

Optimize Games Across Platforms

Pamela Harrison, Software Technical Consulting Engineer

Jennifer DiMatteo, Software Technical Consulting Engineer

03/2023

The Intel logo is displayed in white lowercase letters on a dark blue square background. A smaller, lighter blue square is positioned at the bottom left corner of the main blue square.

intel[®]

Agenda



Profiling Games

- Intel® Graphics Performance Analyzers (Intel® GPA)
- Intel® VTune™ Profiler



Intel® Graphics Performance Analyzers (Intel® GPA)

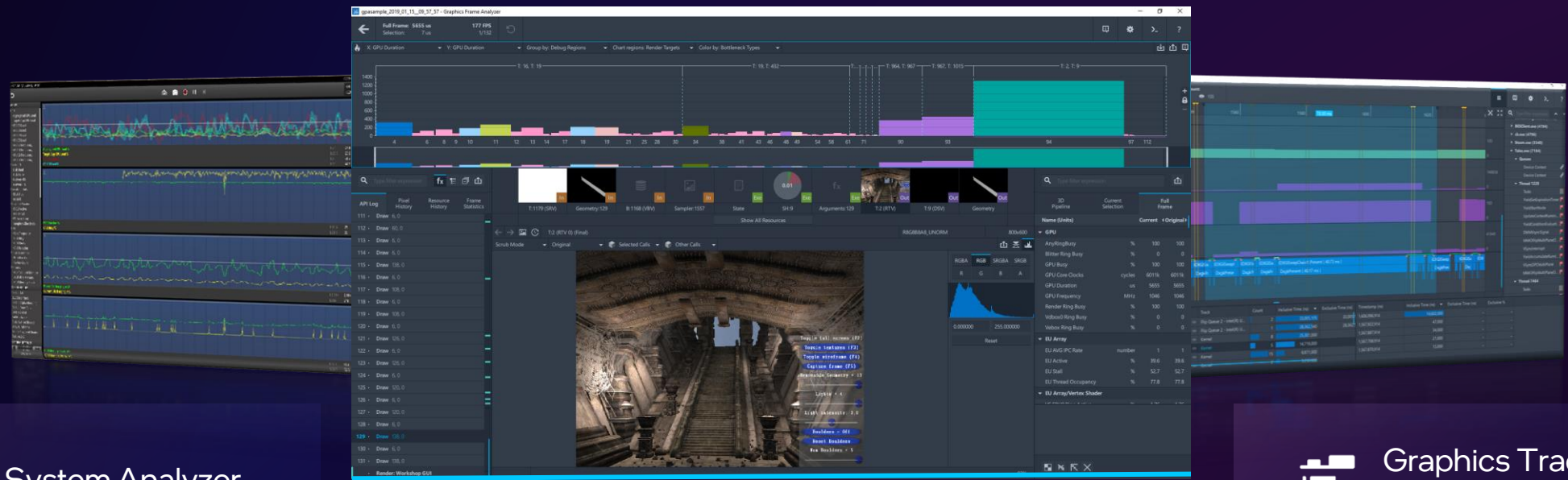
Pamela Harrison

Software Technical Consulting Engineer

What is Intel® GPA?

Tool suite for analyzing games and other real-time graphics applications

Locate graphics bottlenecks



System Analyzer



Graphics Frame Analyzer

Intel® GPA Framework



Graphics Trace Analyzer

Higher Performance Across Platforms

The Lost Legends of Redwall™ - Soma Games®, The Scout



intel® + SOMA
GAMES

PARTNERSHIP

Game Dev Stories
The Lost Legends
of Redwall™
Tuned with Intel® GPA

Game Dev Stories

	Intel UHD Graphics 10 th Generation Core (Comet Lake)	Intel UHD Graphics 12 th Generation Core (Alder Lake)	Intel UHD Graphics 13 th Generation Core (Raptor Lake)	Intel UHD Graphics 12 th Generation Core + Intel Arc™ A770
Original Frame (week 1)	6	9	11	45
No Combined Meshes (week 2)	10	19	22	133
Fastest Frame (week 3)	27	54	63	180

New in Intel® GPA's 2023.1 Release

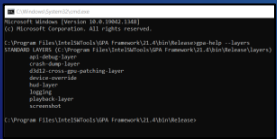
Tool suite for analyzing games and other real-time graphics applications



System Analyzer



P-Core and E-Core Loads



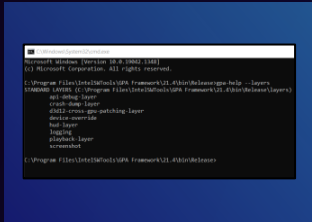
Intel® GPA Framework



Full-Stream Compression

New in Intel® GPA's 2023.1 Release

Tool suite for analyzing games and other real-time graphics applications



Intel® GPA Framework



Full-Stream Compression

```
$ gpa-injector.exe --layer capture:compression=lz4 <stream>
```

Intel® GPA Resources

- Intel® GPA: <https://developer.intel.com/gpa>
- Soma Games – Full Interview:
<https://www.intel.com/content/www/us/en/developer/videos/lost-legends-of-redwall-gpa.html>
- Graphics Frame Analyzer – In Depth:
<https://www.intel.com/content/www/us/en/developer/videos/an-in-depth-look-at-graphics-frame-analyzer.html>



Intel® VTune™ Profiler

Jennifer DiMatteo

Software Technical Consulting Engineer, VTune

Why Intel® VTune™ Profiler?

Created roughly 20 years ago, Intel VTune Profiler is Intel's flagship CPU performance profiler. Use VTune when you want to:

Optimize CPU compute-intensive tasks:

- Identify hotspots and reasons for slow CPU performance.

Tune CPU threading performance:

- Examine several common problems related to parallelism, such as thread imbalance and excessive context switching.

