



Intel[®] Processor Trace in Intel[®] VTune[™] Amplifier

Stanislav Bratanov; Roman Belenov; Ludmila Pakhomova

4/27/2015

What is Intel Processor Trace?

Intel Processor Trace (Intel PT) provides hardware a means to trace branching, transaction, and timing information in a highly-compressed, low-overhead manner

- Intel PT is to be extended in the future with more tracing information

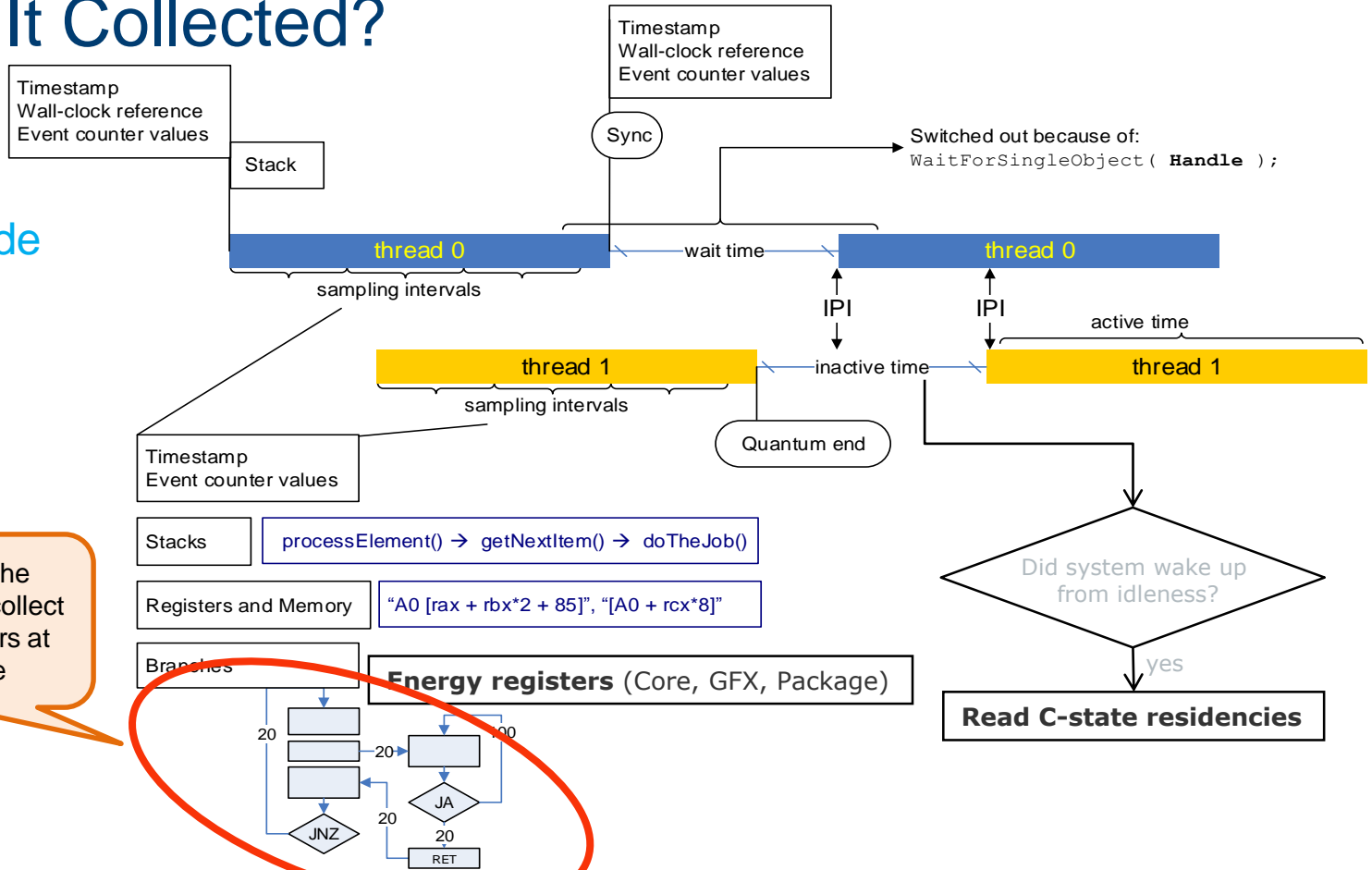
Intel PT differentiates between processes, but not SW threads. Per-thread collection may be necessary

