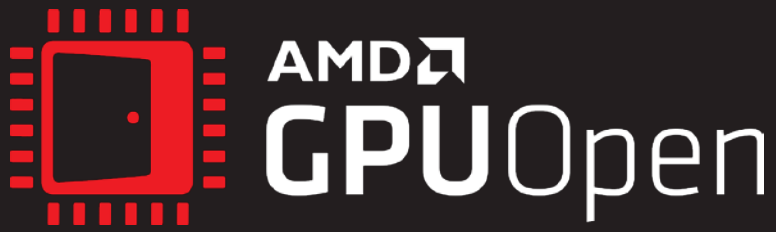


AMD RYZEN™ PROCESSOR SOFTWARE OPTIMIZATION

PRESENTED BY KENNETH MITCHELL

LET'S BUILD… 2020

MAY 15, 2020

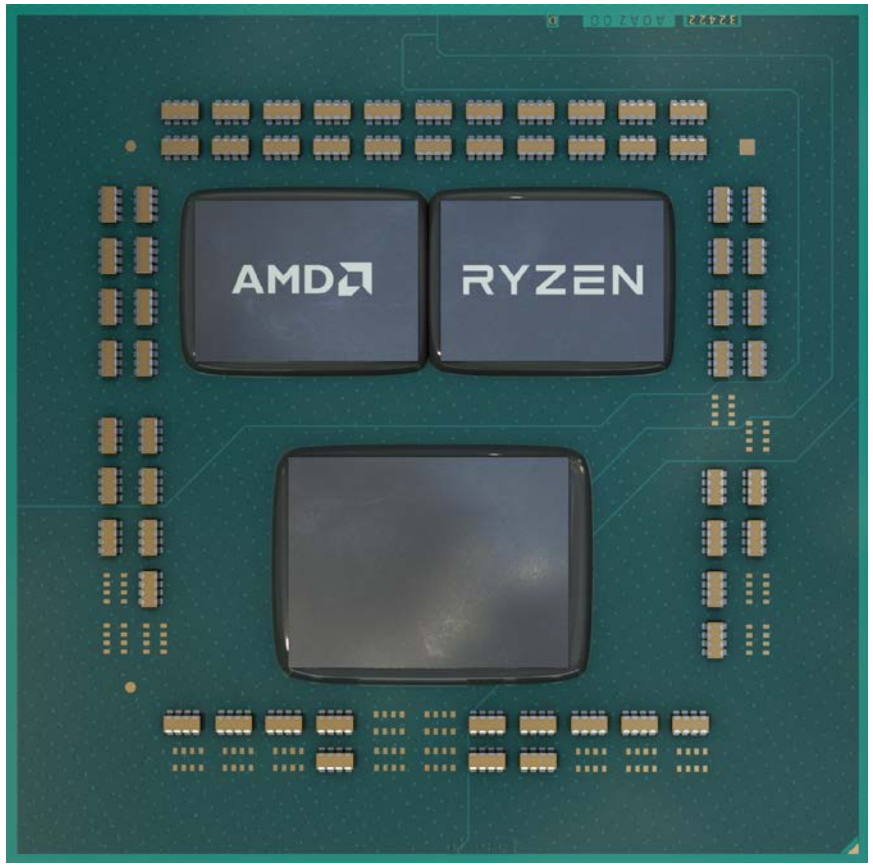
ABSTRACT

Join AMD Game Engineering team members for an introduction to the AMD Ryzen™ family of processors followed by advanced optimization topics. Learn about the high-performance AMD "Zen 2" microarchitecture and profiling tools. Gain insight into code optimization opportunities and lessons learned. Examples may include C/C++, assembly, and hardware performance-monitoring counters.

SPEAKER BIOGRAPHY

Ken Mitchell is a Principal Member of Technical Staff in the Radeon™ Technologies Group/AMD ISV Game Engineering team where he focuses on helping game developers utilize AMD processors efficiently. His previous work includes automating and analyzing PC applications for performance projections of future AMD products as well as developing benchmarks. Ken studied computer science at the University of Texas at Austin.

AGENDA



- Success Stories
- “Zen 2” Architecture Processors
- AMD uProf Profiler
- Optimizations and Lessons Learned
- Contacts

SUCCESS STORIES

SUCCESS STORIES

BORDERLANDS 3

DirectX® 12

GEARS 5

DirectX® 12

WORLD WAR Z

Vulkan®

“ZEN 2” ARCHITECTURE PROCESSORS

“ZEN 2” PRODUCT EXAMPLES

NOTEBOOK

“Renoir”

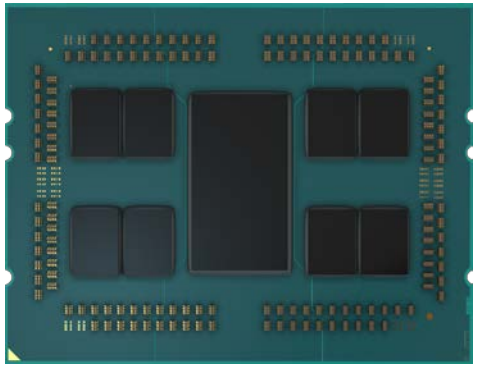
AMD Ryzen™ 7 4800U 8-Core Processor

DESKTOP

“Matisse”

AMD Ryzen™ 9 3950X 16-Core Processor

HIGH END DESKTOP

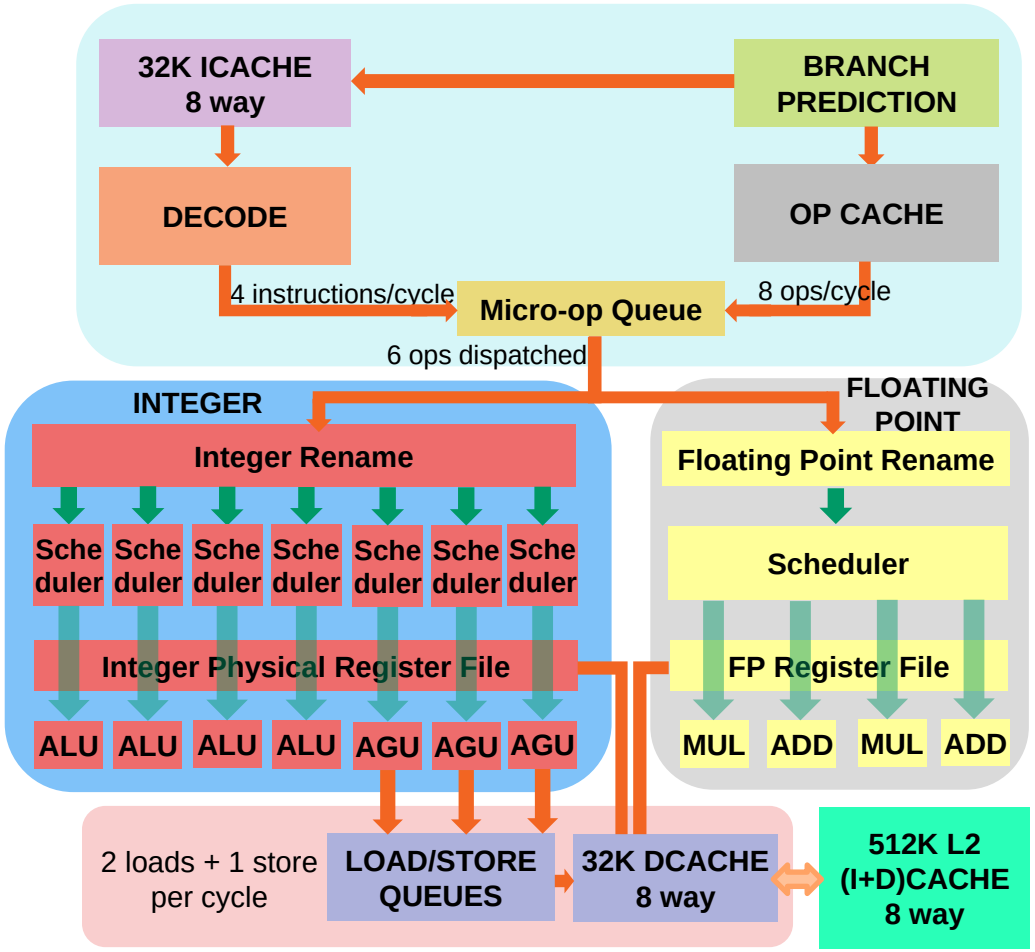


“Castle Peak”

AMD Ryzen™ Threadripper™ 3990X 64-Core Processor

MICROARCHITECTURE

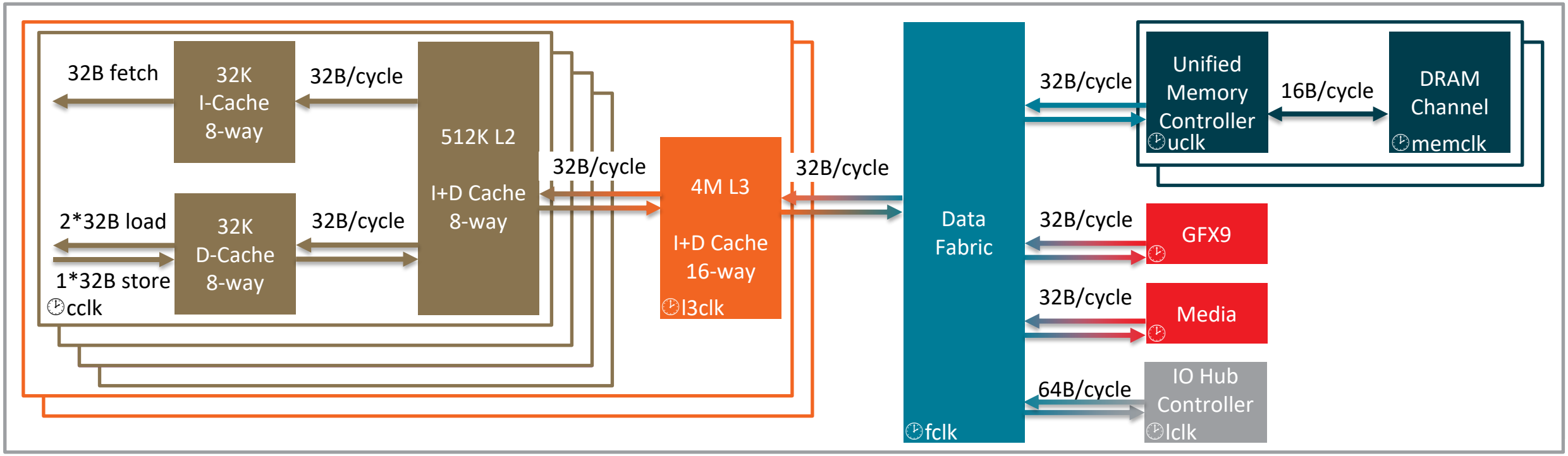
ADVANCES IN “ZEN 2” MICROARCHITECTURE



- +15% IPC Improvement from “Zen” to “Zen 2”
- 2x op cache capacity
- Reoptimized L1I cache
- 3rd address generation unit
- 2x FP data path width
- 2x L3 capacity
- Improved branch prediction accuracy
- Hardware optimized Security Mitigations
- Secure Virtualization with Guest Mode Execute Trap (GMET)
- Improved SMT fairness
  - for ALU and AGU schedulers
- Improved Write Combining Buffer

DATA FLOW

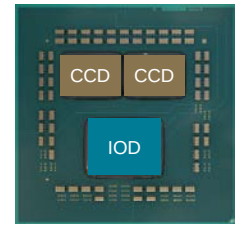
“RENOIR” 8 CORE PROCESSOR

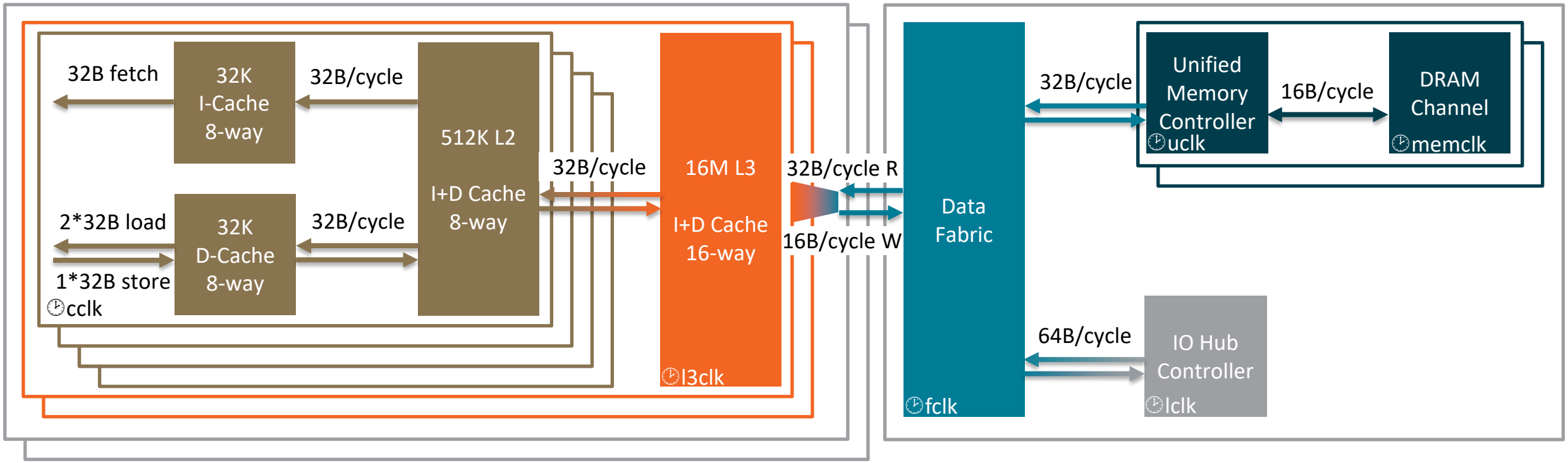


AMD Ryzen™ 7 4800U, 15W TDP, 8 Cores, 16 Threads, 4.2 GHz max boost clock, 1.8 GHz base clock, integrated GPU.

* Monolithic Die. Each 4M L3 Cache has its own 32B/cycle link to the data fabric. 64b DDR4 Channel Shown.

“MATISSE” 16 CORES PROCESSOR



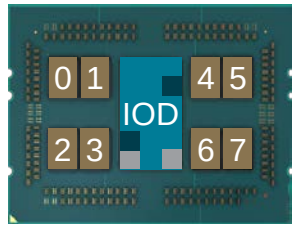


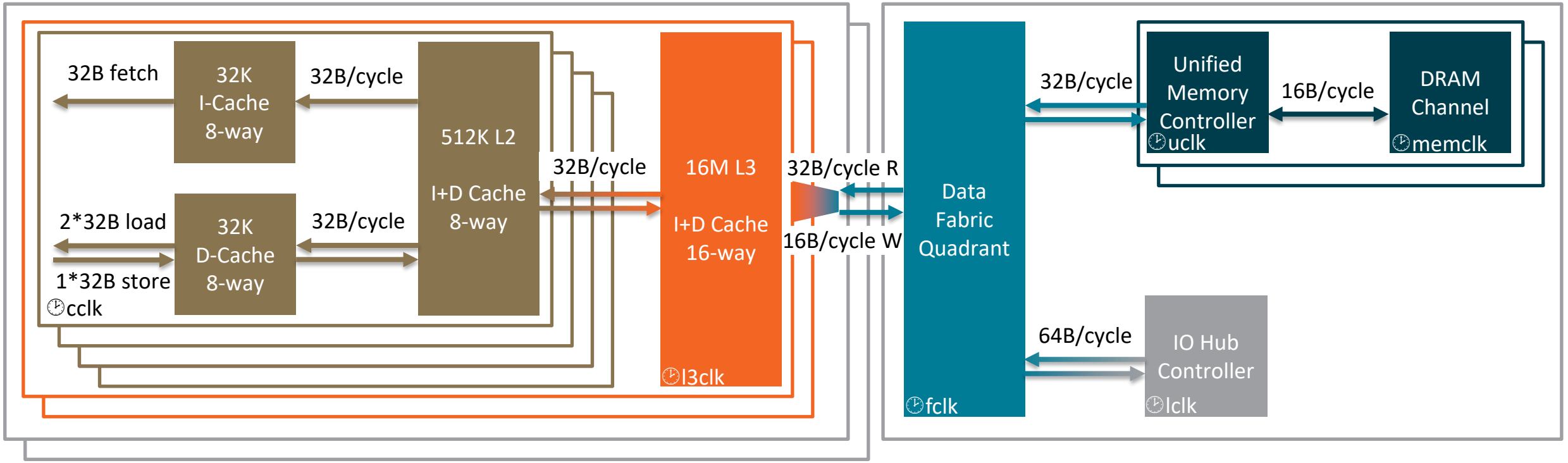
AMD Ryzen™ 9 3950X, 105W TDP, 16 Cores, 32 Threads, 4.7 GHz max boost clock, 3.5 GHz base clock.