Profile games built with Unity* or Unreal Engine*

- View actual hardware performance of tasks annotated with VTune's API.

Optimize cache usage:

- Tune bandwidth-limited software and identify those memory objects which are bottlenecks.

Get best performance on the latest Intel hardware:

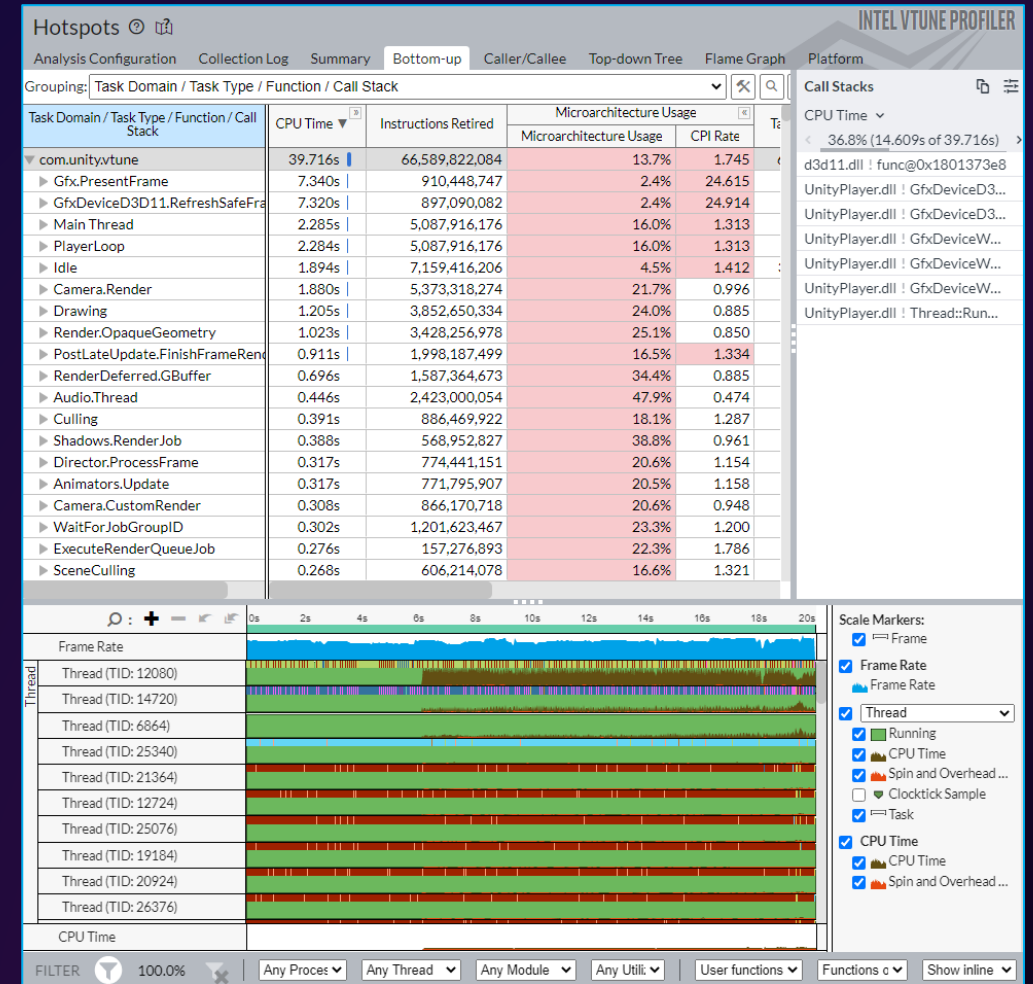
- Take advantage of cutting-edge architectures such as hybrid CPU