Intel PT saves information on conditional and indirect branches only. The rest to be found by static analysis of disassembly to decode Intel PT data stream correctly

Source Code		Intel PT Log	
IP →	<code>mov %eax, offset BasicBlock</code>	TIP:	BasicBlock
IP →	<code>call %eax</code>	CYC:	8 cycles
IP →	<code>... </code>	TNT:	111110
	<code>BasicBlock:</code>	CYC:	18 cycles
	<code>Loop1:</code>	TNT:	110
IP →	<code>..do stuff..</code>	CYC:	16 cycles
IP →	<code>jnz Loop1</code>	TIP:	CALL NLIP
	<code>Loop2:</code>	CYC:	2 cycles
IP →	<code>..more stuff..</code>		
IP →	<code>iz Loop2</code>		
IP →	<code>ret</code>		
	<code>...</code>		

How Is It Collected?

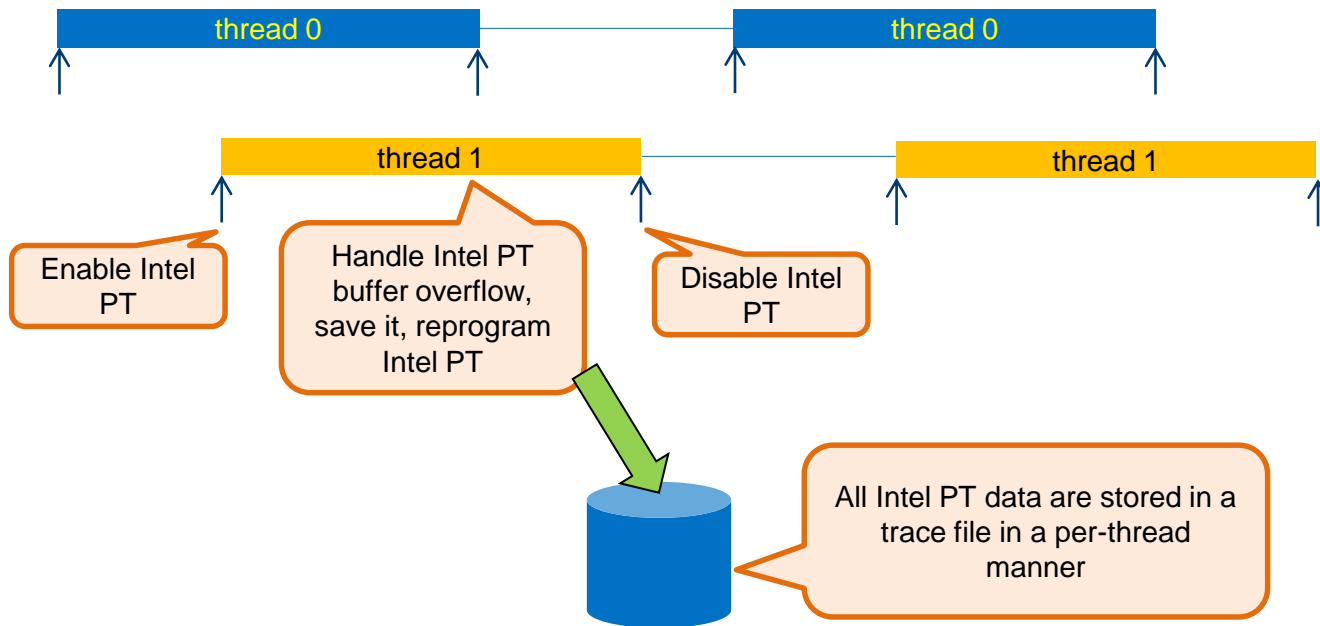
Statistical Mode



Intel PT fits in the current scheme: collect 4k Intel PT buffers at every sample

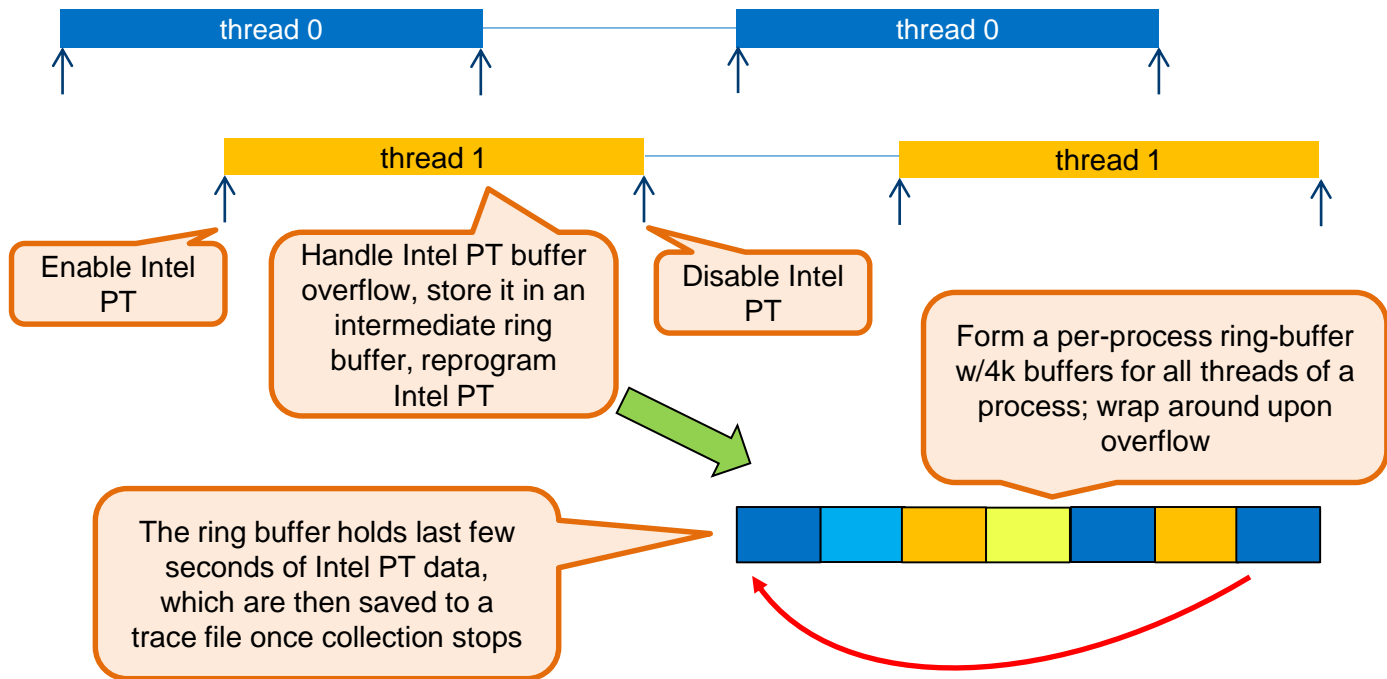
How is It Collected?

Full Trace Mode



How is It Collected?

Ring Buffer Mode



What is It Used for?

Statistical Call Counts and Loop Iteration Counts

Advanced Hotspots Copy

Identify time-consuming code in your application. Advanced Hotspots analysis (formerly, Lightweight Hotspots) uses the OS kernel support or VTune Amplifier kernel driver to extend the Hotspots analysis by collecting call stacks, context switch and statistical call count data as well as analyzing the CPI (Cycles Per Instruction) metric. By default, this analysis uses higher frequency sampling at lower overhead compared to the Basic Hotspots analysis. Press F1 for more details.

CPU sampling interval, ms:

Select a level of details provided with event-based sampling collection. Detailed collection levels cause higher overhead.

- Hotspots
- Hotspots, stacks and context switches
- Hotspots, call counts, stacks and context switches
- Hotspots, call counts, loop trip counts, stacks and context switches

Event mode:

Analyze user tasks

Intel PT is used automatically, if available, when selecting these options of Advanced Hotspots

What is It Used for?