* Two Core Complex Die (CCD). Each CCD has two 16M L3 Cache Complexes.

* The L3 Cache Complexes within a CCD share a single link to the Data Fabric.

“CASTLE PEAK” 64 CORE PROCESSOR





AMD Ryzen™ Threadripper™ 3990X, 280W TDP, 64 Cores, 128 Threads, 4.3 GHz max boost clock, 2.9 GHz base clock.

* Two CCDs per Data Fabric Quadrant.

* Two Data Fabric Quadrants have Unified Memory Controllers and two have IO Hubs.

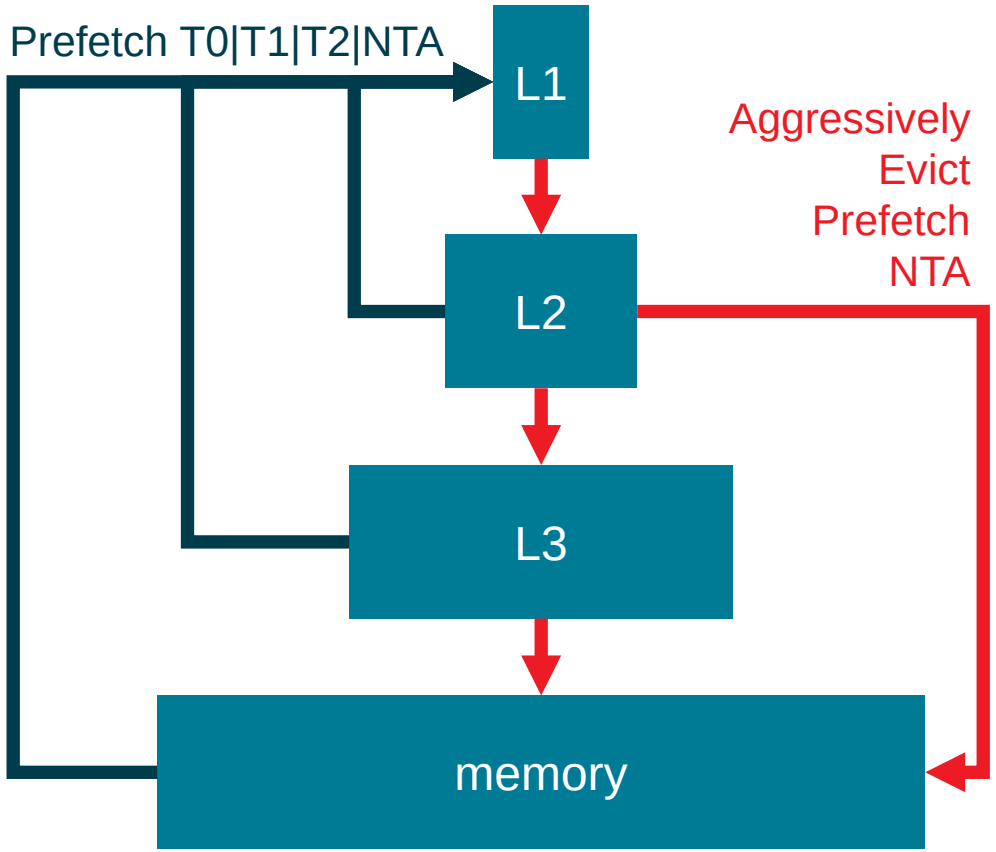
INSTRUCTION SET

INSTRUCTION SET EVOLUTION

YEAR	FAMILY	PRODUCT FAMILY	ARCHITECTURE	EXAMPLE PRODUCT	CLWB	ADX	CLFLUSHOPT	RDSEED	SHA	SMAP	XGETBV	XSAVEC	XSAVES	AVX2	BMI2	MOVBE	RDRND	SMEP	FSGSBASE	XSAVEOPT	BMI	FMA	F16C	AES	AVX	OSXSAVE	PCLMULQDQ	SSE4.1	SSE4.2	XSAVE	SSSE3	MONITORX	CLZERO	WBNOINVD	FMA4	TBM	XOP
2019	17h	“Matisse”	“Zen2”	Ryzen™ 9 3950X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
2017	17h	“Summit Ridge”, ”Pinnacle Ridge”	“Zen”, “Zen+”	Ryzen™ 7 2700X	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0
2015	15h	“Carrizo”, “Bristol Ridge”	“Excavator”	A12-9800	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1
2014	15h	“Kaveri”, “Godavari”	“Steamroller”	A10-7890K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1
2012	15h	“Vishera”	“Piledriver”	FX-8370	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	1
2011	15h	“Zambezi”	“Bulldozer”	FX-8150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	1	0	1
2013	16h	“Kabini”	“Jaguar”	A6-1450	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
2011	14h	“Ontario”	“Bobcat”	E-450	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
2011	12h	“Llano”	“Husky”	A8-3870	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

“Zen 2” added CLWB and the AMD vendor specific instruction WBNOINVD.

SOFTWARE PREFETCH LEVEL INSTRUCTIONS



- Loads a cache line from the specified memory address into the data-cache level specified by the locality reference T0, T1, T2, or NTA.
- If a memory fault is detected, a bus cycle is not initiated and the instruction is treated as an NOP.
- Prefetch levels T0/T1/T2 are treated identically in “Zen” & “Zen 2” microarchitectures.
- The non-temporal cache fill hint, indicated with PREFETCHNTA, reduces cache pollution for data that will only be used once. It is not suitable for cache blocking of small data sets. Lines filled into the L2 cache with PREFETCHNTA are marked for quicker eviction from the L2 and when evicted from the L2 are not inserted into the L3.
- The operation of this instruction is implementation-dependent. Prefetch fill & evict policies may differ for other processor vendors or microarchitecture generations.

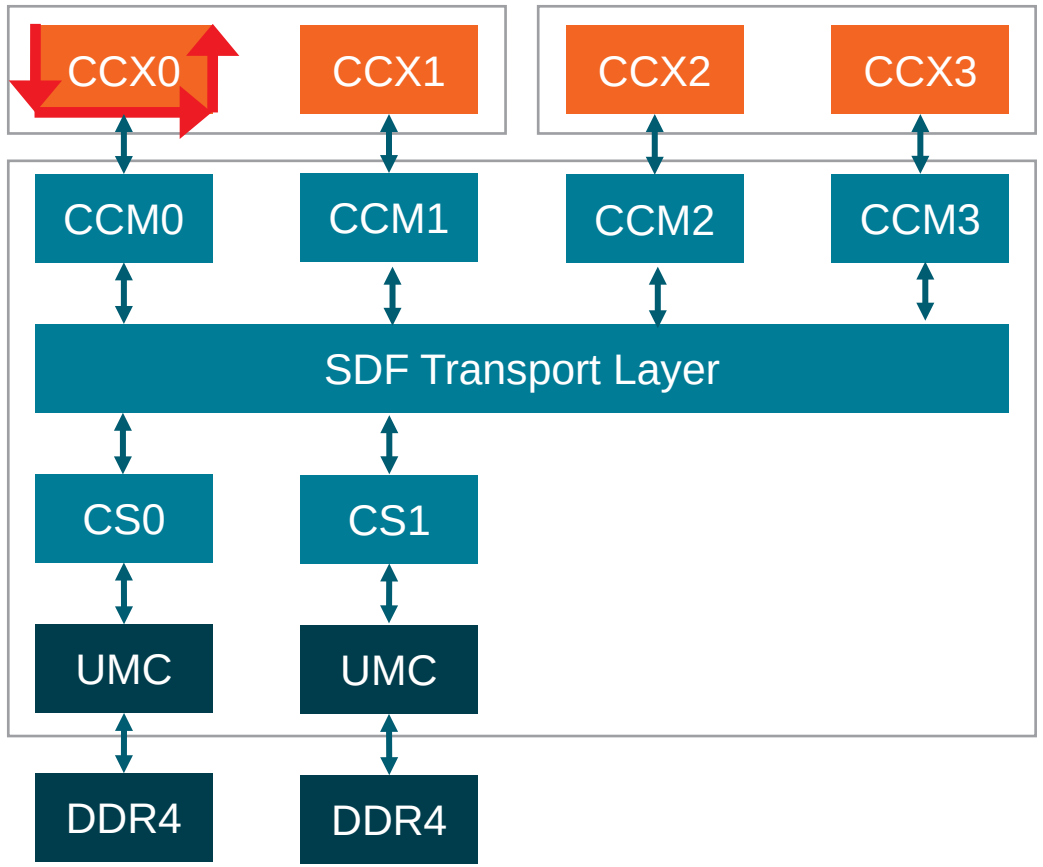
“MATISSE” CACHE AND MEMORY

CACHE LATENCY

Level	Count/ CCD	Capacity	Sets	Ways	Line Size	Latency
uop	8	4 K uops	64	8	8 uops	NA
L1I	8	32 KB	64	8	64 B	4 clocks
L1D	8	32 KB	64	8	64 B	4 clocks
L2U	8	512 KB	1024	8	64 B	12 clocks
L3U	2	16 MB	16384	16	64 B	39 clocks

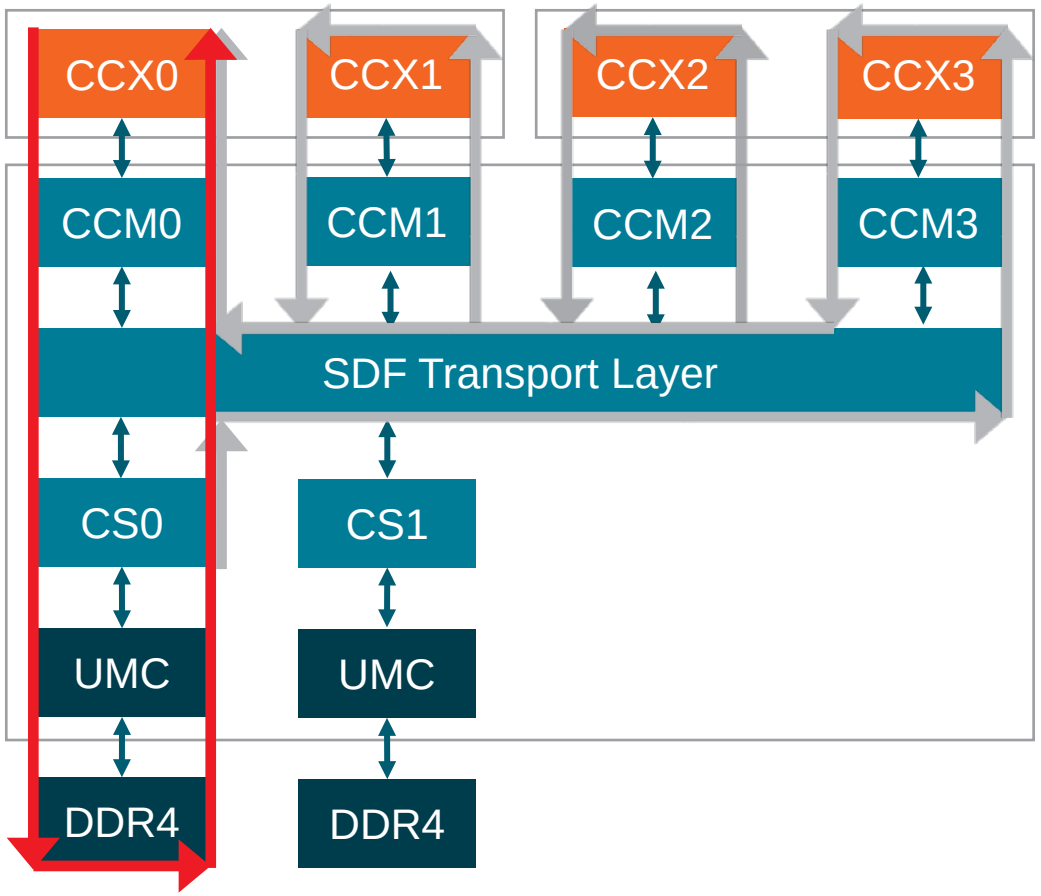
- Cache line size is 64 Bytes.
  - 2 cpu clock cycles to move a single cache line.
- L2 is inclusive of L1.
  - lines filled into L1 are also filled into L2.
- L3 is filled from L2 victims of all 4 cores within its CCX.
  - L2 tags are duplicated in its L3 for fast cache transfers within a CCX.
- L2 capacity evictions may cause L3 capacity evictions.
- “Matisse” products may have 1 or 2 CCDs.
  - Each CCD Core Complex Die (CCD) may have two CCX.
  - CCX: Core Complex (4 Cores, 8 Logical Processors, 16MB).

REFILL WITHIN SAME CCX



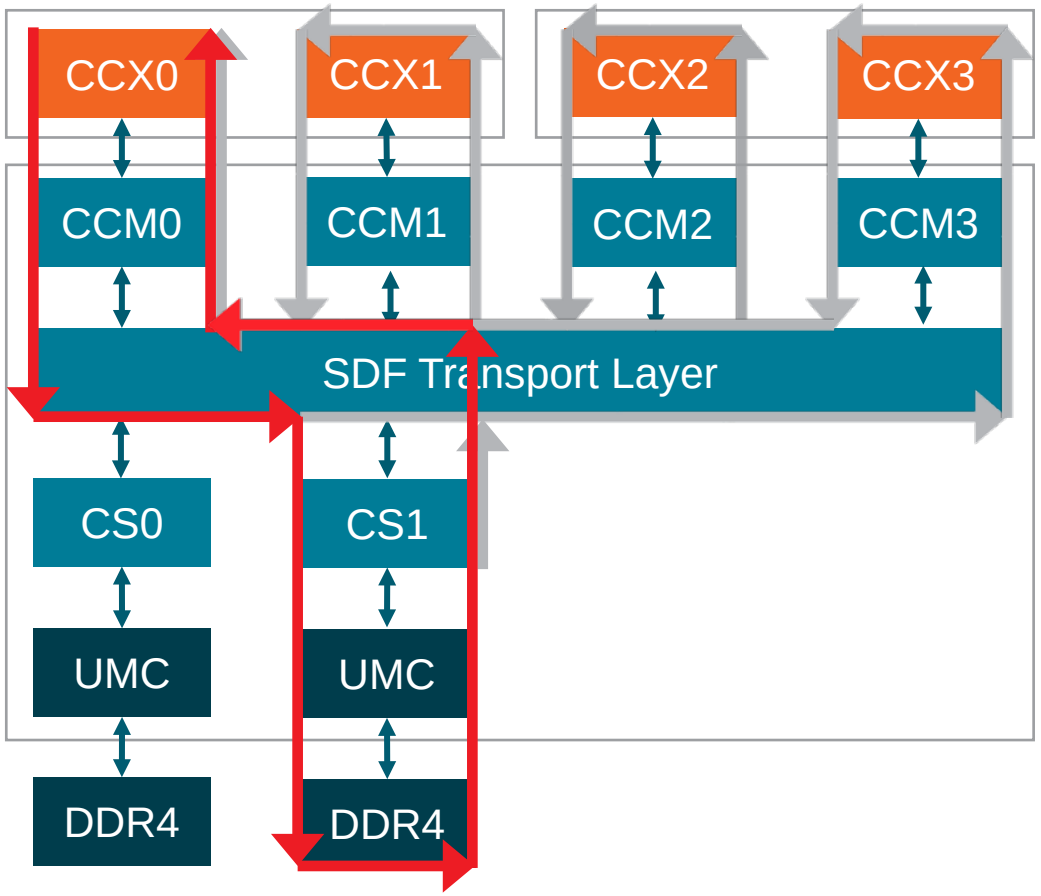
- Refills within the same CCX may be relatively low cost!
  - Some operating system schedulers are CCX aware.
- CCX: Core Complex (4 Cores, 8 Logical Processors, 16MB).
- IFOP: Infinity Fabric™ On-Package.
  - CCM: Cache-Coherent Master has the memory map.
  - SDF Transport Layer: Scalable Data Fabric Transport Layer.
  - CS: Coherent Slave responsible for cache coherency.
  - Electrical interface between chiplets not shown.
- UMC: Unified Memory Controller.