Intel® VTune™ Profiler Features

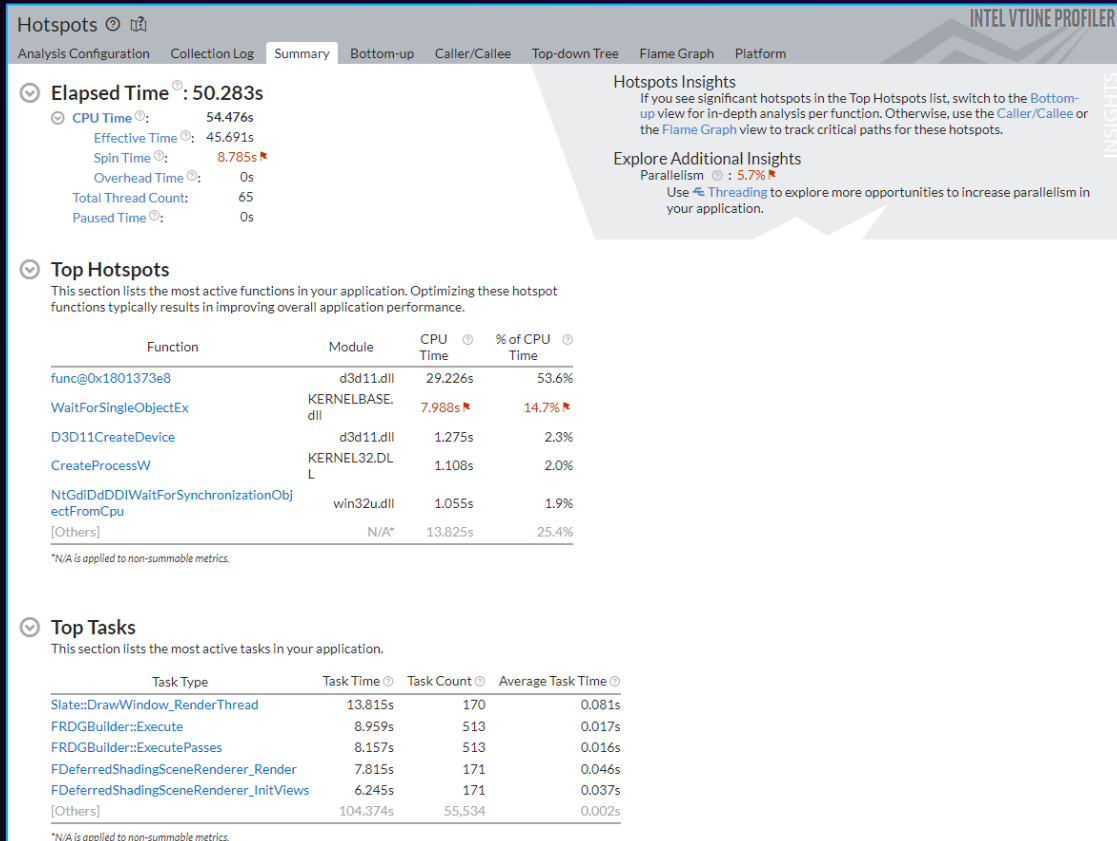
CPU Performance Profiling

- Locate bottlenecks in the CPU due to lock contention, synchronization issues, inefficient cache/memory utilization, and more
- Hardware event-based sampling supported for Intel hybrid CPU platforms
- Supports advanced profiling for Windows* and Linux* applications for unmanaged or managed code, or a mix
- Instrumentation and Tracing Technology (ITT) API built into Unreal Engine and Unity
- Free download with community support



Identify Hotspots

Quickly see which functions and tasks are consuming the most CPU time



Hotspots

Analysis Configuration Collection Log Summary Bottom-up Caller/Callee Top-down Tree Flame Graph Platform

Elapsed Time : 50.283s

- CPU Time : 54.476s
- Effective Time : 45.691s
- Spin Time : 8.785s
- Overhead Time : 0s
- Total Thread Count: 65
- Paused Time : 0s

Hotspots Insights

If you see significant hotspots in the Top Hotspots list, switch to the Bottom-up view for in-depth analysis per function. Otherwise, use the Caller/Callee or the Flame Graph view to track critical paths for these hotspots.

Explore Additional Insights

Parallelism : 5.7%

Use Threading to explore more opportunities to increase parallelism in your application.

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	% of CPU Time
func@0x1801373e8	d3d11.dll	29.226s	53.6%
WaitForSingleObjectEx	KERNELBASE.dll	7.988s	14.7%
D3D11CreateDevice	d3d11.dll	1.275s	2.3%
CreateProcessW	KERNEL32.DLL	1.108s	2.0%
NtGdiDdDDIWaitForSynchronizationObjectFromCpu	win32u.dll	1.055s	1.9%
[Others]	N/A*	13.825s	25.4%

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

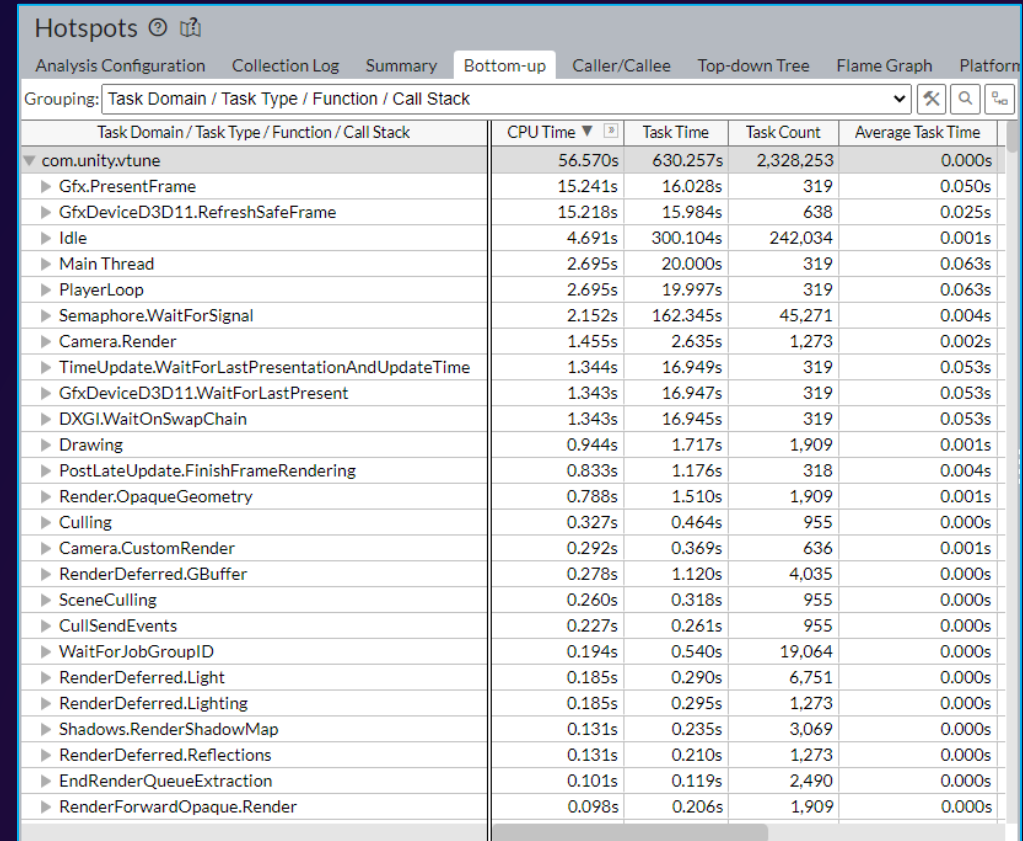
Top Tasks

This section lists the most active tasks in your application.