Call Counts and Loops

Function and loop stat metrics made more accurate with Intel PT

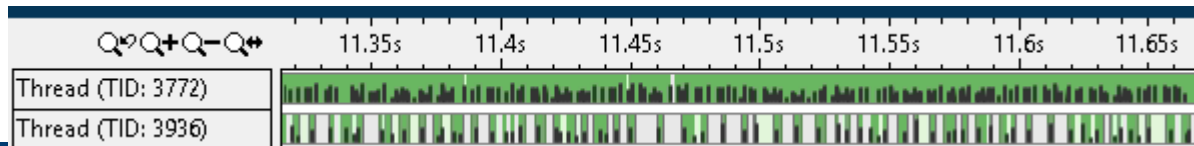
Hardware Event Count: Total by Han

Function Stack	CPU_CLK_UNHALTED...	INST_RETIRED.ANY	CALL_COUNT	ITERATION_COUNT	LOOP_ENTRY_COUNT
mainCRTStartup	32,152,121,993	44,394,297,105	114,297,027	1,730,779,544	70,048,576
main	32,141,906,001	44,383,215,008	114,216,732	1,727,626,469	70,021,343
lame_encoder	32,128,767,182	44,372,830,452	114,173,833	1,725,085,454	70,016,706
[Loop@0x403590 in lame_encoder]	32,128,767,182	44,372,830,452	114,173,833	1,725,085,454	70,016,706
lame_encode_buffer_int	27,869,973,746	38,566,467,791	90,487,943	1,629,166,330	64,316,845
lame_encode_buffer_sample_t	27,696,544,179	38,362,314,162	89,779,807	1,624,460,460	64,238,536
[Loop@0x40a150 in lame_encode_buffer_sam]	27,639,440,208	38,264,463,969	89,734,057	1,619,851,466	64,208,036
lame_encode_frame	23,669,634,269	33,570,163,386	89,443,484	1,503,810,501	63,961,072
AnalyzeSamples	3,692,537,121	4,352,484,890	271,641	96,567,411	216,201
fill_buffer	99,618,022	163,233,675	15,074	19,453,840	15,074

A control flow tree...

...can estimate a cost of a function call or a loop iteration to help make conclusions on necessary optimizations, such as, threading, vectorizing, inlining

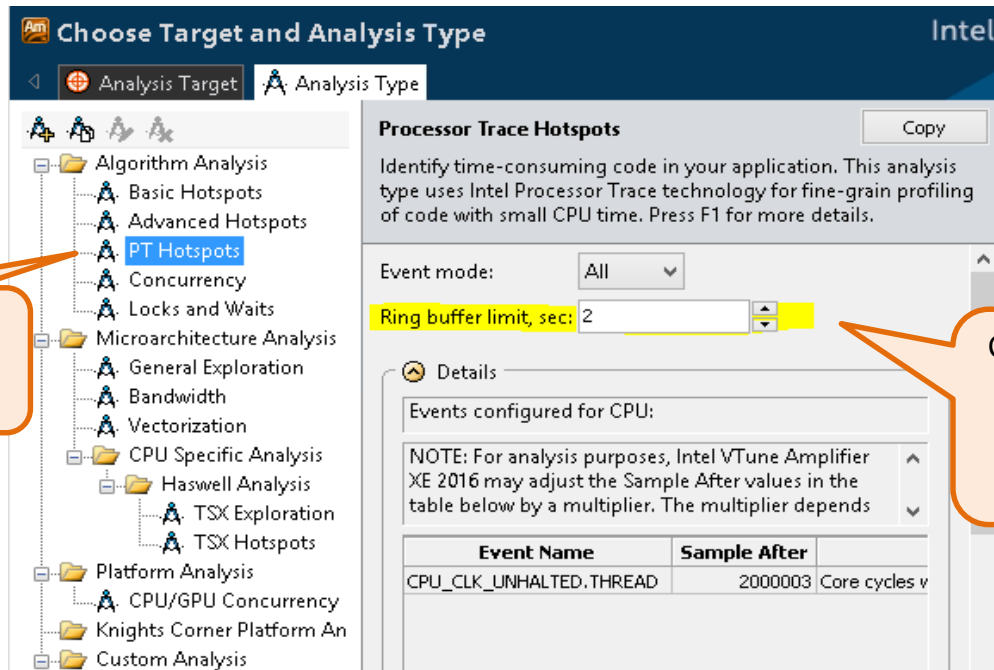
All metrics can be seen on timeline within thread quanta



What is It Used for?

