakpoints



Diagnose in hours instead of months

Learn more: http://intel.ly/inspector-xe

Correctness Tools Increase ROI By 12%-21%[1]

- Errors found earlier are less expensive to fix
- Races & deadlocks not easily reproduced
- Memory errors are hard to find without a tool

Debugger Integration Speeds Diagnosis

- Breakpoint set just before the problem
- Examine variables & threads with the debugger

What's new in 2018

- Fewer false positives
- C++ 17 std::shared_mutex added
- Windows SRW Locks added

[1] **Cost Factors – Square Project Analysis**
CERT: U.S. Computer Emergency Readiness Team, and Carnegie Mellon CyLab NIST: National Institute of Standards & Technology : Square Project Results

Roadmap Notice: All information provided here is subject to change without notice.
Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Optimization Notice
Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.

intel®

Software

Legal Disclaimer & Optimization Notice

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. For more complete information visit www.intel.com/benchmarks.

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.

Copyright © 2018, Intel Corporation. All rights reserved. Intel, Pentium, Xeon, Xeon Phi, Core, VTune, Cilk, and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright © 2018, Intel Corporation. All rights reserved.
*Other names and brands may be claimed as the property of others.