REFILL FROM LOCAL DRAM



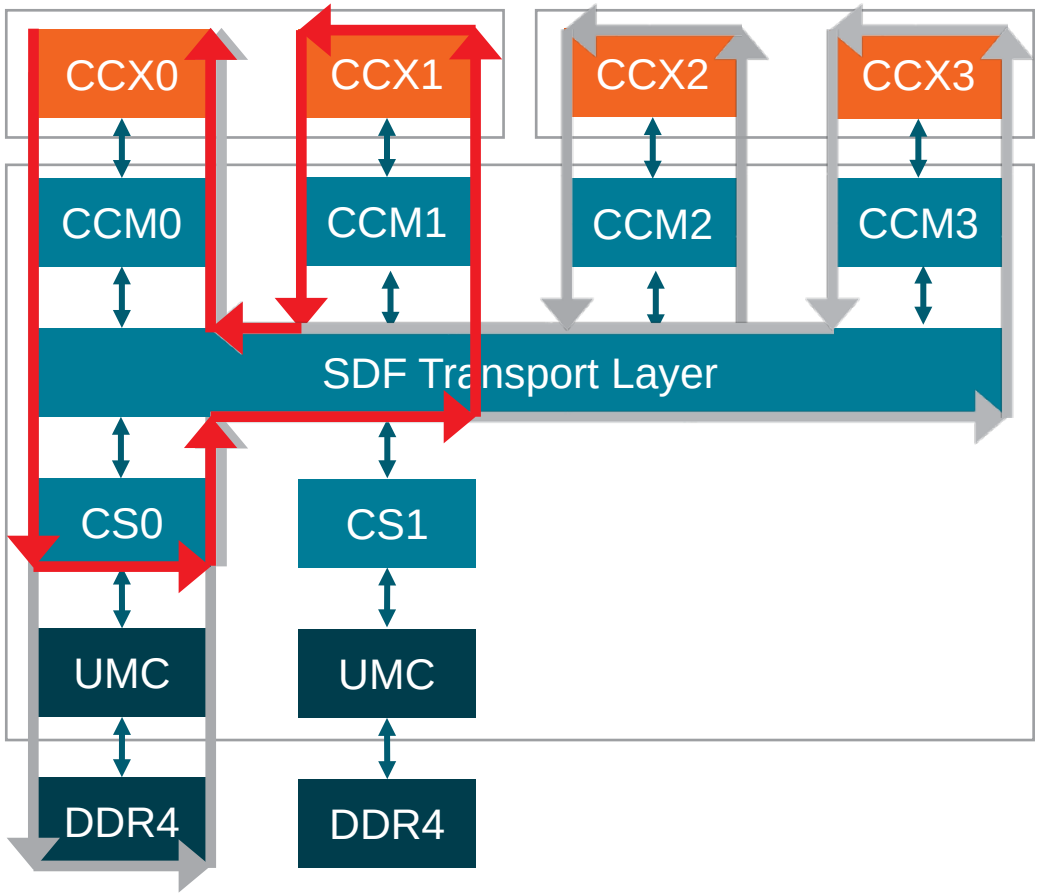
- Minimize refills from local DRAM.
- CCX: Core Complex (4 Cores, 8 Logical Processors, 16MB).
- IFOP: Infinity Fabric™ On-Package.
  - CCM: Cache-Coherent Master has the memory map.
  - SDF Transport Layer: Scalable Data Fabric Transport Layer.
  - CS: Coherent Slave responsible for cache coherency.
  - Electrical interface between chiplets not shown.
- UMC: Unified Memory Controller.

REFILL FROM LOCAL DRAM



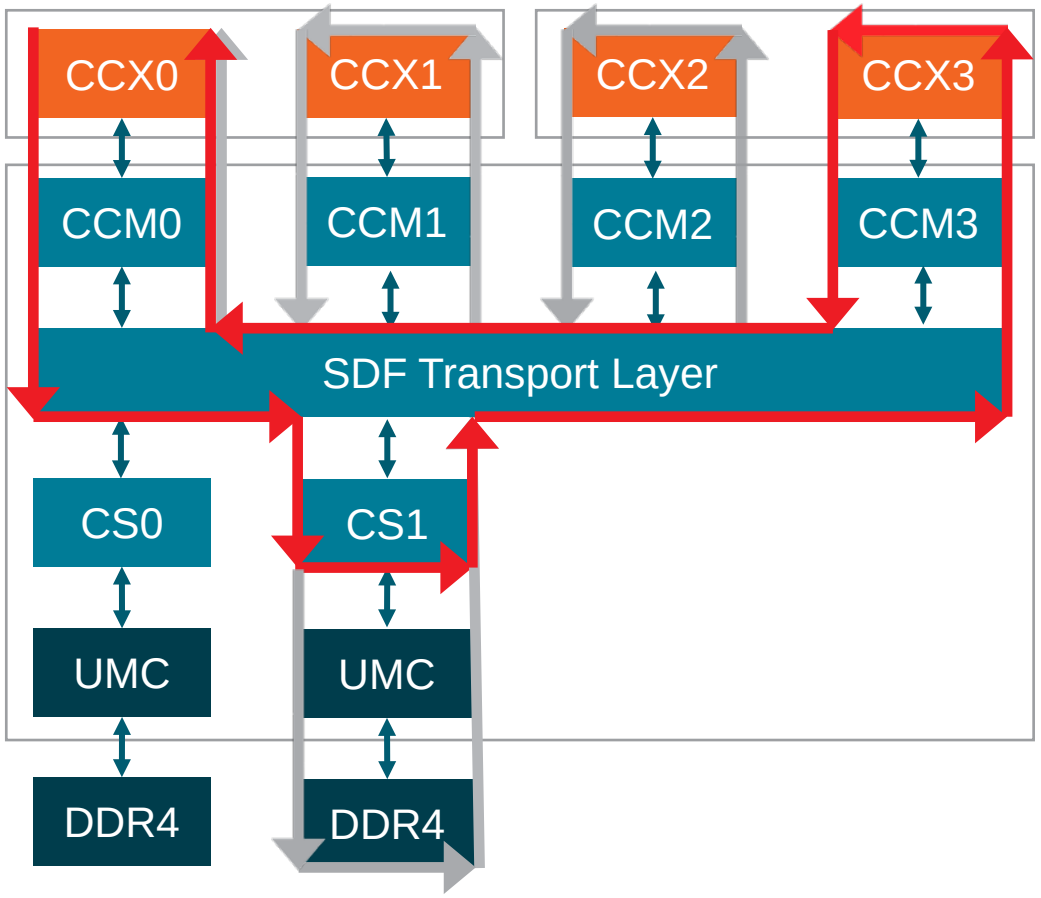
- Minimize refills from local DRAM.
- CCX: Core Complex (4 Cores, 8 Logical Processors, 16MB).
- IFOP: Infinity Fabric™ On-Package.
  - CCM: Cache-Coherent Master has the memory map.
  - SDF Transport Layer: Scalable Data Fabric Transport Layer.
  - CS: Coherent Slave responsible for cache coherency.
  - Electrical interface between chiplets not shown.
- UMC: Unified Memory Controller.

REFILL FROM ANY OTHER CCX



- Refill from any other CCX cost may be similar to memory latency.
- CCX: Core Complex (4 Cores, 8 Logical Processors, 16MB).
- IFOP: Infinity Fabric™ On-Package.
  - CCM: Cache-Coherent Master has the memory map.
  - SDF Transport Layer: Scalable Data Fabric Transport Layer.
  - CS: Coherent Slave responsible for cache coherency.
  - Electrical interface between chiplets not shown.
- UMC: Unified Memory Controller.

REFILL FROM ANY OTHER CCX

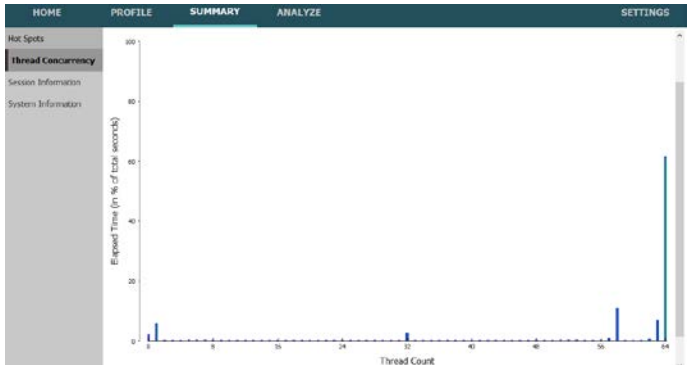


- Refill from any other CCX cost may be similar to memory latency.
- CCX: Core Complex (4 Cores, 8 Logical Processors, 16MB).
- IFOP: Infinity Fabric™ On-Package.
  - CCM: Cache-Coherent Master has the memory map.
  - SDF Transport Layer: Scalable Data Fabric Transport Layer.
  - CS: Coherent Slave responsible for cache coherency.
  - Electrical interface between chiplets not shown.
- UMC: Unified Memory Controller.