Detailed tracing of an app or a portion thereof

- set `AMPLXE_EXPERIMENTAL=full-intel-pt`



Select PT Hotspots analysis type

Choose the depth of history to retain the ring buffer (recommended), or set it to zero to trace the entire app (e.g., for UX analysis)

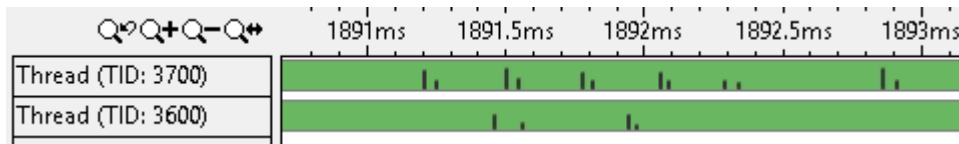
What is It Used for?

Detailed Tracing

Similar to Advanced Hotspots, **clocks and instructions are computed off Intel PT** and all data are “precise” for a traced portion of an app

Function / Call Stack	Hardware Event Count by Hardware Event Type			
	CPU_CLK_UNHALTED....	INST_RETIRED.ANY *	ITERATION_COUNT	LOOP_ENTRY_COUNT
[Loop@0x4242a0 in quantize_lines_xrpow]	3,212,192,253	5,293,751,016	168,345,431	1,570,800
└─ quantize_lines_xrpow	3,212,192,253	5,293,751,016	168,345,431	1,570,800
				1,570,510
				1,570,476
				858,734
				545,076
				145,076
				21,527
Hardware Event Count by Hardware Event Type				
	PT_CLOCKTICKS ▼ *	PT_INSTRUCTIONS	ITERATION_COUNT	LOOP_ENTRY_COUNT
[Loop@0x4242a0 in quantize_lines_xrpow]	28,333,656	18,065,355	325,484	2,966
└─ quantize_lines_xrpow	28,333,656	18,065,355	325,484	2,966
└─ func@0x4244a0	28,235,867	17,992,645	324,183	2,965
└─ count_bits	28,079,265	17,875,825	322,078	2,946
└─ [Loop@0x422af4 in func@0x422a40]	15,105,650	9,854,515	177,532	1,647
└─ [Loop@0x422be0 in func@0x422a40]	9,387,305	6,348,045	114,393	1,029
└─ [Loop@0x421840 in bin_search_StepSize]	3,081,039	1,439,790	25,946	232
└─ bin_search_StepSize	497,878	227,755	4,104	37

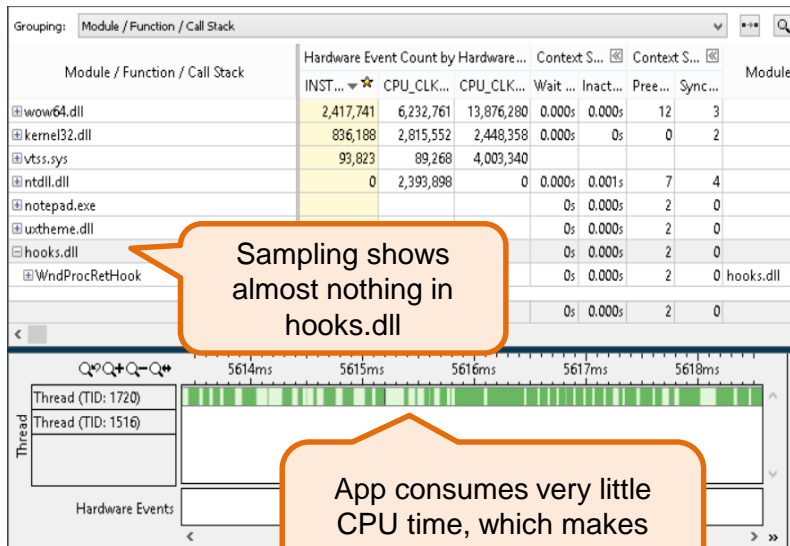
PT_clocks and PT_instructions shown on timeline as in sampling



What is It Used for?

