Intel® Processor Trace in Intel® VTune™ Amplifier

Stanislav Bratanov; Roman Belenov; Ludmila Pakhomova

4/27/2015

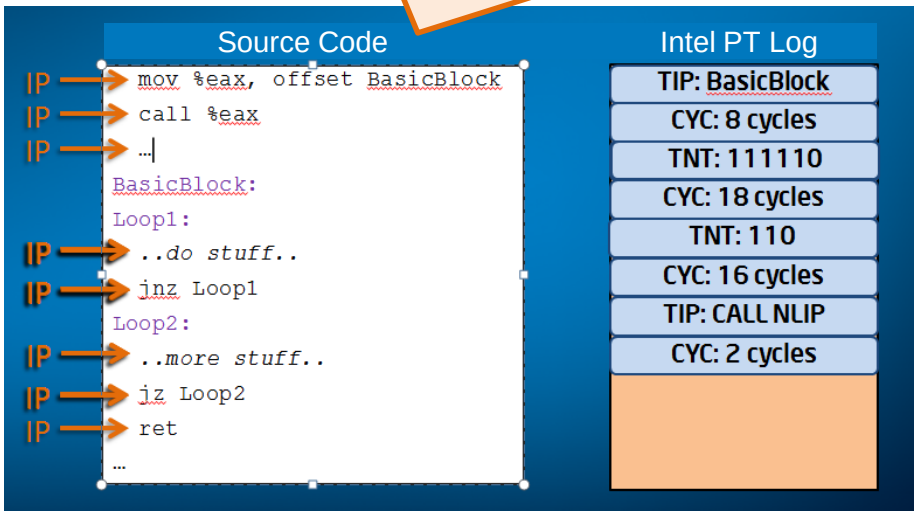
What is Intel Processor Trace?

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

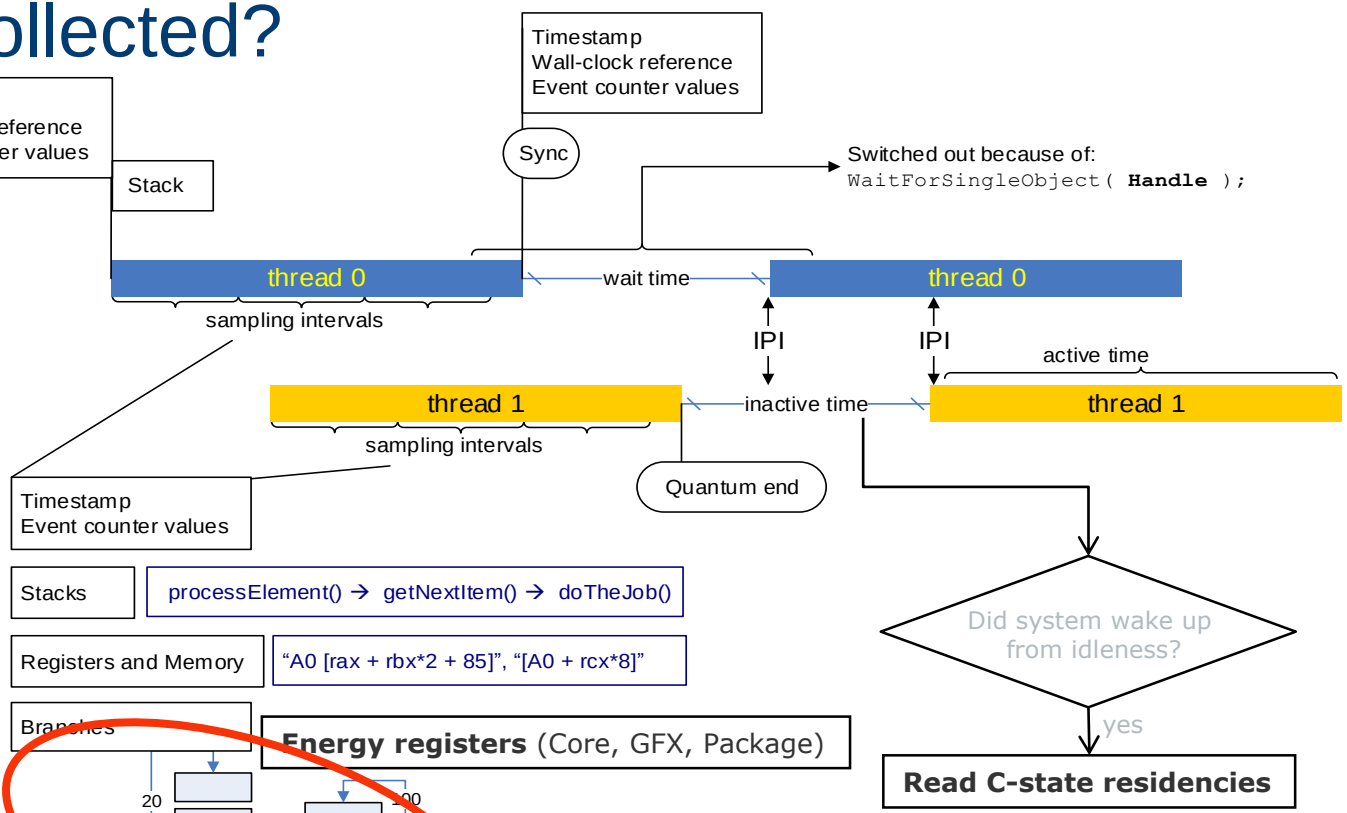
Intel PT differentiates between processes, but not SW threads. Per-thread collection may be necessary