AMDUPROF PROFILER

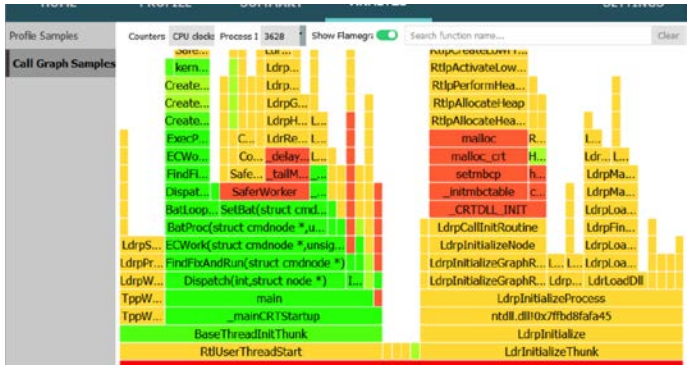
NEW IN V3.2

THREAD CONCURRENCY



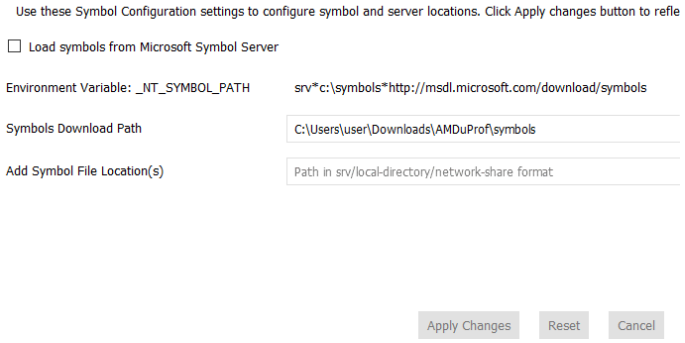
Scaled chart for Threadripper™

FLAME GRAPH

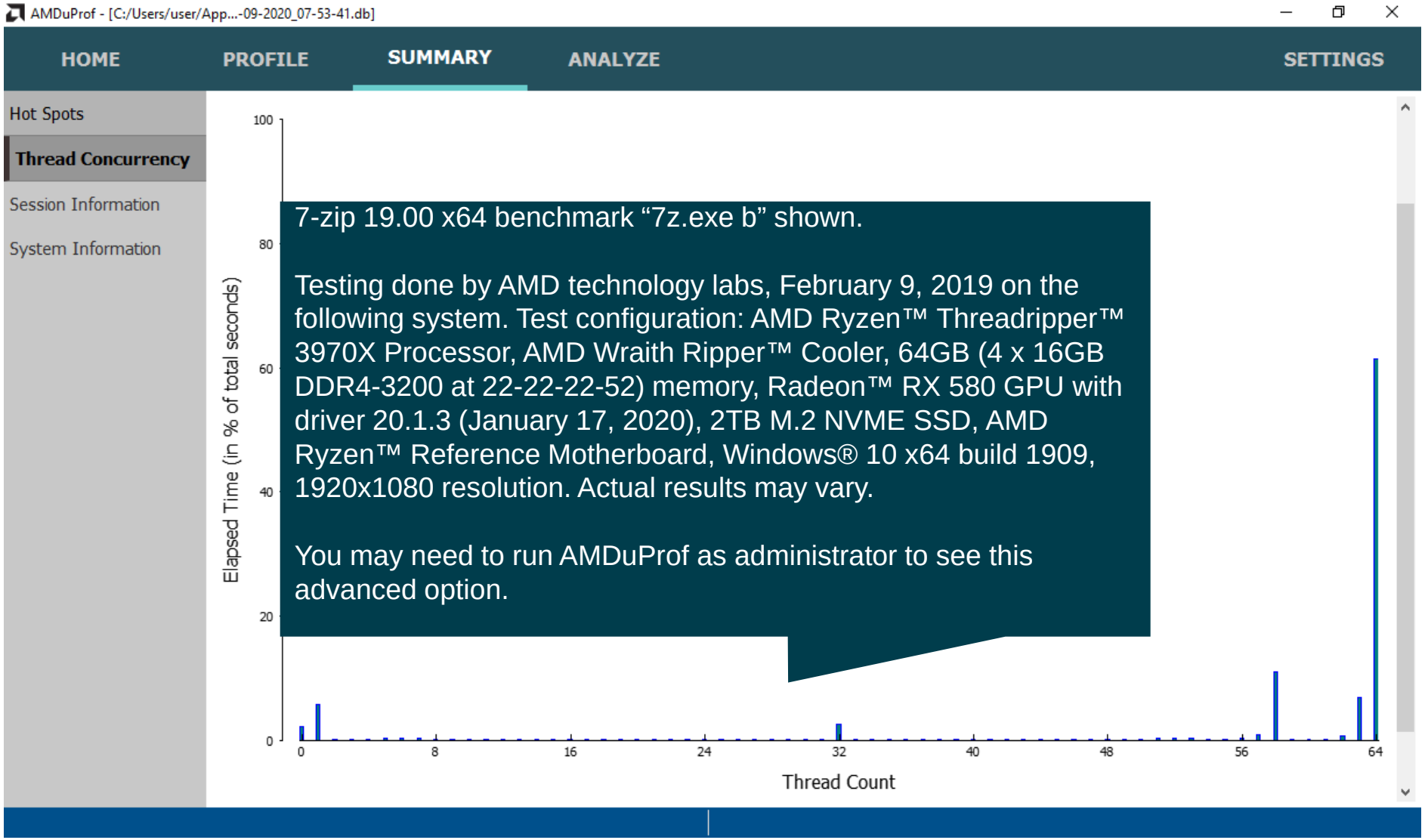


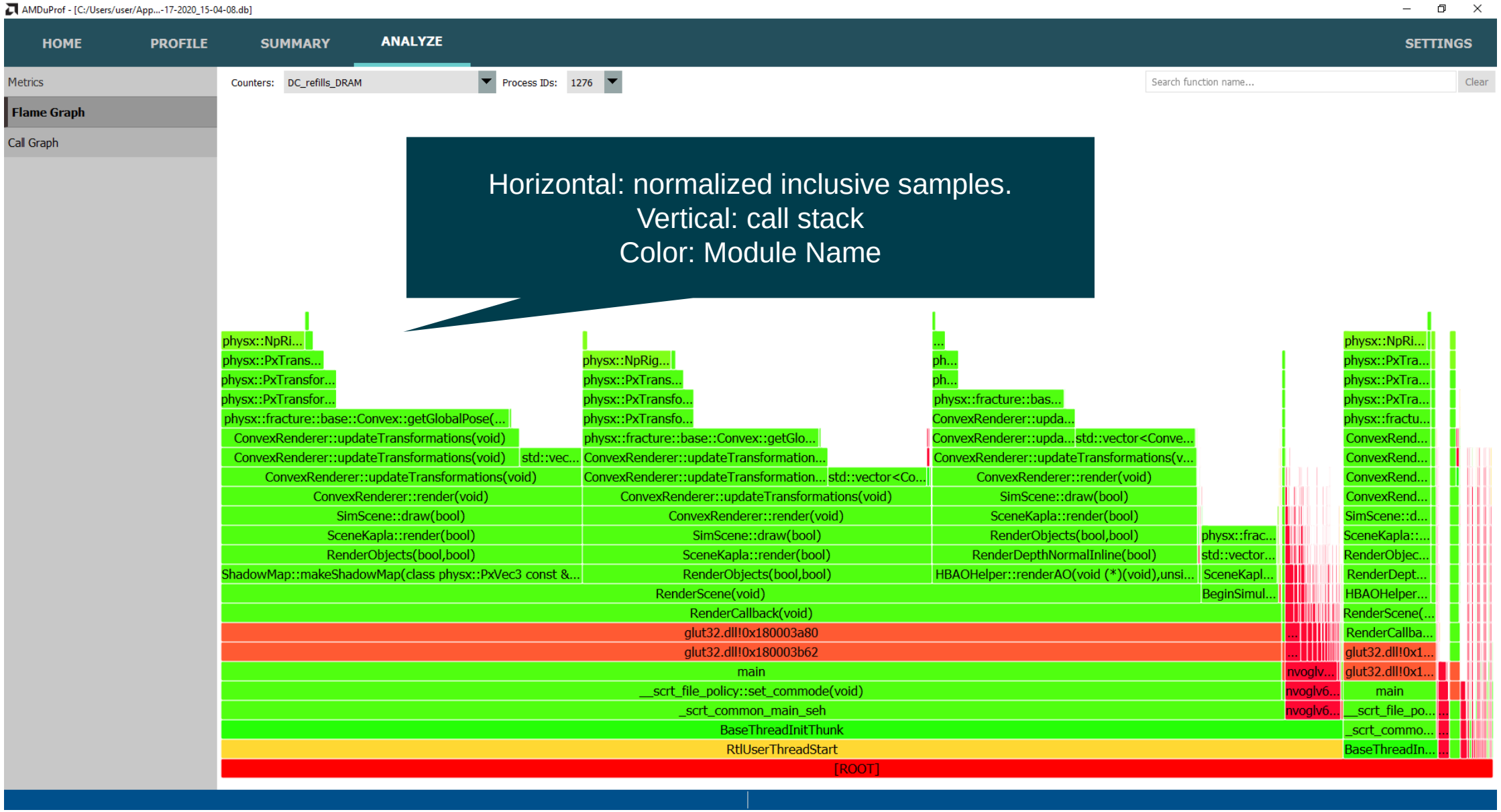
Sorted call stacks

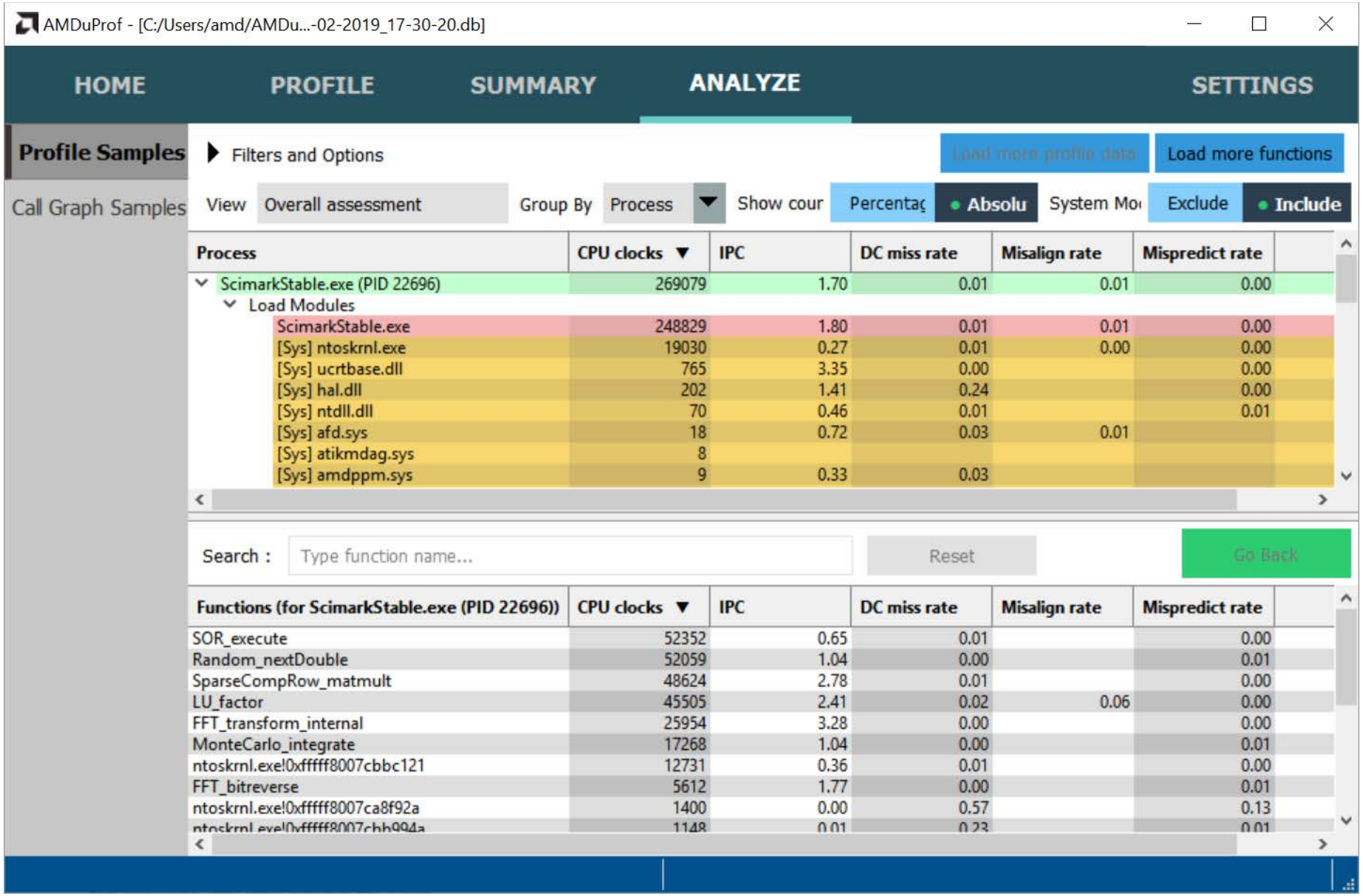
SYMBOLS

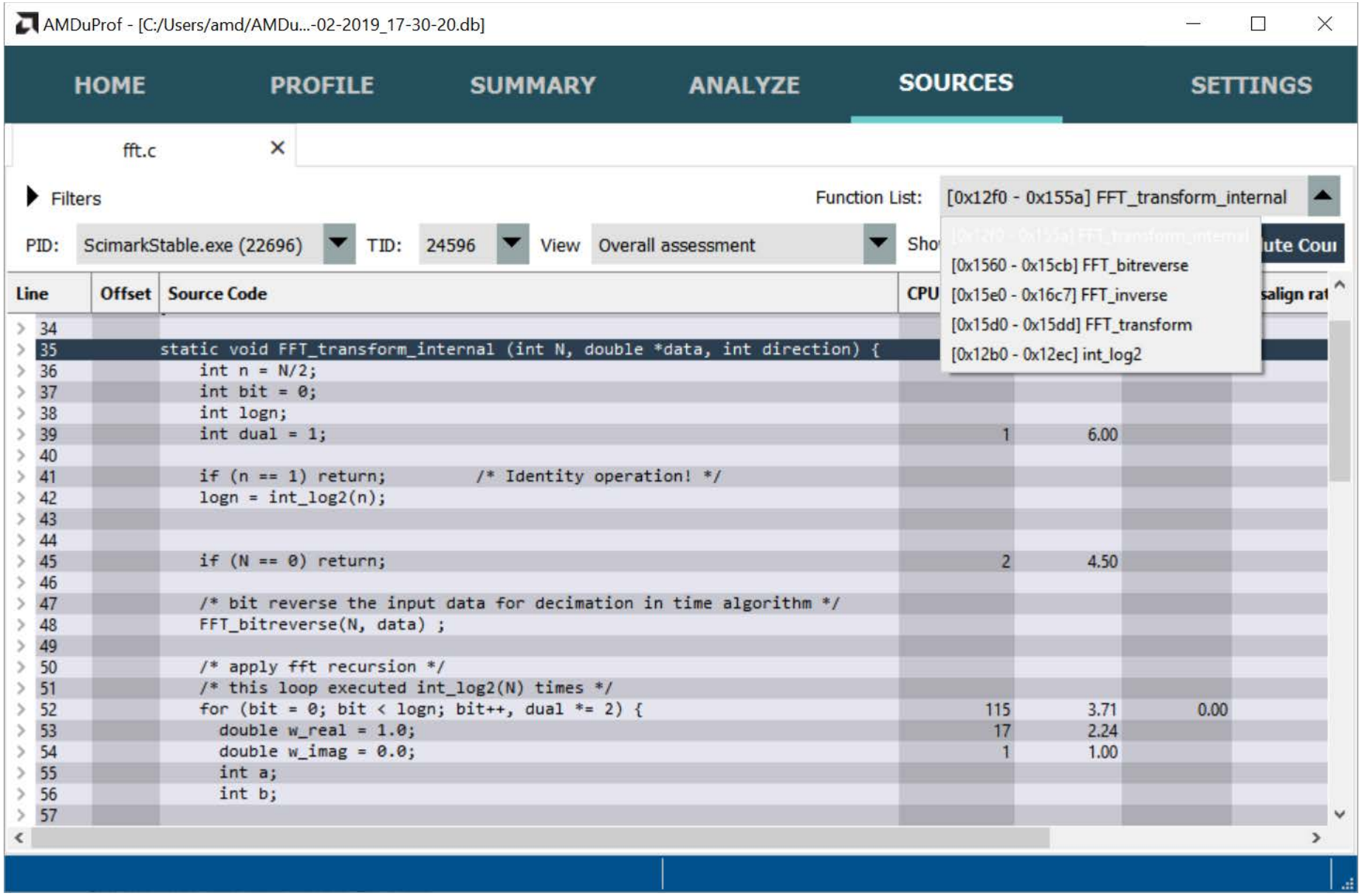


Improved symbol path support

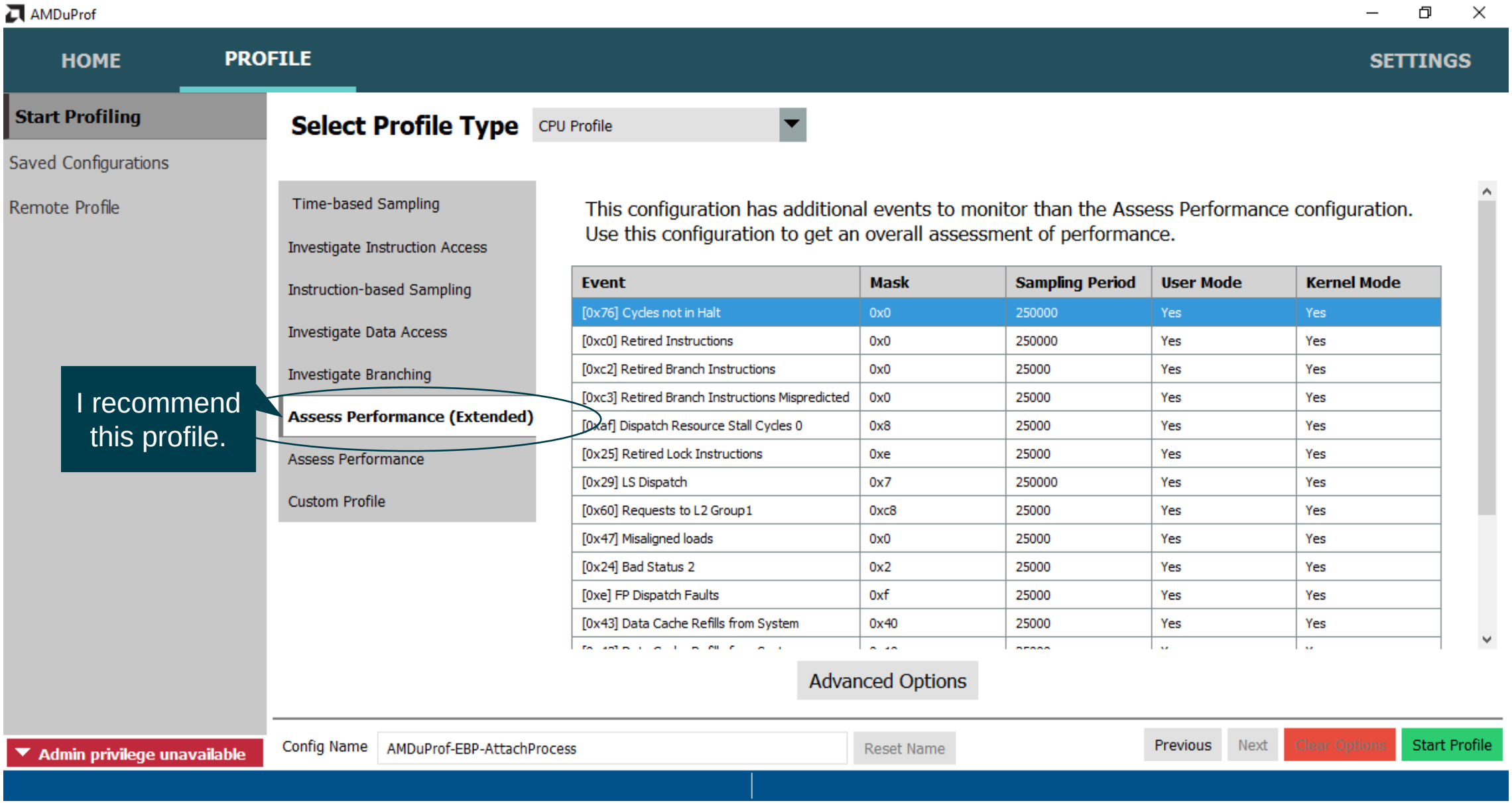

AMDuProf - [C:/Users/user/App...-09-2020_07-53-41.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Hot Spots
Thread Concurrency
Session Information
System Information
Elapsed Time (in % of total seconds)
100
80
60
40
20
0
0
8
16
24
32
40
48
56
64
Thread Count
7-zip 19.00 x64 benchmark “7z.exe b” shown.
Testing done by AMD technology labs, February 9, 2019 on the following system. Test configuration: AMD Ryzen™ Threadripper™ 3970X Processor, AMD Wraith Ripper™ Cooler, 64GB (4 x 16GB DDR4-3200 at 22-22-22-52) memory, Radeon™ RX 580 GPU with driver 20.1.3 (January 17, 2020), 2TB M.2 NVME SSD, AMD Ryzen™ Reference Motherboard, Windows® 10 x64 build 1909, 1920x1080 resolution. Actual results may vary.
You may need to run AMDuProf as administrator to see this advanced option.