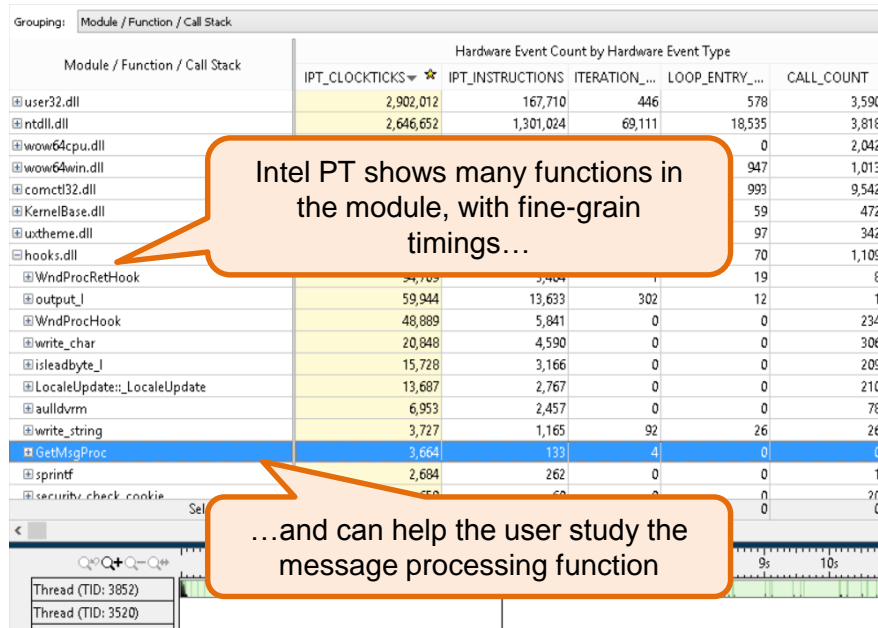
UX and low-latency analysis:

- Sampling revealed no issues, while Intel® PT showed the structure of message processing and enabled user to find a responsiveness issue:



Sampling shows almost nothing in hooks.dll

App consumes very little CPU time, which makes sampling less effective



Intel PT shows many functions in the module, with fine-grain timings...

...and can help the user study the message processing function

What is It Used for?

TSX analysis extension

TSX Hotspots - Haswell Copy

Analyze hotspots inside transactions. This analysis type is based on the hardware event-based sampling collection. Press F1 for more details.

Collect stacks
 Analyze user tasks
 Use Intel Processor Trace technology

Details

Events configured for CPU: 5th generation Intel(R) Core(TM) Processor family

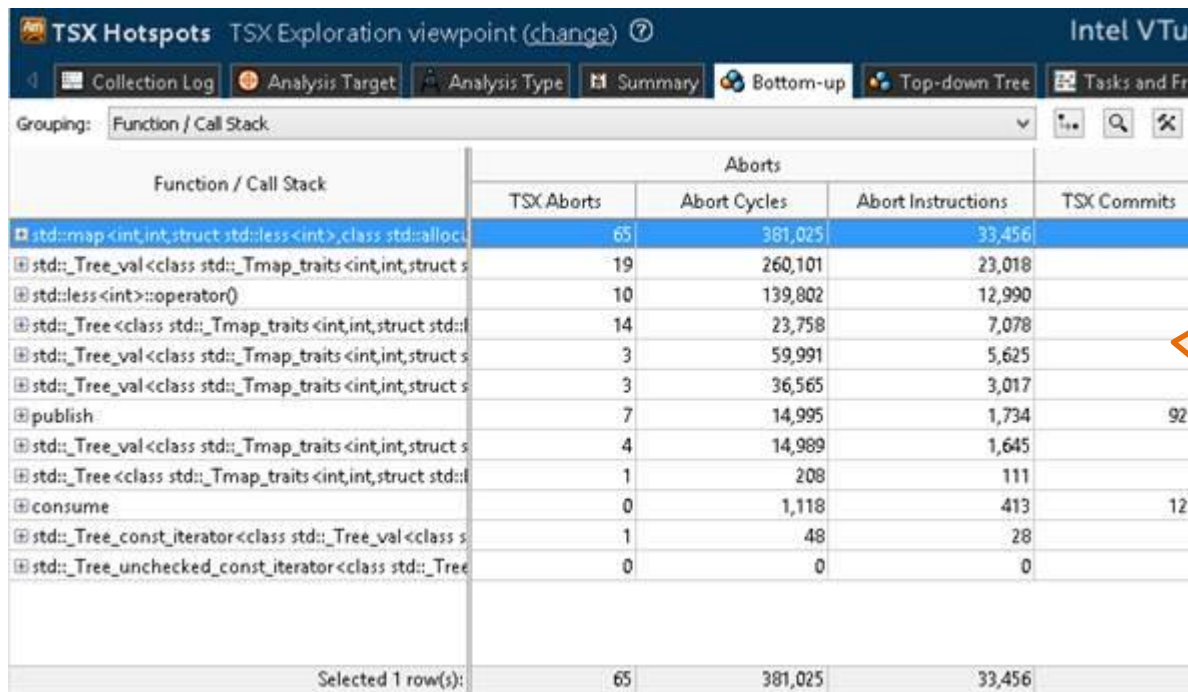
NOTE: For analysis purposes, Intel VTune Amplifier 2015 for Systems may adjust the Sample After values in the table below by a multiplier. The multiplier depends on the value of the Duration time estimate option specified in the Project Properties dialog.