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



How Is It Collected?

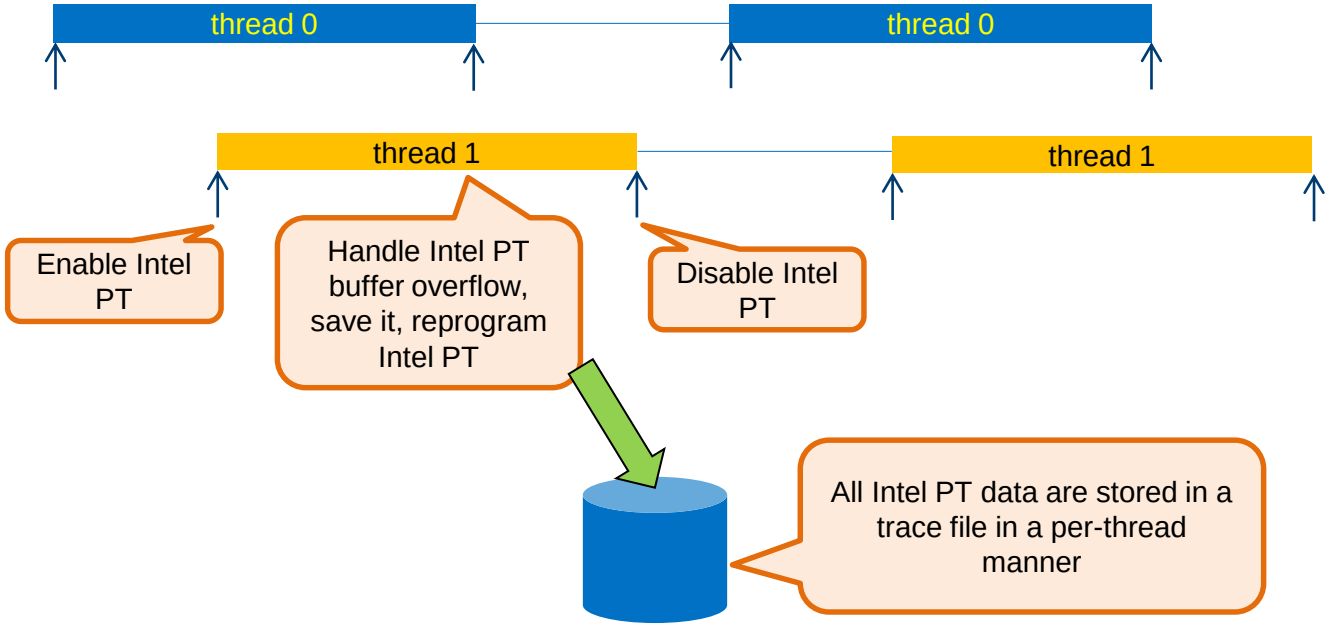