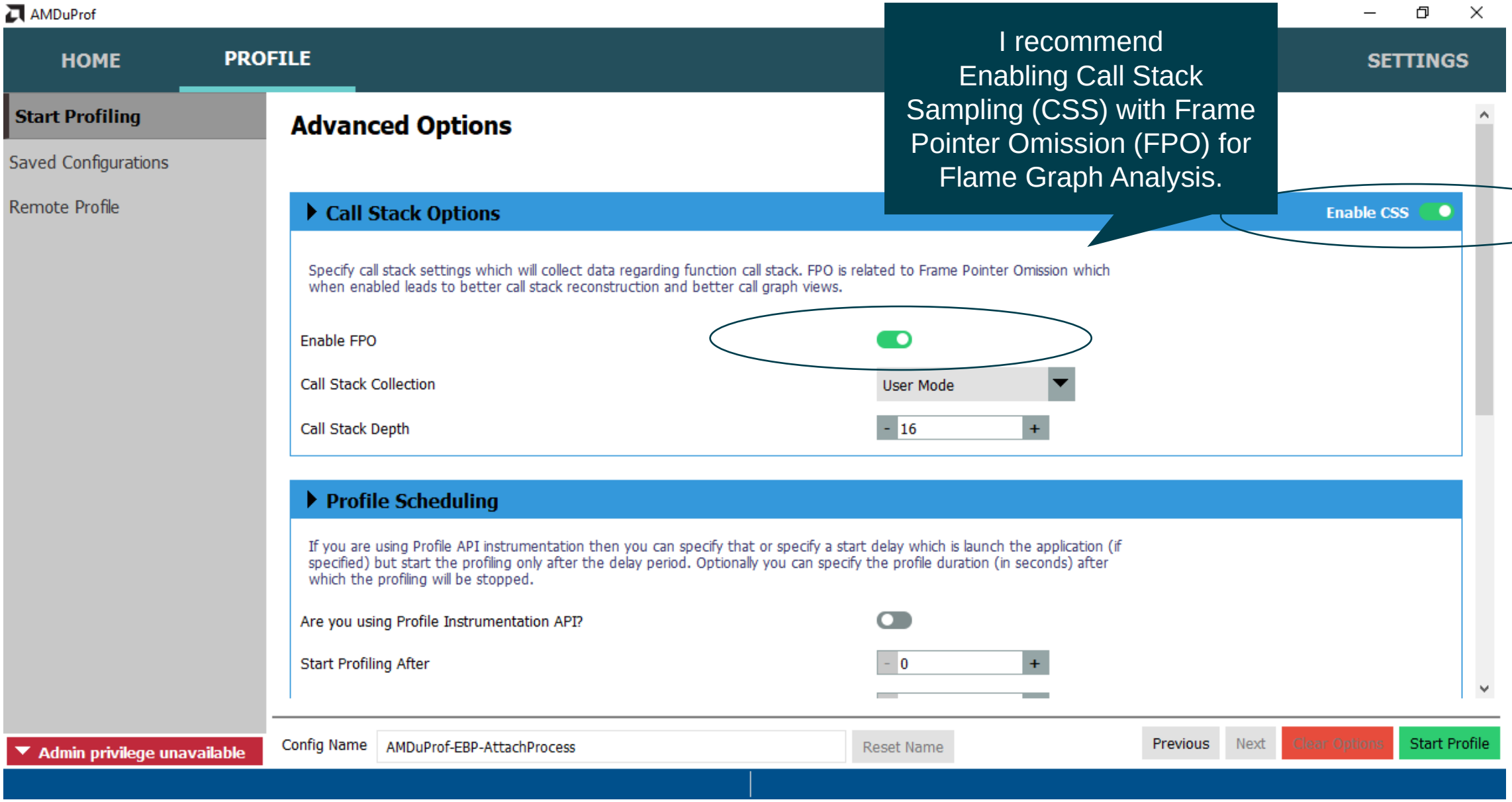

AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Metrics
Flame Graph
Call Graph
Counters:
DC_refills_DRAM
Process IDs:
1276
Search function name...
Clear
Horizontal: normalized inclusive samples.
Vertical: call stack
Color: Module Name
physx::NpRi...
physx::PxTrans...
physx::PxTransfor...
physx::PxTransfor...
physx::fracture::base::Convex::getGlobalPose(...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::updateTransformations(void)
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
ShadowMap::makeShadowMap(class physx::PxVec3 const &...
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commode(void)
_scrt_common_main_seh
BaseThreadInitThunk
RtlUserThreadStart
[ROOT]
physx::NpRig...
physx::PxTrans...
physx::PxTransfo...
physx::PxTransfo...
physx::fracture::base::Convex::getGlo...
ConvexRenderer::updateTransformation...
ConvexRenderer::updateTransformation...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderDepthNormalInline(bool)
HBAOHelper::renderAO(void (*)(void),unsi...
ph...
ph...
physx::fracture::bas...
ConvexRenderer::upda...
ConvexRenderer::upda...
ConvexRenderer::updateTransformations(v...
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderDepthNormalInline(bool)
HBAOHelper::renderAO(void (*)(void),unsi...
physx::NpRi...
physx::PxTra...
physx::PxTra...
physx::PxTra...
physx::fractu...
ConvexRend...
ConvexRend...
ConvexRend...
ConvexRend...
SimScene::d...
SceneKapla::...
RenderObjec...
RenderDept...
HBAOHelper...
RenderScene(...
RenderCallba...
glut32.dll!0x1...
glut32.dll!0x1...
main
__scrt_file_po...
_scrt_commo...
BaseThreadIn...

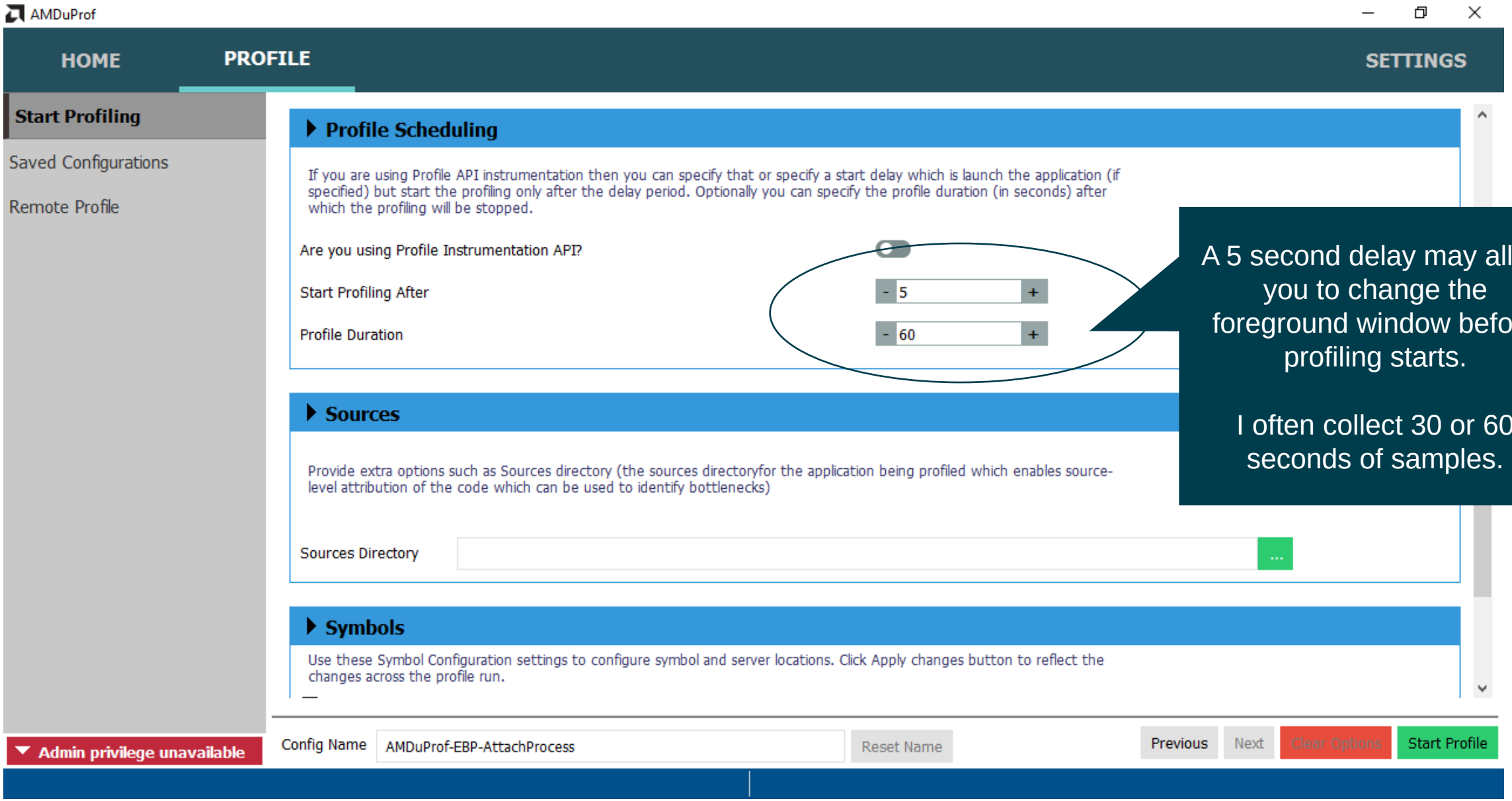

AMDuProf - [C:/Users/amd/AMDu...-02-2019_17-30-20.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Profile Samples
Call Graph Samples
Filters and Options
Load more profile data
Load more functions
View
Overall assessment
Group By
Process
Show cour
Percentag
Absolu
System Mo
Exclude
Include
Process
CPU clocks
IPC
DC miss rate
Misalign rate
Mispredict rate
ScimarkStable.exe (PID 22696) 269079 1.70 0.01 0.01 0.00
Load Modules
ScimarkStable.exe 248829 1.80 0.01 0.01 0.00
[Sys] ntoskrnl.exe 19030 0.27 0.01 0.00 0.00
[Sys] ucrtbase.dll 765 3.35 0.00 0.00
[Sys] hal.dll 202 1.41 0.24 0.00
[Sys] ntdll.dll 70 0.46 0.01 0.01
[Sys] afd.sys 18 0.72 0.03 0.01
[Sys] atikmdag.sys 8
[Sys] amdppm.sys 9 0.33 0.03
Search :
Type function name...
Reset
Go Back
Functions (for ScimarkStable.exe (PID 22696))
CPU clocks
IPC
DC miss rate
Misalign rate
Mispredict rate
SOR_execute 52352 0.65 0.01 0.00
Random_nextDouble 52059 1.04 0.00 0.01
SparseCompRow_matmult 48624 2.78 0.01 0.00
LU_factor 45505 2.41 0.02 0.06 0.00
FFT_transform_internal 25954 3.28 0.00 0.00
MonteCarlo_integrate 17268 1.04 0.00 0.01
ntoskrnl.exe!0xfffff8007cbbc121 12731 0.36 0.01 0.00
FFT_bitreverse 5612 1.77 0.00 0.01
ntoskrnl.exe!0xfffff8007ca8f92a 1400 0.00 0.57 0.13


AMDuProf - [C:/Users/amd/AMDu...-02-2019_17-30-20.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
fft.c
Filters
Function List:
[0x12f0 - 0x155a] FFT_transform_internal
PID:
ScimarkStable.exe (22696)
TID:
24596
View
Overall assessment
[0x1560 - 0x15cb] FFT_bitreverse
[0x15e0 - 0x16c7] FFT_inverse
[0x15d0 - 0x15dd] FFT_transform
[0x12b0 - 0x12ec] int_log2
Line
Offset
Source Code
CPU
34
35 static void FFT_transform_internal (int N, double *data, int direction) {
36 int n = N/2;
37 int bit = 0;
38 int logn;
39 int dual = 1; 1 6.00
40
41 if (n == 1) return; /* Identity operation! */
42 logn = int_log2(n);
43
44
45 if (N == 0) return; 2 4.50
46
47 /* bit reverse the input data for decimation in time algorithm */
48 FFT_bitreverse(N, data) ;
49
50 /* apply fft recursion */
51 /* this loop executed int_log2(N) times */
52 for (bit = 0; bit < logn; bit++, dual *= 2) { 115 3.71 0.00
53 double w_real = 1.0; 17 2.24
54 double w_imag = 0.0; 1 1.00
55 int a;
56 int b;
57


AMDuProf
HOME
PROFILE
TIMECHART
SETTINGS
Graph Visibility
Energy [Parent Counter]
Temperature
System #0
Energy
System #0
Controllers
System #0
Millijoules
01:4.3
System [Energy]
Counter
Value
Socket0 Package energy
577.88
Celsius
01:4.3
System [Temperature] #0
Counter
Value
Socket0 VDDCR Soc Temperature
52.63
Millijoules
01:4.3
Counter
Value
Core0 Energy
37.69
Core1 Energy
19.84
Core2 Energy
22.13
Core3 Energy
112.34
Play Graphs
Stop Profiling
Close View

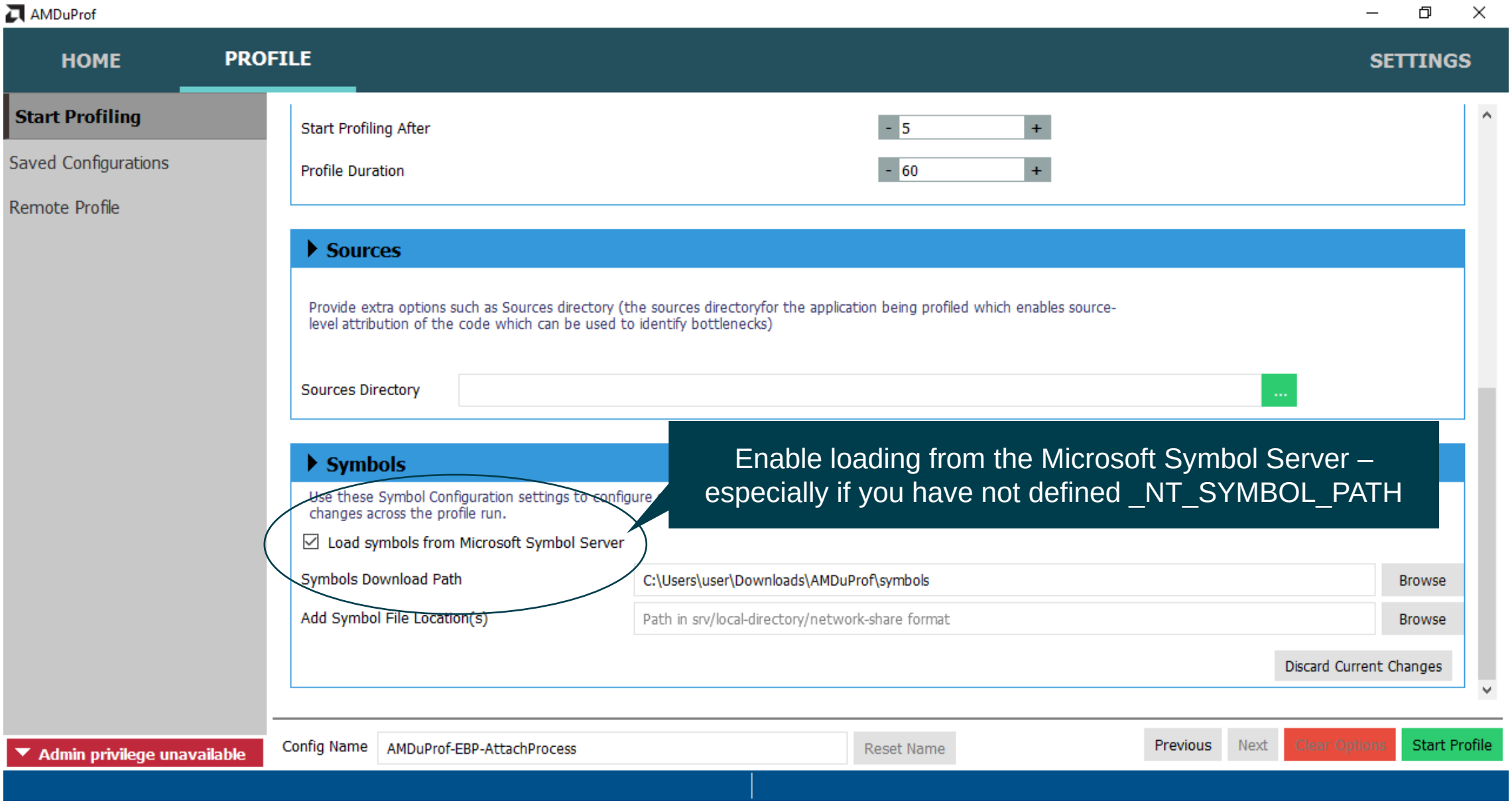

AMDuProf
HOME
PROFILE
SETTINGS
Start Profiling
Saved Configurations
Remote Profile
Select Profile Type
CPU Profile
Time-based Sampling
Investigate Instruction Access
Instruction-based Sampling
Investigate Data Access
Investigate Branching
Assess Performance (Extended)
Assess Performance
Custom Profile
I recommend this profile.
This configuration has additional events to monitor than the Assess Performance configuration.
Use this configuration to get an overall assessment of performance.
Event | Mask | Sampling Period | User Mode | Kernel Mode
[0x76] Cycles not in Halt | 0x0 | 250000 | Yes | Yes
[0xc0] Retired Instructions | 0x0 | 250000 | Yes | Yes
[0xc2] Retired Branch Instructions | 0x0 | 25000 | Yes | Yes
[0xc3] Retired Branch Instructions Mispredicted | 0x0 | 25000 | Yes | Yes
[0xaf] Dispatch Resource Stall Cycles 0 | 0x8 | 25000 | Yes | Yes
[0x25] Retired Lock Instructions | 0xe | 25000 | Yes | Yes
[0x29] LS Dispatch | 0x7 | 250000 | Yes | Yes
[0x60] Requests to L2 Group1 | 0xc8 | 25000 | Yes | Yes
[0x47] Misaligned loads | 0x0 | 25000 | Yes | Yes
[0x24] Bad Status 2 | 0x2 | 25000 | Yes | Yes
[0xe] FP Dispatch Faults | 0xf | 25000 | Yes | Yes
[0x43] Data Cache Refills from System | 0x40 | 25000 | Yes | Yes
Advanced Options
Admin privilege unavailable
Config Name
AMDuProf-EBP-AttachProcess
Reset Name
Previous
Next
Clear Options
Start Profile


AMDuProf
HOME
PROFILE
SETTINGS
Start Profiling
Saved Configurations
Remote Profile
Advanced Options
I recommend Enabling Call Stack Sampling (CSS) with Frame Pointer Omission (FPO) for Flame Graph Analysis.
Call Stack Options
Enable CSS
Specify call stack settings which will collect data regarding function call stack. FPO is related to Frame Pointer Omission which when enabled leads to better call stack reconstruction and better call graph views.
Enable FPO
Call Stack Collection
User Mode
Call Stack Depth
16
Profile Scheduling
If you are using Profile API instrumentation then you can specify that or specify a start delay which is launch the application (if specified) but start the profiling only after the delay period. Optionally you can specify the profile duration (in seconds) after which the profiling will be stopped.
Are you using Profile Instrumentation API?
Start Profiling After
0
Admin privilege unavailable
Config Name
AMDuProf-EBP-AttachProcess
Reset Name
Previous
Next
Clear Options
Start Profile



A 5 second delay may allow you to change the foreground window before profiling starts.