Event Name	Sample After	Event Description
BR_INST_RETIRED.NEAR_TAKEN	5003	Taken branch instructions retired.

Select TSX Hotspots and check Intel® Processor Trace

What is It Used for?

TSX analysis extension



The screenshot shows the Intel VTune TSX Hotspots tool interface. The title bar reads "TSX Hotspots TSX Exploration viewpoint (change) Intel VTune". Below the title bar are several tabs: "Collection Log", "Analysis Target", "Analysis Type", "Summary", "Bottom-up", "Top-down Tree", and "Tasks and Fr". The "Grouping" dropdown is set to "Function / Call Stack". The main table displays the following data:

Function / Call Stack	Aborts			TSX Commits
	TSX Aborts	Abort Cycles	Abort Instructions	
std::map<int,int,struct std::less<int>,class std::alloc...	65	381,025	33,456	
std::_Tree_val<class std::_Tmap_traits<int,int,struct s...	19	260,101	23,018	
std::less<int>::operator()	10	139,802	12,990	
std::_Tree<class std::_Tmap_traits<int,int,struct std::l...	14	23,758	7,078	
std::_Tree_val<class std::_Tmap_traits<int,int,struct s...	3	59,991	5,625	
std::_Tree_val<class std::_Tmap_traits<int,int,struct s...	3	36,565	3,017	
publish	7	14,995	1,734	92
std::_Tree_val<class std::_Tmap_traits<int,int,struct s...	4	14,989	1,645	
std::_Tree<class std::_Tmap_traits<int,int,struct std::l...	1	208	111	
consume	0	1,118	413	12
std::_Tree_const_iterator<class std::_Tree_val<class s...	1	48	28	
std::_Tree_unchecked_const_iterator<class std::_Tree...	0	0	0	
Selected 1 row(s):	65	381,025	33,456	

User can study control flow paths leading to aborted vs. committed transactions, plus review the cost of aborts

Intel Processor Trace in VTune: Future Agenda

- *HW-related:*
 - Granular timing, and more tracing info as Intel PT evolves
- *Collection-related:*
 - Ring-buffer triggered by ITT API/external events, multiple ring-buffers per run
 - System-wide ring-buffer
- *Analysis-related:*
 - Loop dependency and cache efficiency analysis introduced
 - BW analysis improved (theoretical BW and scaling predictions)
 - Visualization – flame-charts on timeline, grouping by execution sequence

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

Intel, the Intel logo, Intel Processor Trace, and Intel VTune™ Amplifier are trademarks of Intel Corporation in the U.S. and/or other countries.

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

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

© 2015 Intel Corporation. All Rights Reserved.

Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT.

INTEL DISCLAIMS ALL EXPRESS OR IMPLIED WARRANTIES INCLUDING WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, AND NON-INFRINGEMENT.