Statistical Mode



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

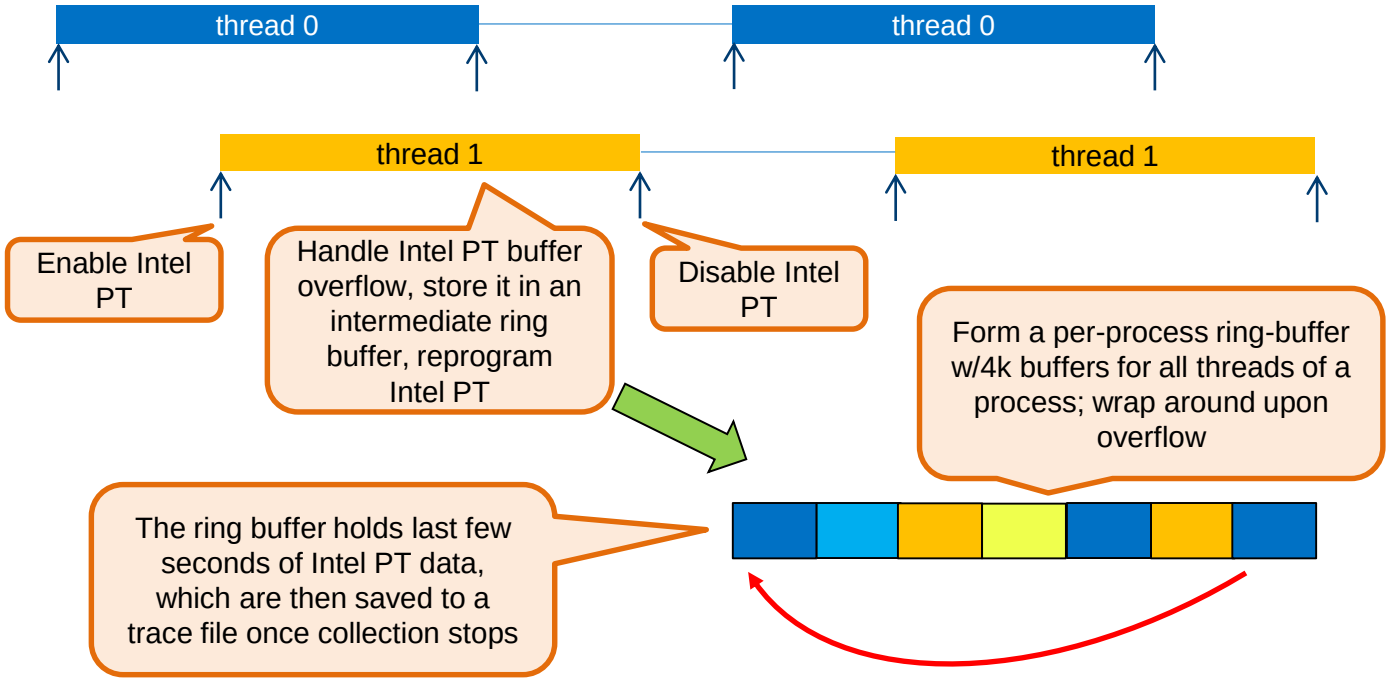
How is It Collected?