I often collect 30 or 60 seconds of samples.


AMDuProf
HOME
PROFILE
SETTINGS
Start Profiling
Saved Configurations
Remote Profile
Start Profiling After
5
Profile Duration
60
Sources
Provide extra options such as Sources directory (the sources directoryfor the application being profiled which enables source-level attribution of the code which can be used to identify bottlenecks)
Sources Directory
Symbols
Use these Symbol Configuration settings to configure changes across the profile run.
Load symbols from Microsoft Symbol Server
Symbols Download Path
C:\Users\user\Downloads\AMDuProf\symbols
Browse
Add Symbol File Location(s)
Path in srv/local-directory/network-share format
Browse
Discard Current Changes
Enable loading from the Microsoft Symbol Server – especially if you have not defined _NT_SYMBOL_PATH
Admin privilege unavailable
Config Name
AMDuProf-EBP-AttachProcess
Reset Name
Previous
Next
Clear Options
Start Profile

OPTIMIZATIONS AND LESSONS LEARNED

TOPICS

- General Guidance
- Use Best Practices with Scalability
- Verify Parallel DX12 Pipeline State Creation
- Verify Parallel DX12 Command List Generation
- Use Best Practices with Locks
- Reorder Hot Struct Members
- Use Prefetch Level while iterating std::vector<T*>

GENERAL GUIDANCE

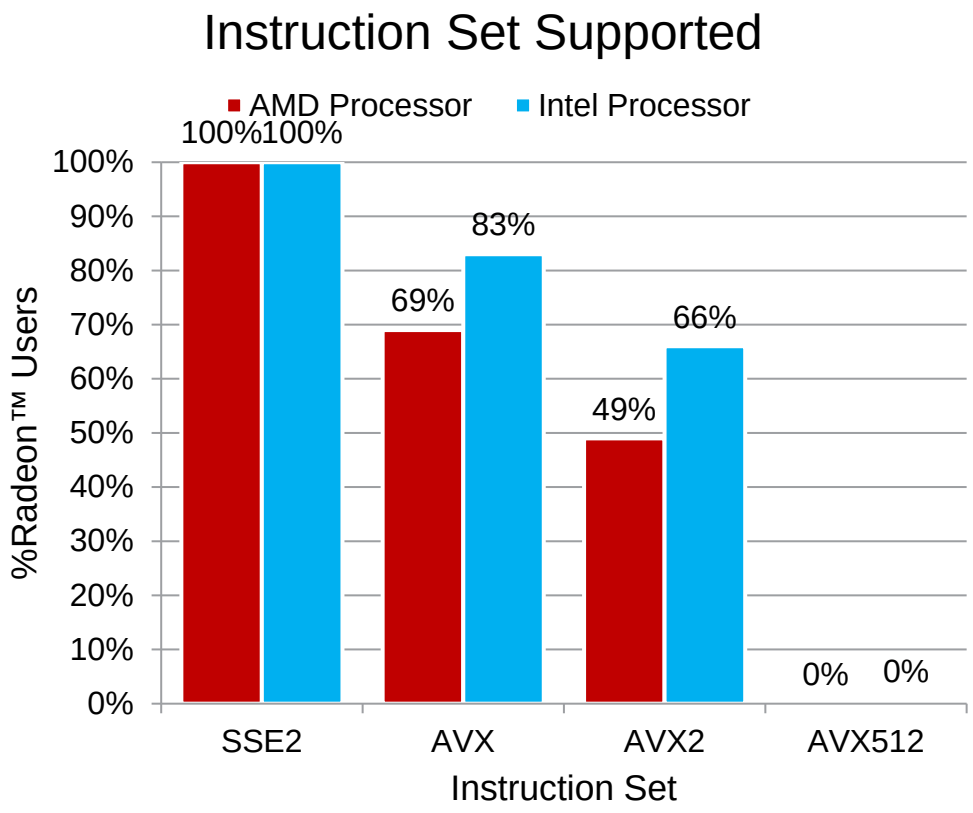
USE THE LATEST COMPILER & SDK

- "Zen 2" recommended compiler flags:
  - /GL /arch:AVX2 /MT /fp:fast /favor:blend
- JeMalloc may benefit some applications.
  - See http://jemalloc.net/

Guidance for "Zen 2" and subject to change.
Use /favor:blend and NOT /favor:amd64.

Year	Visual Studio Changes	AMD Products (implicit)
2019	Additional SIMD intrinsics optimizations including constant-folding and arithmetic simplifications. Build throughput improvements. New -Ob3 inlining option. Memcpy & Memset optimizations.	"Pinnacle Ridge"
2017	**Update v15.9.14 and later may improve AMD Ryzen™ memcpy/memset performance.** Improved code generation of loops: Support for automatic vectorization of division of constant integers, better identification of memset patterns. Added Cmake support. Added faster database engine. Improved STL & .NET optimizations. New /Qspectre option.	"Summit Ridge"
2015	Improved autovectorization & scalar optimizations. Faster build times with /LTCG:incremental. Added assembly optimized memset & memcpy using ERMS & SSE2.	"Kaveri"

USE A SUPPORTED INSTRUCTION SET



- Using /arch:AVX or /arch:AVX2 may improve code gen of inline code.
  - memcpy & memset may be inline if the length is known at compile time.
- AVX is supported on many systems and growing over time.
- AVX512 is not supported by AMD processors and was present on less than 1% of users with Intel processors.
- Source: AMD User Experience Program Users Survey including 4 Million systems sampled from January 2019 to October 2019.

USE ALL PHYSICAL CORES

- This advice is specific to AMD processors and is not general guidance for all processor vendors.
- Generally, applications show SMT benefits and use of all logical processors is recommended.
- However, games often suffer from SMT contention on the main or render threads during gameplay.
  - One strategy to reduce this contention is to create threads based on physical core count rather than logical processor count.
  - Profile your application/game to determine the ideal thread count.
  - Recommend game options to:
    - Set Max Thread Pool Size
    - Force Thread Pool Size
    - Force SMT
    - Force Single NUMA Node (implicitly Group)
  - Avoid setting thread pool size as a constant.
- See https://gpuopen.com/cpu-core-count-detection-windows/

```
// This advice is specific to AMD processors and is
// not general guidance for all processor vendors
DWORD get_default_thread_count() {
  DWORD cores, logical;
  get_processor_count(cores, logical);
  DWORD count = logical;
  char vendor[13];
  get_cpuid_vendor(vendor);
  if (0 == strcmp(vendor, "AuthenticAMD")) {
    if (0x15 == get_cpuid_family()) {
      // AMD "Bulldozer" family microarchitecture
      count = logical;
    } else {
      count = cores;
    }
  }
  return count;
}
```

DISABLE DEBUG FEATURES BEFORE YOU SHIP

- While investigating open issues, developers may submit change requests which enable debug features on Test and Shipping configurations. These debug features may greatly reduce performance due to disabling multi-threading, cache pollution from STATS, and increased serialization from logging.
- Some Unreal Engine settings to verify include:
  - Build.h #define FORCE_USE_STATS and #define STATS
    - See 4.24/Engine/Source/Runtime/Core/Public/Misc/Build.h
  - Parallel Rendering CVARS
    - See 4.24/Engine/Source/Runtime/RHI/Private/RHICommandList.cpp
    - See https://docs.unrealengine.com/en-US/Programming/Rendering/ParallelRendering

Command	Recommended Value
r.rhicmdbypass	0
r.rhicmdusedeferredcontexts	1
r.rhicmduseparallelalgorithms	1
r.rhithread.enable	1

USE BEST PRACTICES WITH SCALABILITY

OPTIMIZE SCALABILITY FOR INTEGRATED GRAPHICS

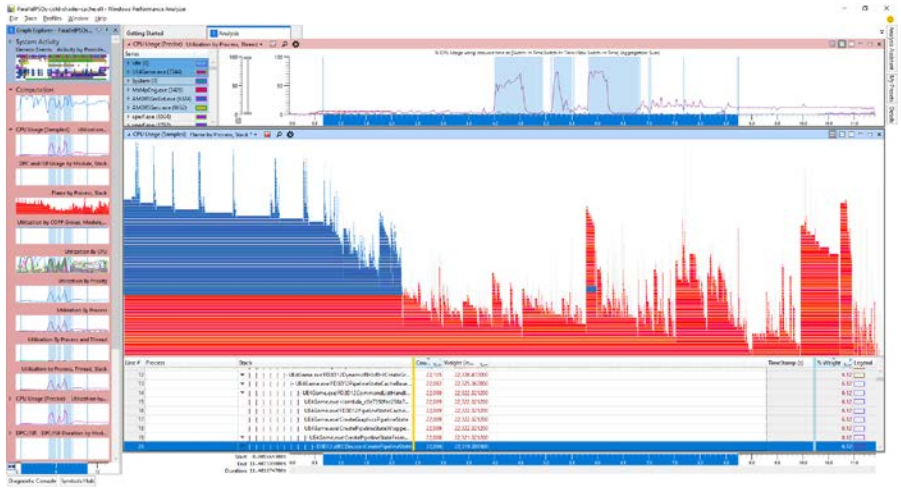
- Goal: >= 60 FPS Average at 720p 100% Very Low
- Try:
  - Use DXGI_FORMAT_R11G11B10_FLOAT rather than DXGI_FORMAT_R16G16B16A16_FLOAT
  - Reduce shadow map quality
  - Reduce volumetric fog quality
  - Disable Ambient Occlusion
- For Unreal Engine
  - r.SceneColorFormat
  - r.AmbientOcclusionLevels=0

USE PROPER VIDEO MEMORY BUDGET FOR APU

- AGS SDK 5.4
  - Added isAPU flag.
    - If true, set the video memory budget to sharedMemoryInBytes for APU (AMD Accelerated Processing Unit with integrated graphics).
    - If false, set the video memory budget to localMemoryInBytes for discrete GPU.
  - Example:
    - unsigned long long memory_budget = (device.isAPU)? device.sharedMemoryInBytes: device.localMemoryInBytes;
  - See https://gpuopen.com/ags-sdk-5-4-improves-handling-video-memory-reporting-apus/

VERIFY PARALLEL DX12 PIPELINE STATE CREATION

VERIFY PARALLEL DX12 PIPELINE STATE CREATION

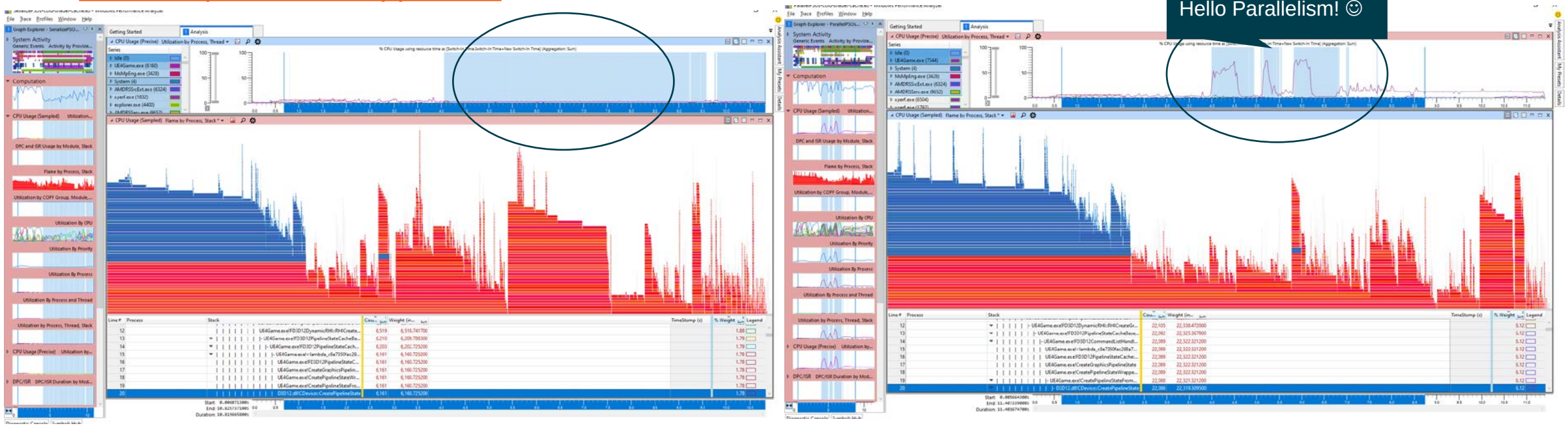


- Game shows parallel DX12 Pipeline State Creation.
- Performance of binary compiled with:
  - Microsoft® Visual Studio 2019 v16.4.5.
  - UnrealEngine-4.24.2-release from https://github.com/EpicGames/UnrealEngine
  - Windows (64-bit) Packaged Project “Infiltrator Demo” from Epic Games Store; https://www.unrealengine.com/marketplace/en-US/product/infiltrator-demo
- Testing done by AMD technology labs, February 18, 2020 on the following system. Test configuration: AMD Ryzen™ 9 3950X Processor, AMD Wraith Prism Cooler, 16GB (2 x 8GB DDR4-3200 at 22-22-22-52) memory, Radeon™ VII GPU with driver 20.1.4 (January 24, 2020), 2TB M.2 NVME SSD, AMD Ryzen™ Reference Motherboard, Windows® 10 x64 build 1909, 1920x1080 resolution. Actual results may vary.