Task Type	Task Time	Task Count	Average Task Time
Slate::DrawWindow_RenderThread	13.815s	170	0.081s
FRDGBuilder::Execute	8.959s	513	0.017s
FRDGBuilder::ExecutePasses	8.157s	513	0.016s
FDeferredShadingSceneRenderer_Render	7.815s	171	0.046s
FDeferredShadingSceneRenderer_InitViews	6.245s	171	0.037s
[Others]	104.374s	55,534	0.002s

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

Hotspot Analysis Summary



Hotspots

Analysis Configuration Collection Log Summary Bottom-up Caller/Callee Top-down Tree Flame Graph Platform

Grouping: Task Domain / Task Type / Function / Call Stack

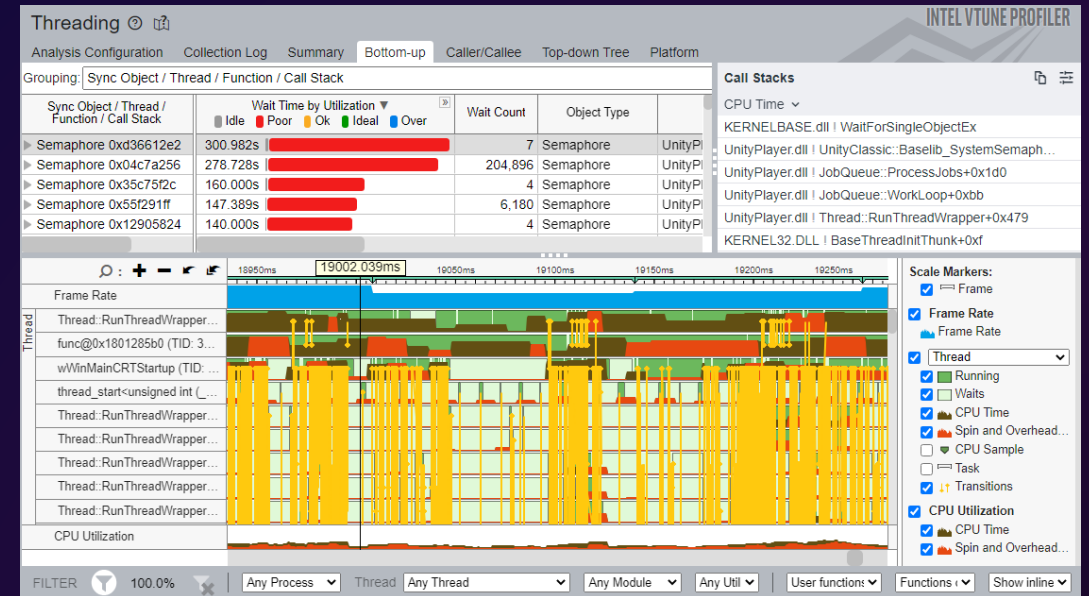
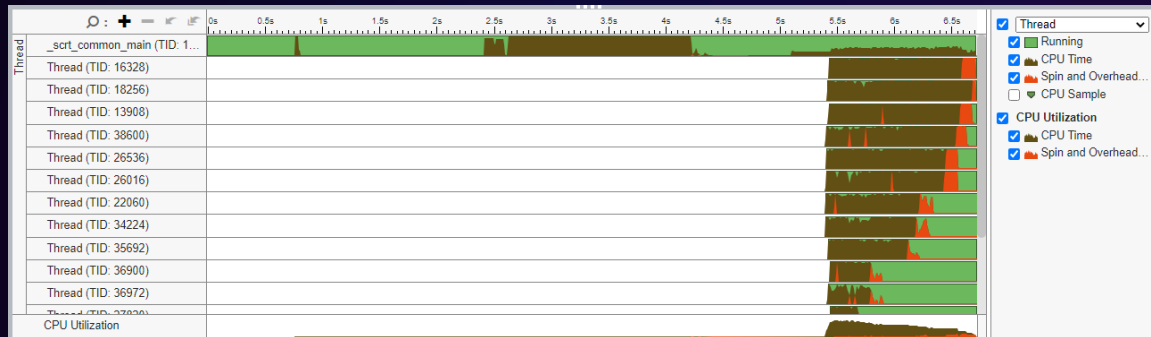
Task Domain / Task Type / Function / Call Stack	CPU Time	Task Time	Task Count	Average Task Time
com.unity.vtune	56.570s	630.257s	2,328,253	0.000s
▶ Gfx.PresentFrame	15.241s	16.028s	319	0.050s
▶ GfxDeviceD3D11.RefreshSafeFrame	15.218s	15.984s	638	0.025s
▶ Idle	4.691s	300.104s	242,034	0.001s
▶ Main Thread	2.695s	20.000s	319	0.063s
▶ PlayerLoop	2.695s	19.997s	319	0.063s
▶ Semaphore.WaitForSignal	2.152s	162.345s	45,271	0.004s
▶ Camera.Render	1.455s	2.635s	1,273	0.002s
▶ TimeUpdate.WaitForLastPresentationAndUpdateTime	1.344s	16.949s	319	0.053s
▶ GfxDeviceD3D11.WaitForLastPresent	1.343s	16.947s	319	0.053s
▶ DXGI.WaitOnSwapChain	1.343s	16.945s	319	0.053s
▶ Drawing	0.944s	1.717s	1,909	0.001s
▶ PostLateUpdate.FinishFrameRendering	0.833s	1.176s	318	0.004s
▶ Render.OpacityGeometry	0.788s	1.510s	1,909	0.001s
▶ Culling	0.327s	0.464s	955	0.000s
▶ Camera.CustomRender	0.292s	0.369s	636	0.001s
▶ RenderDeferred.GBuffer	0.278s	1.120s	4,035	0.000s
▶ SceneCulling	0.260s	0.318s	955	0.000s
▶ CullSendEvents	0.227s	0.261s	955	0.000s
▶ WaitForJobGroupID	0.194s	0.540s	19,064	0.000s
▶ RenderDeferred.Light	0.185s	0.290s	6,751	0.000s
▶ RenderDeferred.Lighting	0.185s	0.295s	1,273	0.000s
▶ Shadows.RenderShadowMap	0.131s	0.235s	3,069	0.000s
▶ RenderDeferred.Reflections	0.131s	0.210s	1,273	0.000s
▶ EndRenderQueueExtraction	0.101s	0.119s	2,490	0.000s
▶ RenderForwardOpaque.Render	0.098s	0.206s	1,909	0.000s

Detailed View

Visualize Thread Behavior

Poor Parallelism

Threads are running in parallel, but most threads finish and then spin while waiting for the rest to complete.