Full Trace Mode



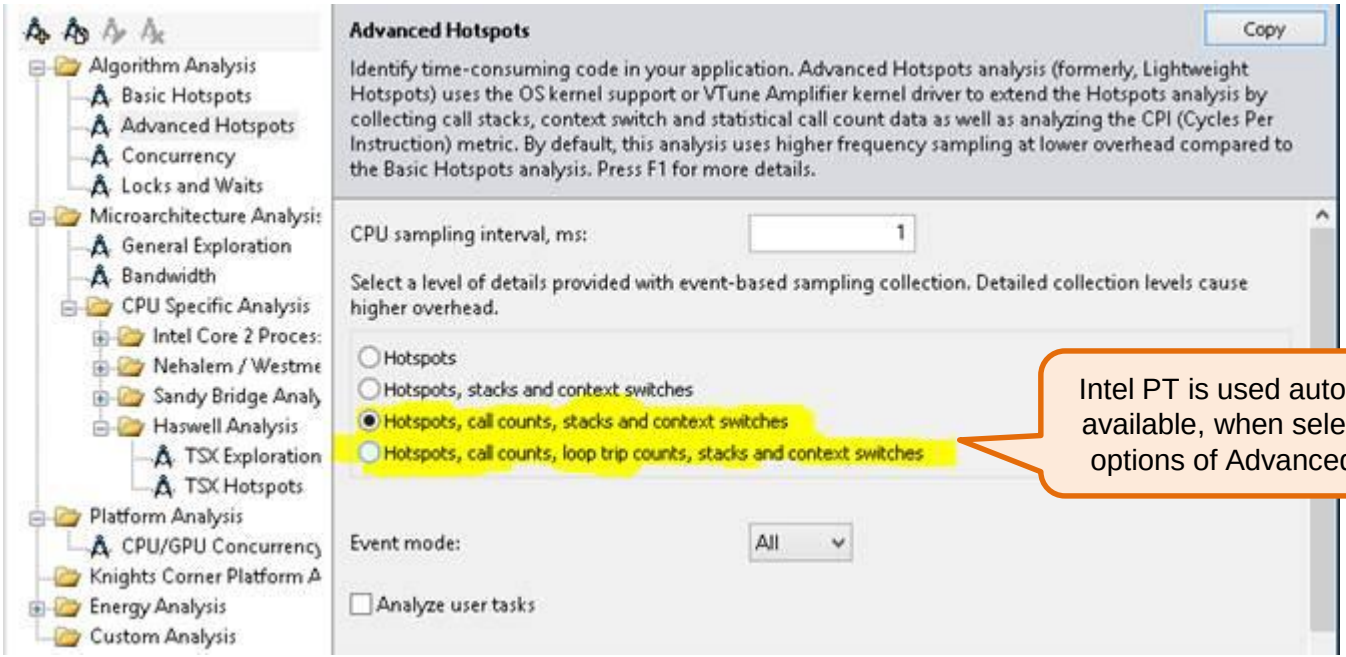
How is It Collected?

Ring Buffer Mode



What is It Used for?

Statistical Call Counts and Loop Iteration Counts



What is It Used for?

Call Counts and Loops

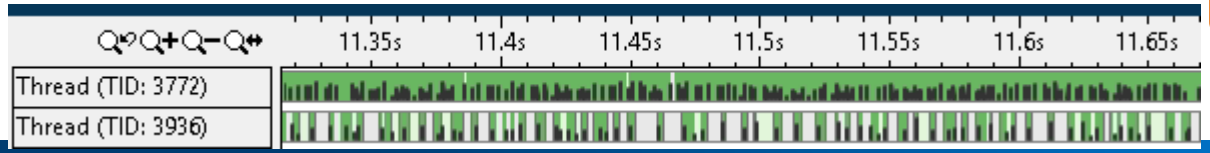
Function and loop stat metrics made more accurate with Intel PT

Function Stack	CPU_CLK_UNHALTED....	INST_RETIRED.ANY	CALL_COUNT	ITERATION_COUNT	LOOP_ENTRY_COUNT
	Hardware Event Count: Total by Har				
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