UE4.24 PARALLELIZED DX12 PIPELINE STATE CREATION ☺

Before
https://github.com/EpicGames/UnrealEngine/blob/4.23/Engine/Source/Runtime/D3D12RHI/Private/D3D12Pipelinestate.cpp#L473

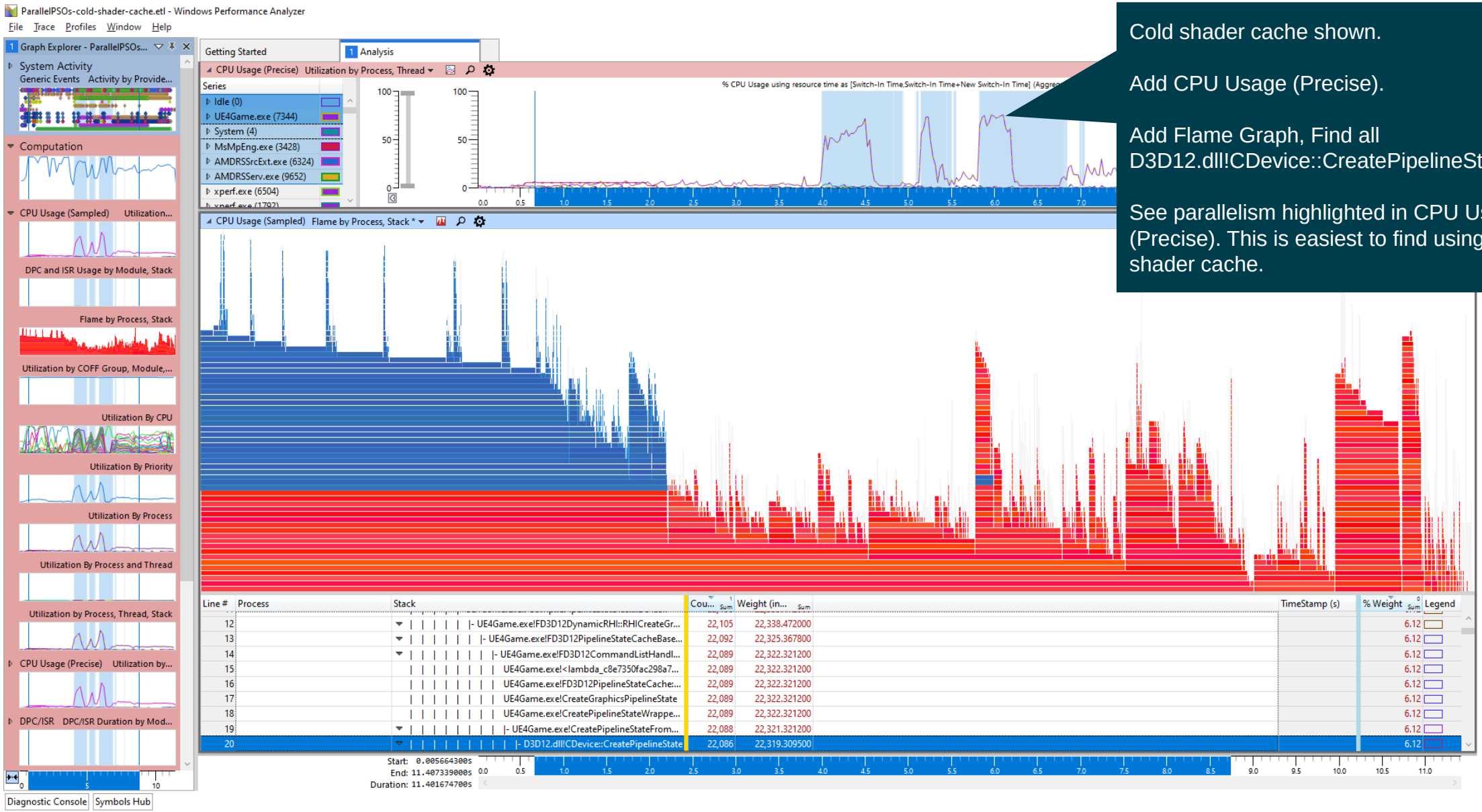
After
https://github.com/EpicGames/UnrealEngine/blob/4.24/Engine/Source/Runtime/D3D12RHI/Private/D3D12Pipelinestate.cpp#L488



TEST COLD SHADER CACHE

- Using a cold shader cache may simplify verifying if D3D12.dll!CDevice::CreatePipelineState was called in parallel.
- Install the Windows® SDK Windows® Performance Toolkit. Add the GPUView folder to the PATH.
- Applications and games may vary configurations of shader caches on disk yielding different results.
- Results may vary based on GPU vendor & driver versions used.

```
rmdir /s /q "%LOCALAPPDATA%\D3DSCache"
rmdir /s /q "%LOCALAPPDATA%\AMD\DxCache"
rmdir /s /q "%LOCALAPPDATA%\AMD\VkCache"
rmdir /s /q "%LOCALAPPDATA%\AMD\GLCache"

call log.cmd

pushd "C:\WindowsNoEditor"
start InfiltratorDemo.exe -dx12
popd

timeout.exe /t 10

call log.cmd
```

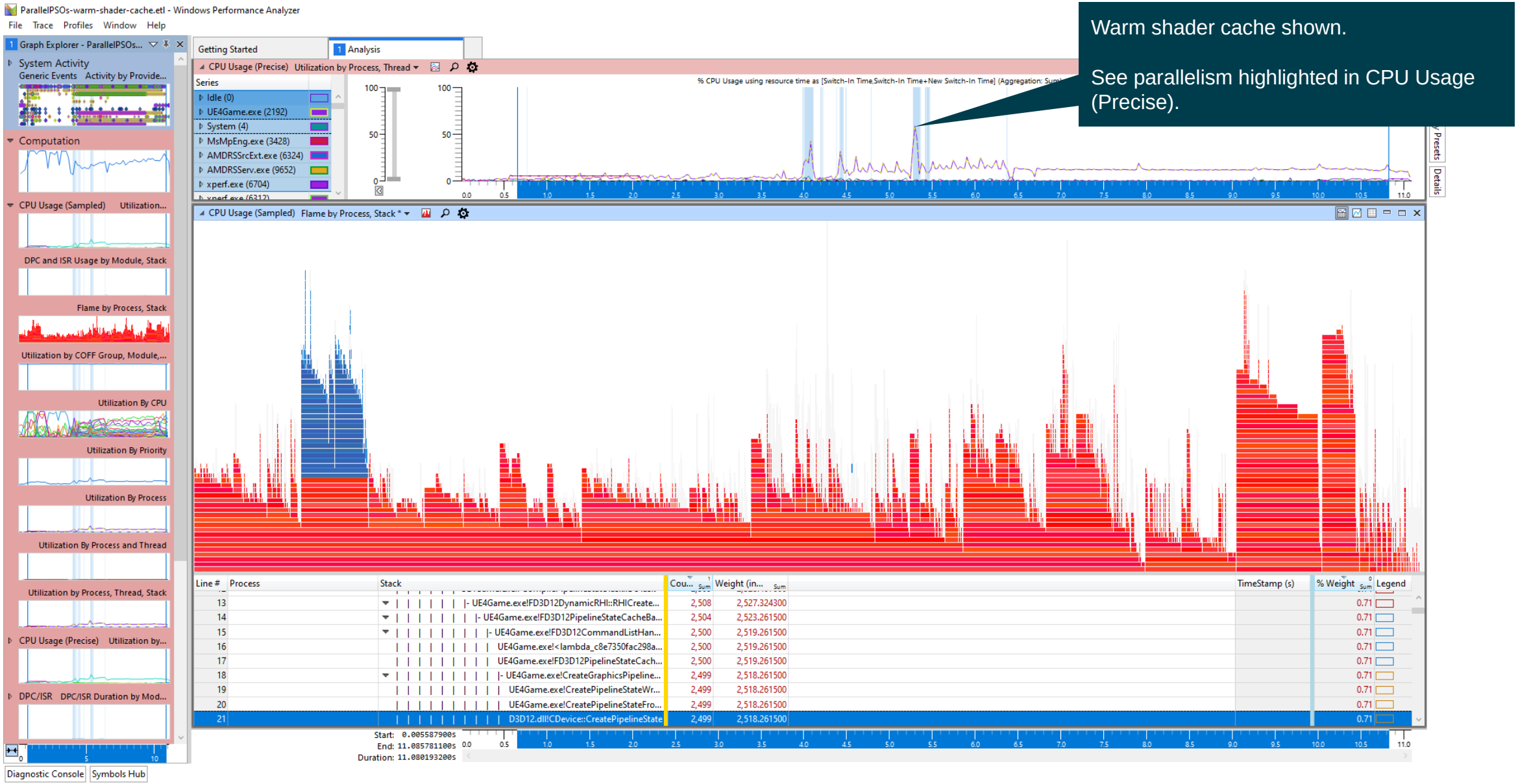


Cold shader cache shown.

Add CPU Usage (Precise).

Add Flame Graph, Find all D3D12.dll!CDevice::CreatePipelineState.

See parallelism highlighted in CPU Usage (Precise). This is easiest to find using a cold shader cache.

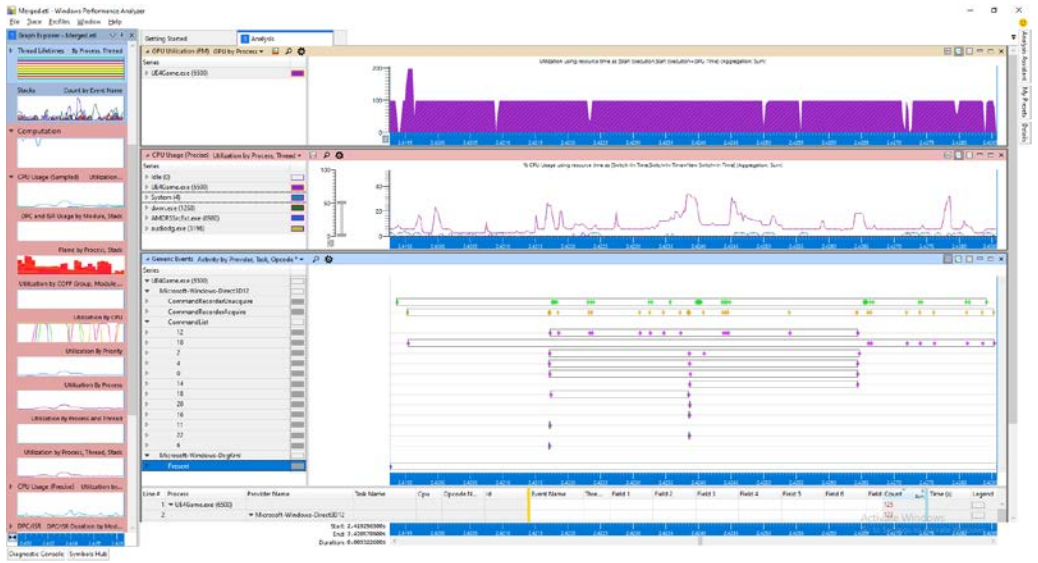


Warm shader cache shown.

See parallelism highlighted in CPU Usage (Precise).

VERIFY PARALLEL DX12 COMMAND LIST GENERATION

VERIFY PARALLEL DX12 COMMAND LIST GENERATION



- Game shows parallel DX12 Command List Generation.
- Performance of binary compiled with:
  - Microsoft® Visual Studio 2019 v16.4.5.
  - UnrealEngine-4.24.2-release from https://github.com/EpicGames/UnrealEngine
  - Windows (64-bit) Packaged Project “Infiltrator Demo” from Epic Games Store; https://www.unrealengine.com/marketplace/en-US/product/infiltrator-demo
- Testing done by AMD technology labs, February 13, 2020 on the following system. Test configuration: AMD Ryzen™ 9 3950X Processor, AMD Wraith Prism Cooler, 16GB (2 x 8GB DDR4-3200 at 22-22-22-52) memory, Radeon™ VII GPU with driver 20.1.4 (January 24, 2020), 2TB M.2 NVME SSD, AMD Ryzen™ Reference Motherboard, Windows® 10 x64 build 1909, 1920x1080 resolution. Actual results may vary.

Run:
InfiltratorDemo.exe -dx12

Run as admin:
timeout.exe /t 5
call log.cmd
timeout.exe /t 3
call log.cmd

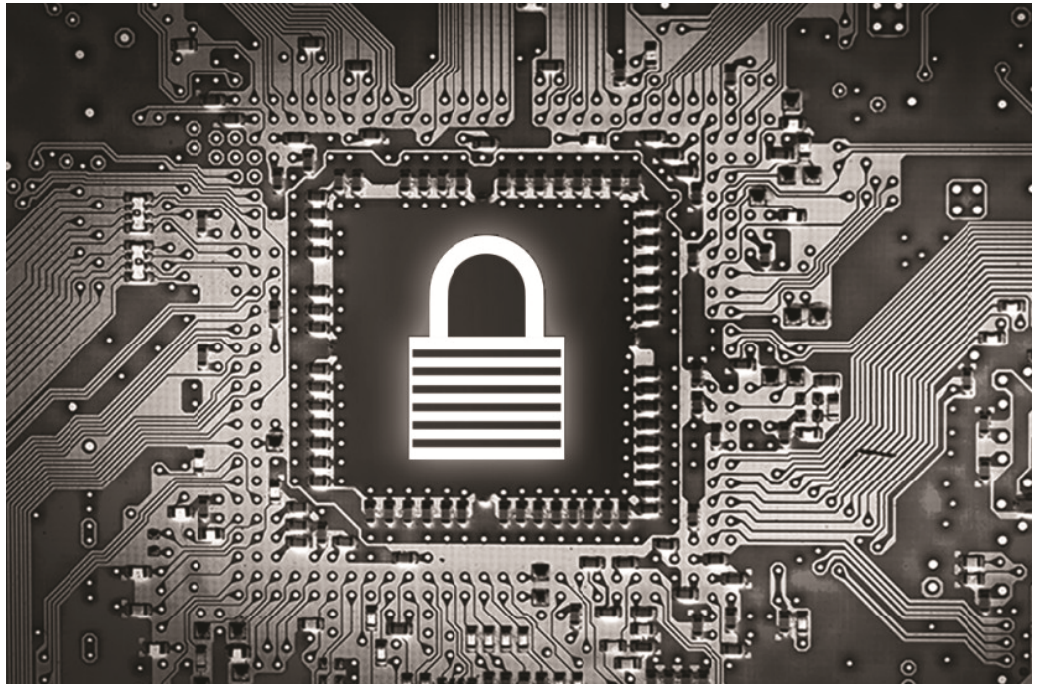
Open merged.etl using the Windows® Performance Analyzer.

Zoom to single frame using Present markers.

Move CPU Column next to Task Name then filter and expand CommandList.