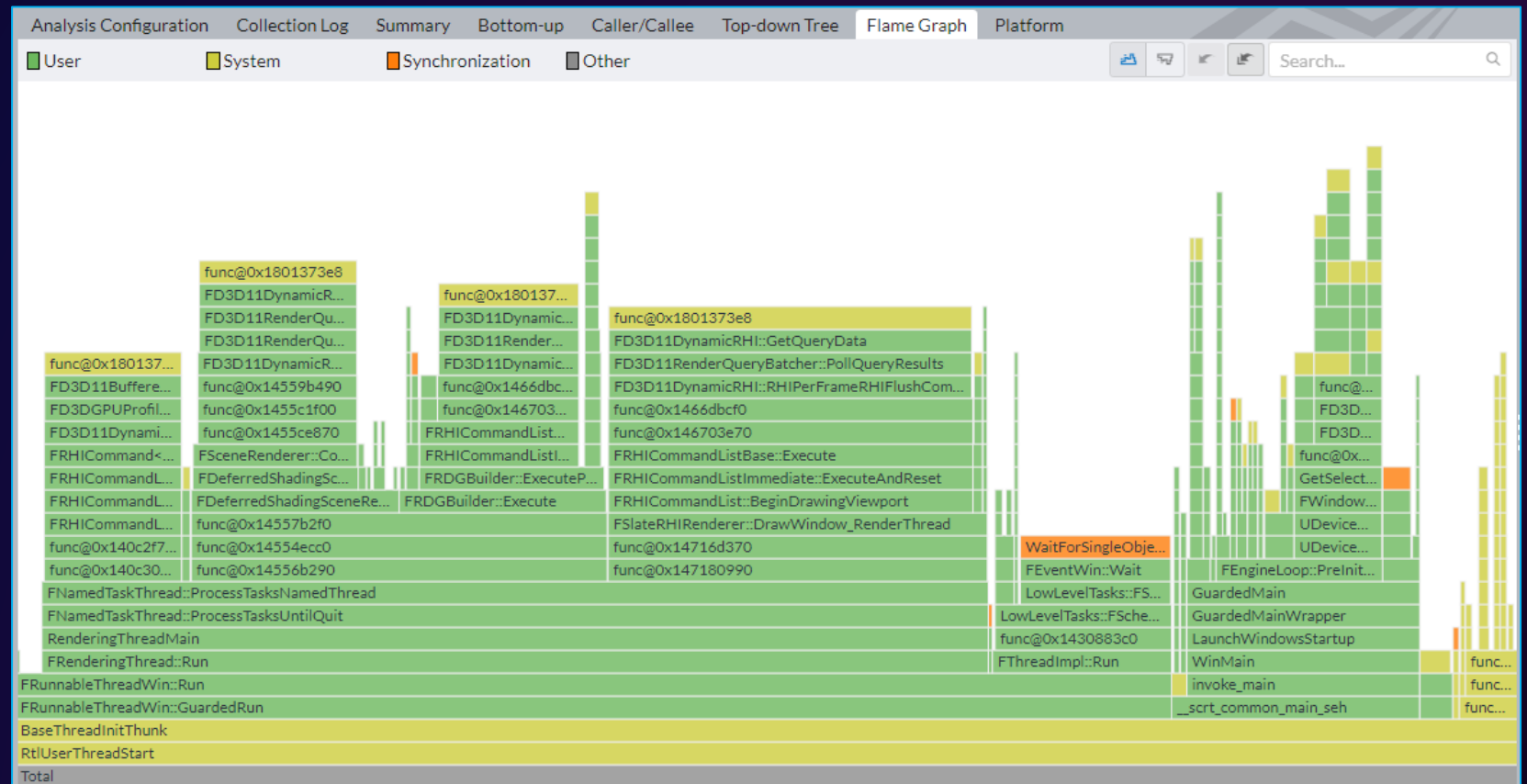
Threading Overhead

A large number of transitions happening between many threads that spend significant time waiting.

Stack Trace Visualization

Flame Graph

- Provides a clear view of the code path
- Column width represents CPU time
- Differentiate between user, system, and synchronization tasks





Instrumentation API


Instrumentation and Tracing Technology (ITT) APIs

- Control collection with pause/resume
- Mark up frames and tasks for filtering and grouping
- Integrated with Unity* and Unreal Engine*
 - Build the game for development and include debug files
 - Run with the VTune enabling parameter
 - VTune Cookbook recipes:
 - Profiling games built with Unity
 - Profiling games built with Unreal Engine


Launch Application ▾

Specify and configure your analysis target: an application or a script to execute. Follow [Prepare Application for Analysis](#) to compile your app for best analysis productivity.