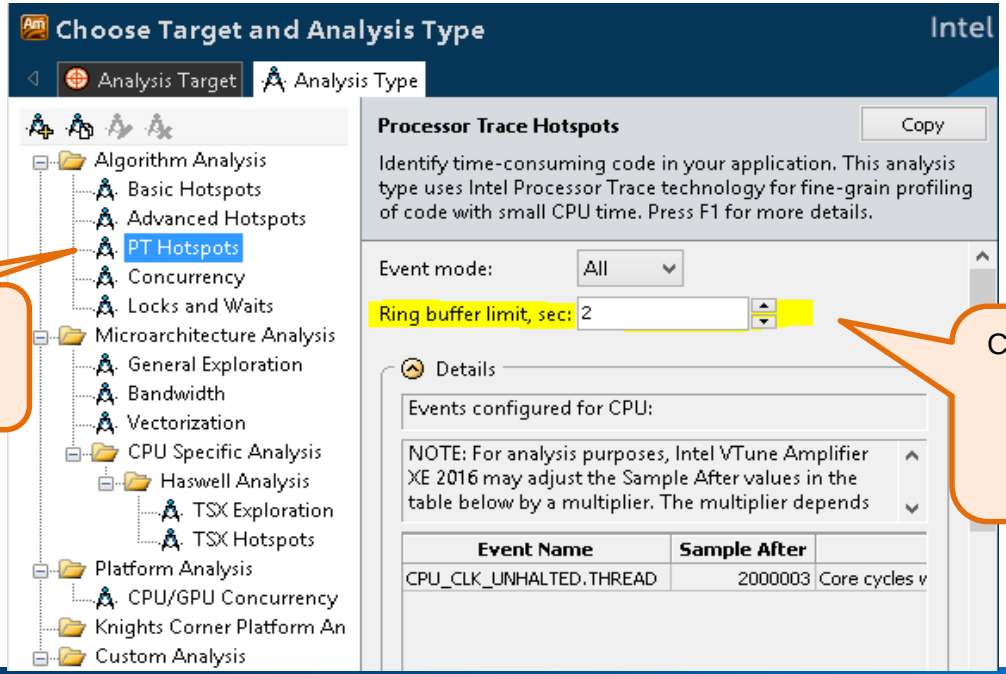
All metrics can be seen on timeline within thread quanta



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

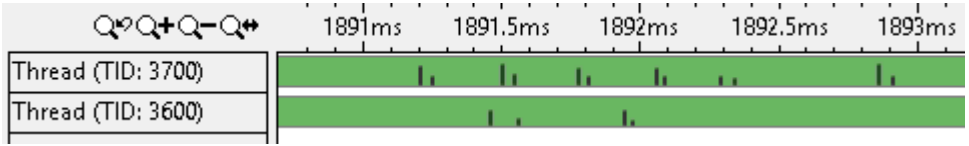
Similar to Advanced Hotspots, **clocks and instructions are computed off Intel PT** and all data are “precise” for a traced portion of an app

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

	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

1,570,510
1,570,476
858,734
545,076
145,076
21,527

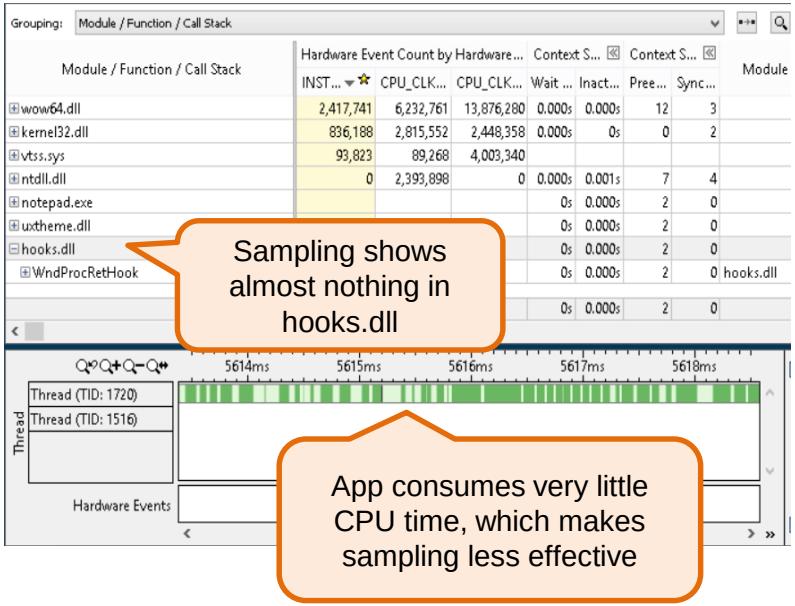
PT_clocks and PT_instructions shown on timeline as in sampling