Application:
C:\Users\dimatteo\fps_sample\Build\fps_sample.exe  

Application parameters:
-profiler-enable-vtune-markers 

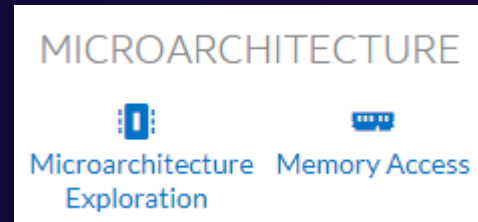
Use application directory as working directory

Advanced 

- > Intel® VTune™ Profiler Performance Analysis Cookbook
 - > Methodologies
 - ▾ Configuration Recipes
 - Analyzing Hot Code Paths Using Flame Graphs (NEW)
 - Improving Hotspot Observability in a C++ Application Using Flame Graphs
 - Profiling Games built with Unity* (NEW)
 - Profiling Games built with Unreal Engine* (NEW)

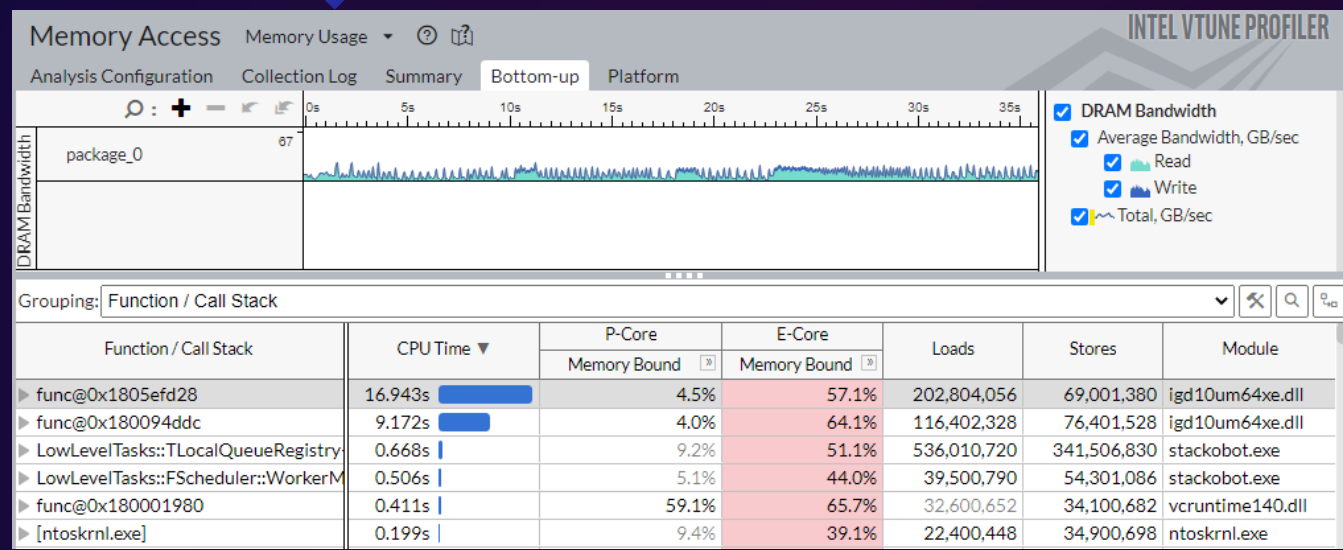
Microarchitecture Analysis

How is your game using hardware resources?



Microarchitecture Analysis

How is your game using CPU resources?



Microarchitecture Analysis

How is your game using CPU resources?

Microarchitecture Exploration Microarchitecture Exploration

Analysis Configuration Collection Log Summary Bottom-up Event Count

Elapsed Time: 23.593s

- Clockticks: 53,007,552,000
- Instructions Retired: 28,194,816,000
- CPI Rate: 1.880
 - P-Core: 1.905
 - E-Core: 1.463
- MUX Reliability: 0.952
- P-Core:
 - Retiring: 9.7% of Pipeline Slots
 - Front-End Bound: 10.0% of Pipeline Slots
 - Bad Speculation: 3.4% of Pipeline Slots
 - Back-End Bound: 77.0% of Pipeline Slots
- E-Core:
 - Retiring: 87.2% of Pipeline Slots
 - Front-End Bound: 24.1% of Pipeline Slots
 - Bad Speculation: 0.0% of Pipeline Slots
 - Back-End Bound: 29.7% of Pipeline Slots
 - Core Bound: 0.0% of Clockticks
 - Memory Bound: 29.7% of Clockticks
 - Back-End Bound Auxiliary: 29.7% of Pipeline Slots
- Average CPU Frequency: 2.2 GHz
- Total Thread Count: 64
- Paused Time: 0s

MICROARCHITECTURE

Microarchitecture Memory Access Exploration

Memory Access Memory Usage

INTEL VTUNE PROFILER

Analysis Configuration Collection Log Summary Bottom-up Platform

DRAM Bandwidth

package_0 87

0s 5s 10s 15s 20s 25s 30s 35s

DRAM Bandwidth

Average Bandwidth, GB/sec
 Read
 Write
 Total, GB/sec

Grouping: Function / Call Stack

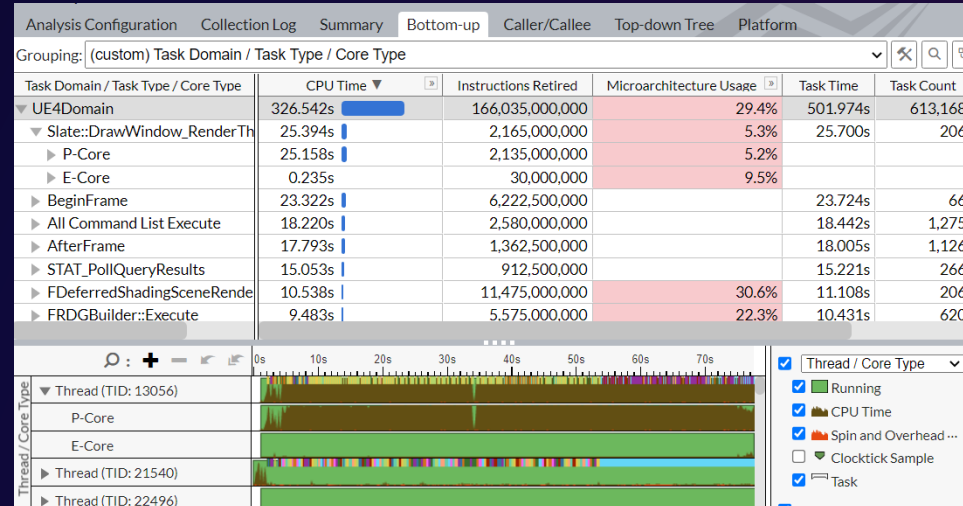
Function / Call Stack	CPU Time	P-Core		E-Core		Loads	Stores	Module
		Memory Bound	Memory Bound	Memory Bound	Memory Bound			
func@0x1805efd28	16.943s	4.5%	57.1%	202,804,056	69,001,380	igd10um64xe.dll		
func@0x180094ddc	9.172s	4.0%	64.1%	116,402,328	76,401,528	igd10um64xe.dll		
LowLevelTasks::TLocalQueueRegistry	0.668s	9.2%	51.1%	536,010,720	341,506,830	stackobot.exe		
LowLevelTasks::FScheduler::WorkerM	0.506s	5.1%	44.0%	39,500,790	54,301,086	stackobot.exe		
func@0x180001980	0.411s	59.1%	65.7%	32,600,652	34,100,682	vcruntime140.dll		
[ntoskrnl.exe]	0.199s	9.4%	39.1%	22,400,448	34,900,698	ntoskrnl.exe		



Hybrid CPU Analysis

Support for hybrid CPU configurations (processor code name Alder Lake and Raptor Lake)

- See how tasks, functions, and threads are utilizing performance and/or efficient cores (P-Core and E-Core)

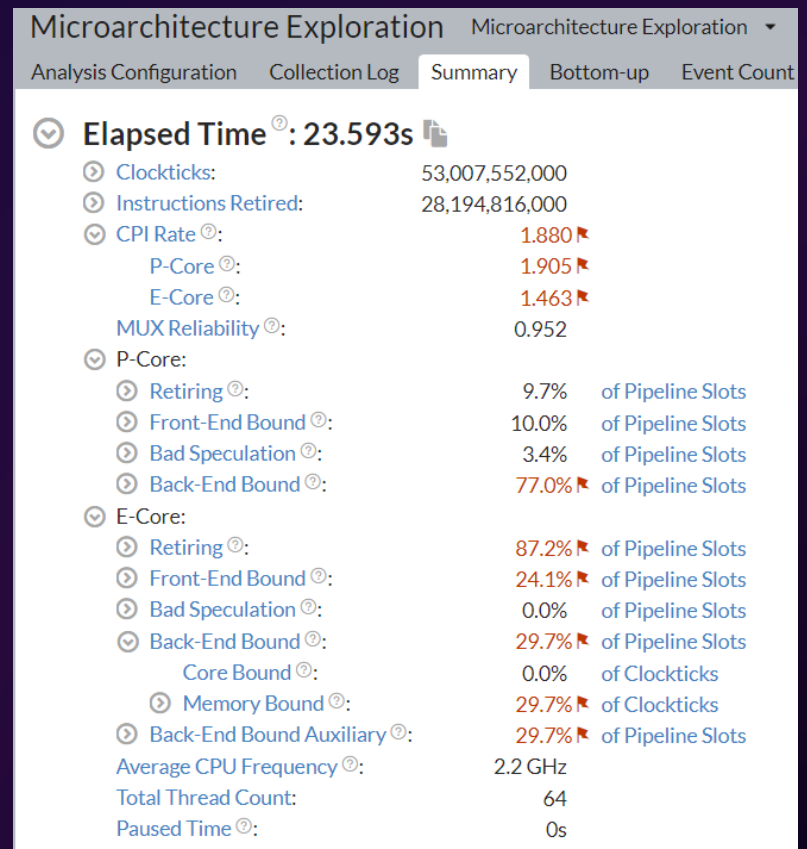
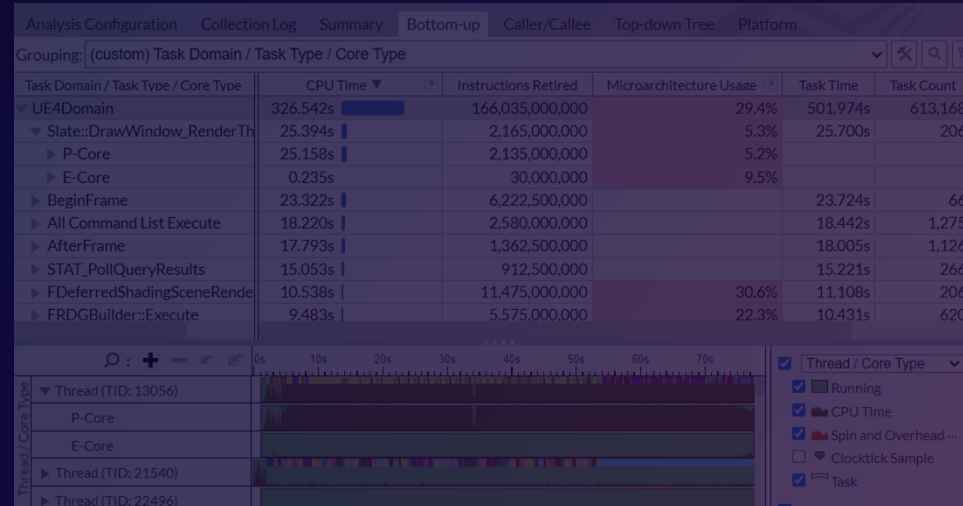


Hybrid CPU Analysis

Support for hybrid CPU configurations (processor code name Alder Lake and Raptor Lake)