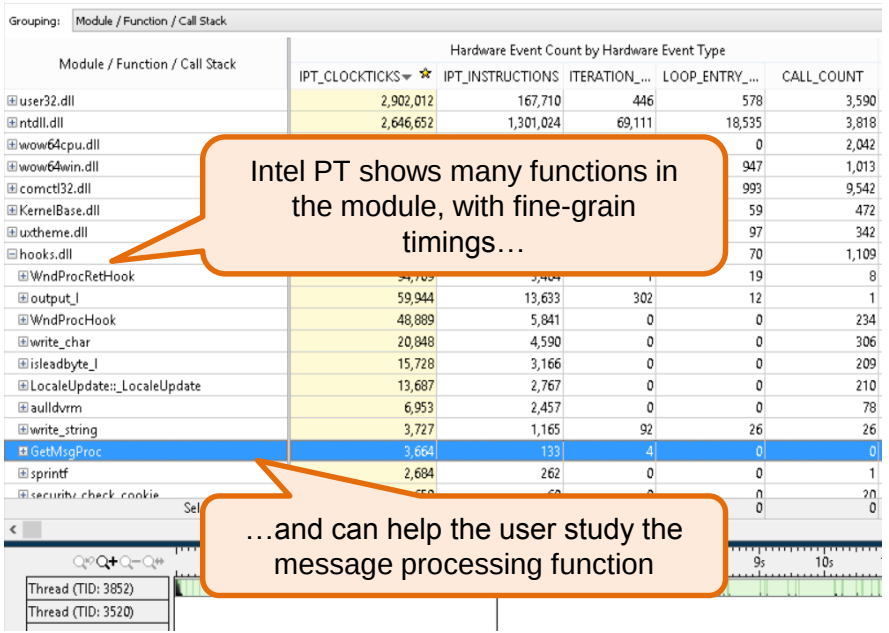


What is It Used for?

UX and low-latency analysis:

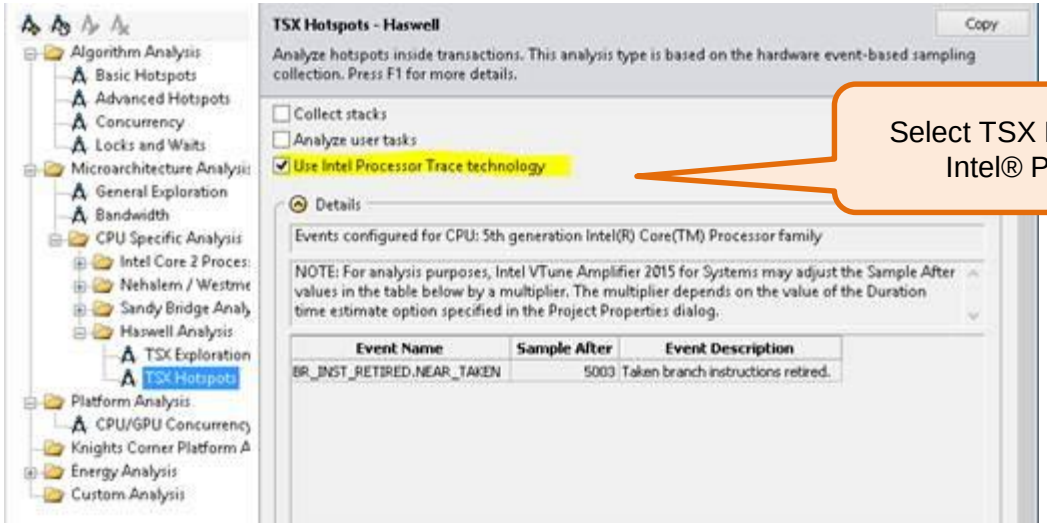
- Sampling revealed no issues, while Intel® PT showed the structure of message processing and enabled user to find a responsiveness issue:





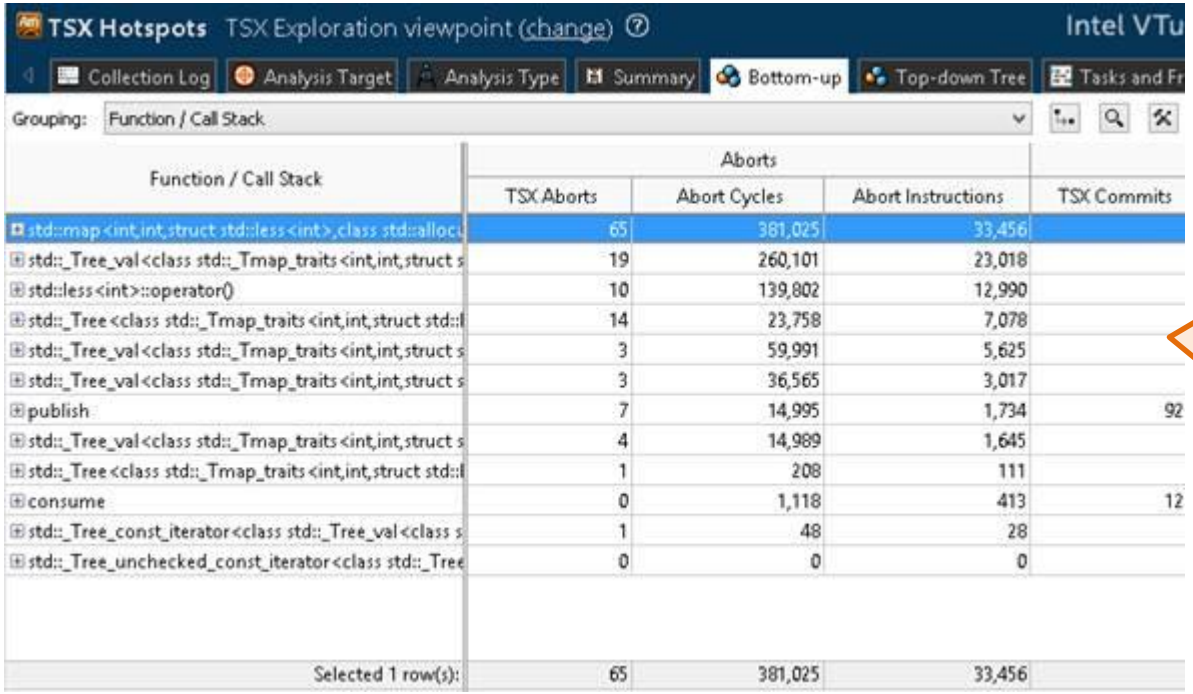
What is It Used for?

TSX analysis extension



What is It Used for?

TSX analysis extension



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

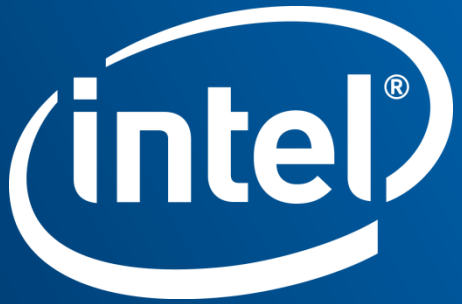
Intel Processor Trace in VTune: Future Agenda

- *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®