- See how tasks, functions, and threads are utilizing performance and/or efficient cores (P-Core and E-Core)

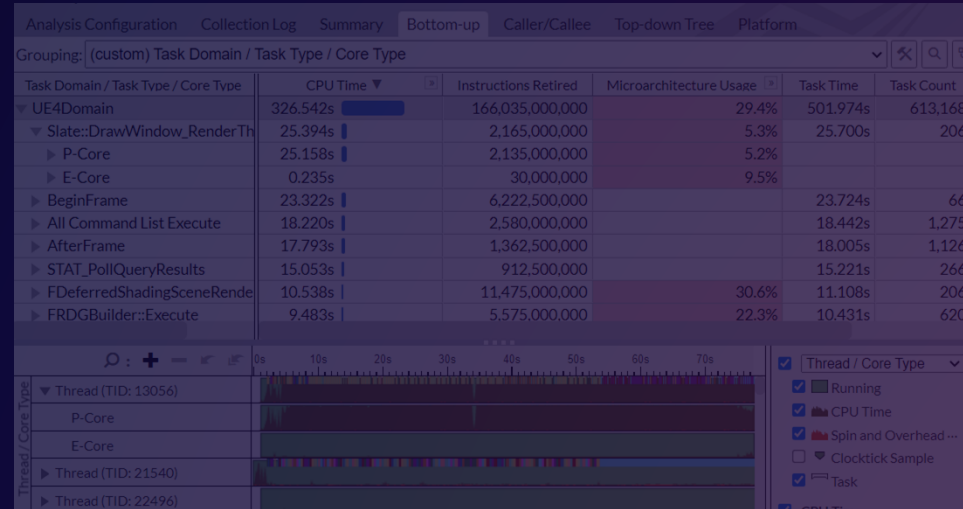
- Identify performance issues related to cache and memory utilization for each core type



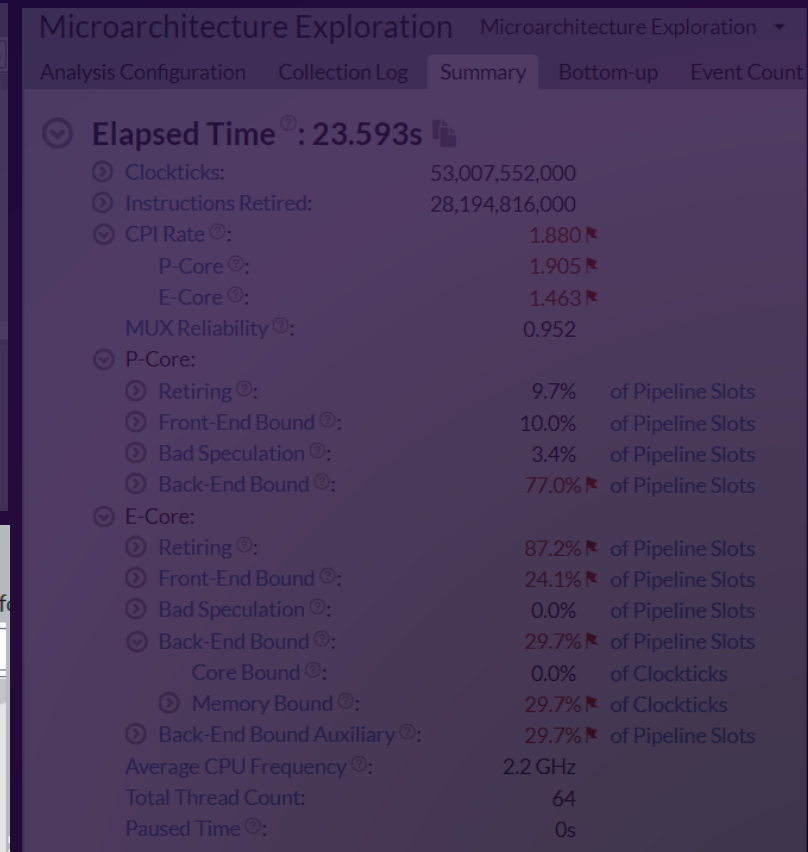
Hybrid CPU Analysis

Support for hybrid CPU configurations (processor code name Alder Lake and Raptor Lake)

- See how tasks, functions, and threads are utilizing performance and/or efficient cores (P-Core and E-Core)
- Identify performance issues related to cache and memory utilization for each core type
- View hardware events triggered by vector and floating-point instructions to understand SIMD performance



Function / Call Stack	DD 128	INT_VEC_RETIREDD.ADD 256	INT_VEC_RETIREDD.MUL 2
multiply3	0	6,003,018,009	6,003,018,009
func@0x14041dd80	0	0	0
init_arr	0	0	0



Intel® VTune™ Profiler Resources

- Intel® VTune™ Profiler: <https://developer.intel.com/vtune>
- Cookbook: <https://www.intel.com/content/www/us/en/docs/vtune-profiler/cookbook>
- Game Tuning with Intel Profilers: <https://www.intel.com/content/www/us/en/developer/articles/guide/game-tuning-with-intel.html>

Thank you

Notices & Disclaimers

Performance varies by use, configuration and other factors. Learn more at www.intel.com/PerformanceIndex (graphics and accelerators).

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. No product or component can be absolutely secure.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

Intel technologies may require enabled hardware, software or service activation.

All product plans and roadmaps are subject to change without notice.

Code names are used by Intel to identify products, technologies, or services that are in development and not publicly available. These are not "commercial" names and not intended to function as trademarks.

Statements that refer to future plans or expectations are forward-looking statements. These statements are based on current expectations and involve many risks and uncertainties that could cause actual results to differ materially from those expressed or implied in such statements. For more information on the factors that could cause actual results to differ materially, see our most recent earnings release and SEC filings at www.intc.com.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

intel®