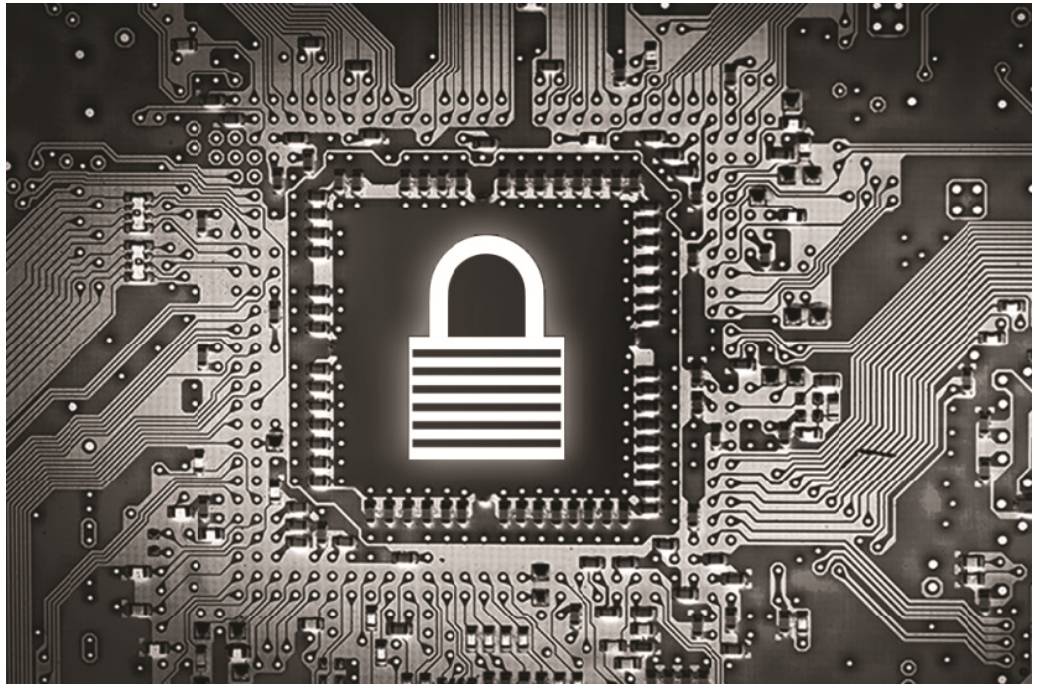
USE BEST PRACTICES WITH LOCKS

SUMMARY



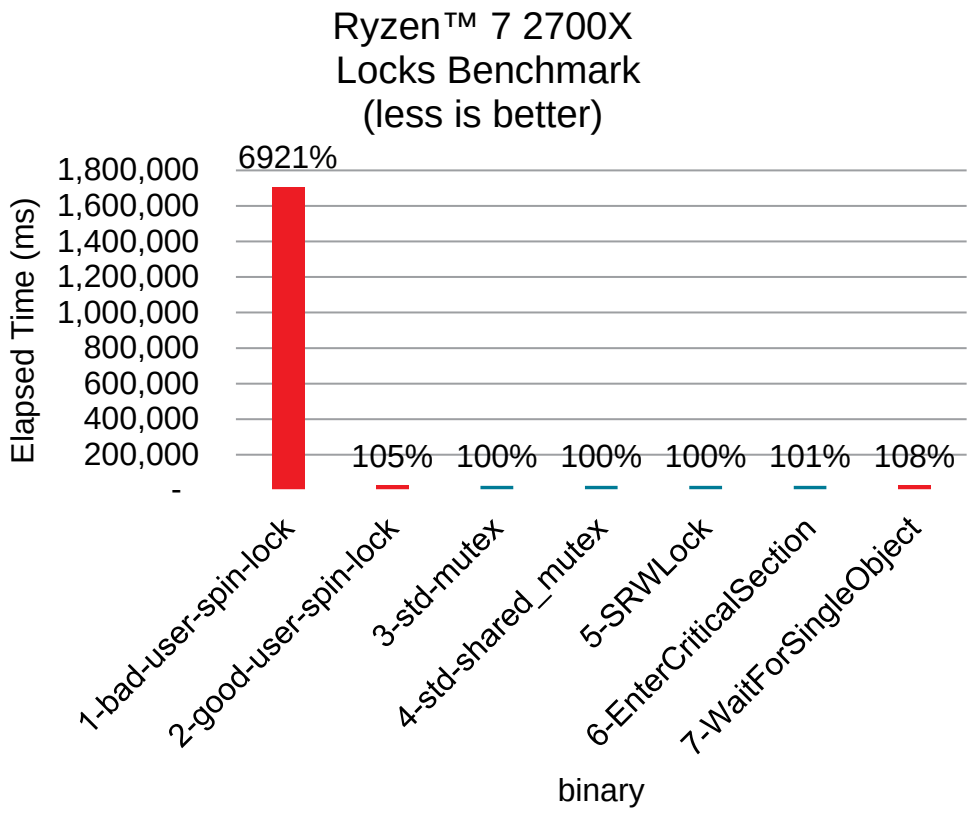
- Use modern OS synchronization APIs
  - Recommended:
    - std::mutex
    - std::shared_mutex
    - SRWLock
    - EnterCriticalSection
  - May allow more efficient scheduling and longer battery life

SUMMARY



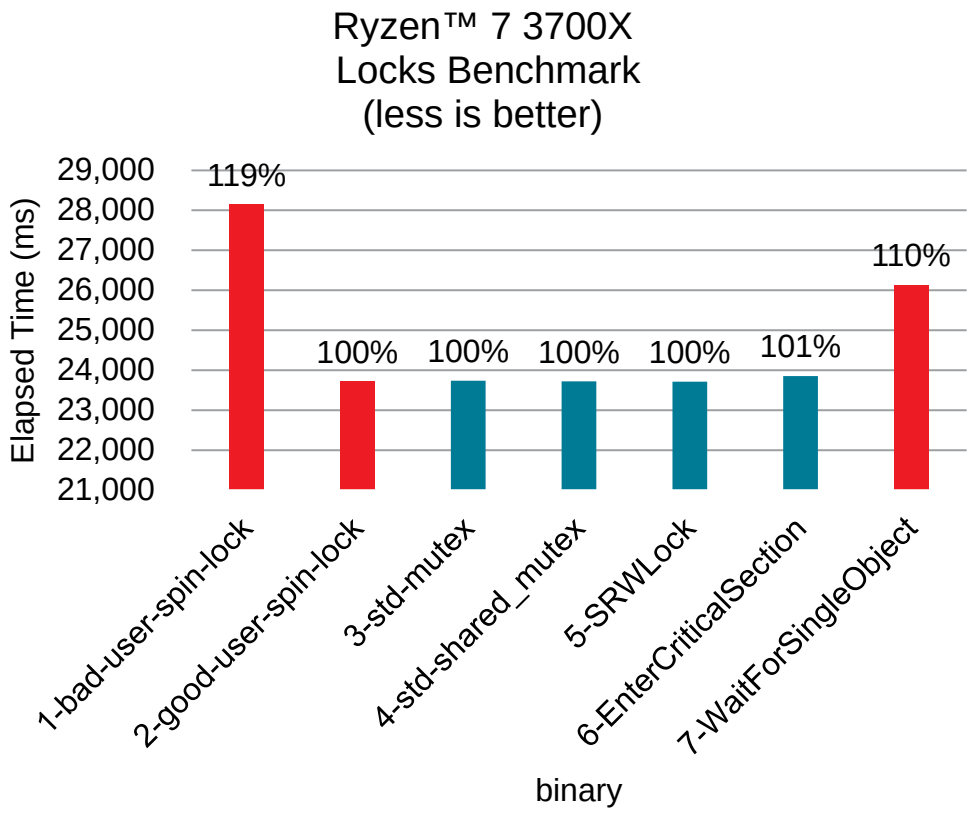
- Otherwise, for user spin locks:
  - Use the pause instruction
  - Alignas(64) lock variable
  - Test and test-and-set
  - Avoid lock prefix instructions
  - The OS may be unaware that threads are spinning; scheduling efficiency and battery life may be lost
  - Use spin locks only if held for a very short time

“ZEN 1” PERFORMANCE



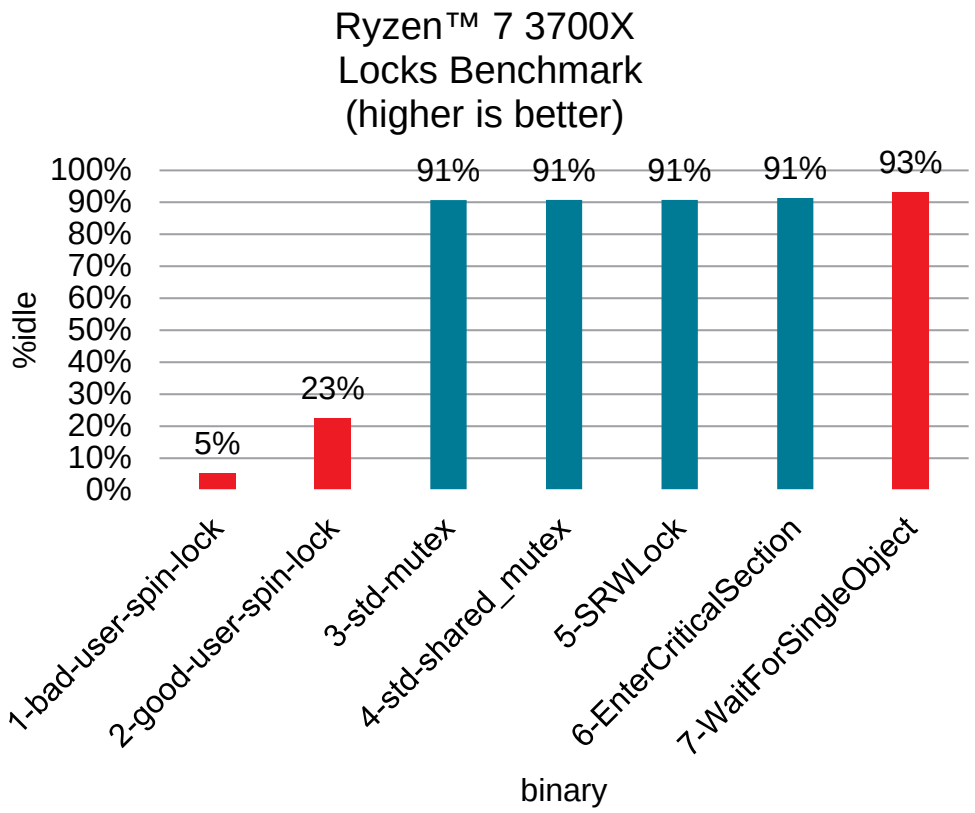
- Binaries compiled using best practices show improved performance.
- Performance of binary compiled with Microsoft Visual Studio 2019 v16.4.5.
- Testing done by AMD technology labs, February 13, 2020 on the following system. Test configuration: AMD Ryzen™ 7 2700X Processor, AMD Wraith Prism Cooler, 16GB (2 x 8GB DDR4-2667 at 20-19-19-43) memory, Radeon™ RX 5700 XT GPU with driver 20.1.4 (January 24, 2020), 512GB M.2 NVME SSD, AMD Ryzen™ Reference Motherboard, Windows® 10 x64 build 1909, 1920x1080 resolution. Actual results may vary.

“ZEN 2” PERFORMANCE



- “Zen 2” improved SMT fairness for ALU schedulers.
  - This helps mitigate bad user spin lock code .
- Binaries compiled using best practices show improved performance.
- Performance of binary compiled with Microsoft Visual Studio 2019 v16.4.5.
- Testing done by AMD technology labs, February 13, 2020 on the following system. Test configuration: AMD Ryzen™ 7 3700X Processor, AMD Wraith Prism Cooler, 16GB (2 x 8GB DDR4-3200 at 22-22-22-52) memory, Radeon™ RX 5700 XT GPU with driver 20.1.4 (January 24, 2020), 512GB M.2 NVME SSD, AMD Ryzen™ Reference Motherboard, Windows® 10 x64 build 1909, 1920x1080 resolution. Actual results may vary.

“ZEN 2” PERFORMANCE



- Binaries compiled using best practices shows improved idle.
- Performance of binary compiled with Microsoft Visual Studio 2019 v16.4.5.
- Testing done by AMD technology labs, February 13, 2020 on the following system. Test configuration: AMD Ryzen™ 7 3700X Processor, AMD Wraith Prism Cooler, 16GB (2 x 8GB DDR4-3200 at 22-22-22-52) memory, Radeon™ RX 5700 XT GPU with driver 20.1.4 (January 24, 2020), 512GB M.2 NVME SSD, AMD Ryzen™ Reference Motherboard, Windows® 10 x64 build 1909, 1920x1080 resolution. Actual results may vary.

PROFILING

- Use AMD uProf to find possible user spin locks
  - AMD uProf v3.2 "Assess Performance (Extended)" Event Based Sampling Profile
    - ALUTokenStall PTI
      - >= 3K Per Thousand Instructions is bad for top functions
  - Replace user spin locks with modern OS synchronization APIs when possible. Otherwise, use best practices.

- Use Microsoft Windows® Performance Analyzer to find call stacks using OS synchronization APIs
  - rem Recommend using public Microsoft symbol server
  - rem _NT_SYMBOL_PATH=srv*http://msdl.microsoft.com/download/symbols
  - rem “–start gpu –start video” wpr profiles are useful for game analysis for short durations
  - wpr.exe –setprofint 1221 –start power –filemode
  - test.exe
  - wpr.exe –stop log.etl

EXAMPLES SHARED CODE

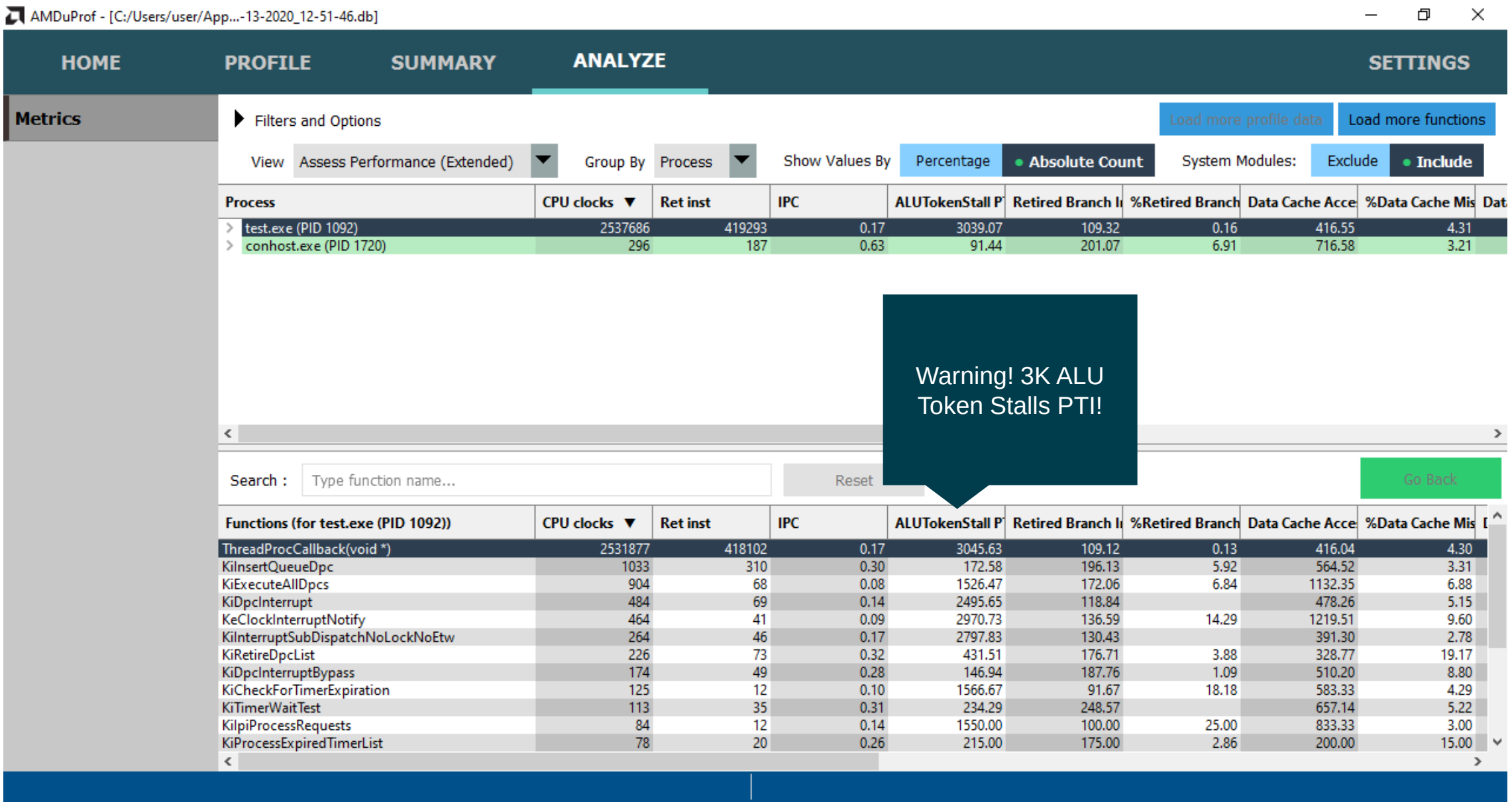
```
#include "intrin.h"
#include "stdio.h"
#include "windows.h"
#include <chrono>
#include <numeric>
#include <thread>
#include <mutex>
#include <shared_mutex>

#define LEN 512
alignas(64) float b[LEN][4][4];
alignas(64) float c[LEN][4][4];

int main(int argc, char* argv[]) {
    using namespace std::chrono;
    float b0 = (argc > 1) ? strtof(argv[1], NULL) : 1.0f;
    float c0 = (argc > 2) ? strtof(argv[2], NULL) : 2.0f;
    std::fill((float*)b, (float*)(b + LEN), b0);
    std::fill((float*)c, (float*)(c + LEN), c0);
    int num_threads = std::thread::hardware_concurrency();
    HANDLE* threads = new HANDLE[num_threads];
    high_resolution_clock::time_point t0 = \
        high_resolution_clock::now();

    for (size_t i = 0; i < num_threads; ++i) {
        threads[i] = CreateThread(NULL, \
            0, ThreadProcCallback, NULL, 0, NULL);
    }
    WaitForMultipleObjects(num_threads, \
        threads, TRUE, INFINITE);

    high_resolution_clock::time_point t1 = \
        high_resolution_clock::now();
    duration<double> time_span = \
        duration_cast<duration<double>>(t1 - t0);
    printf("time (milliseconds): %lf\n", \
        1000.0 * time_span.count());
    delete[] threads;
    return EXIT_SUCCESS;
}
```

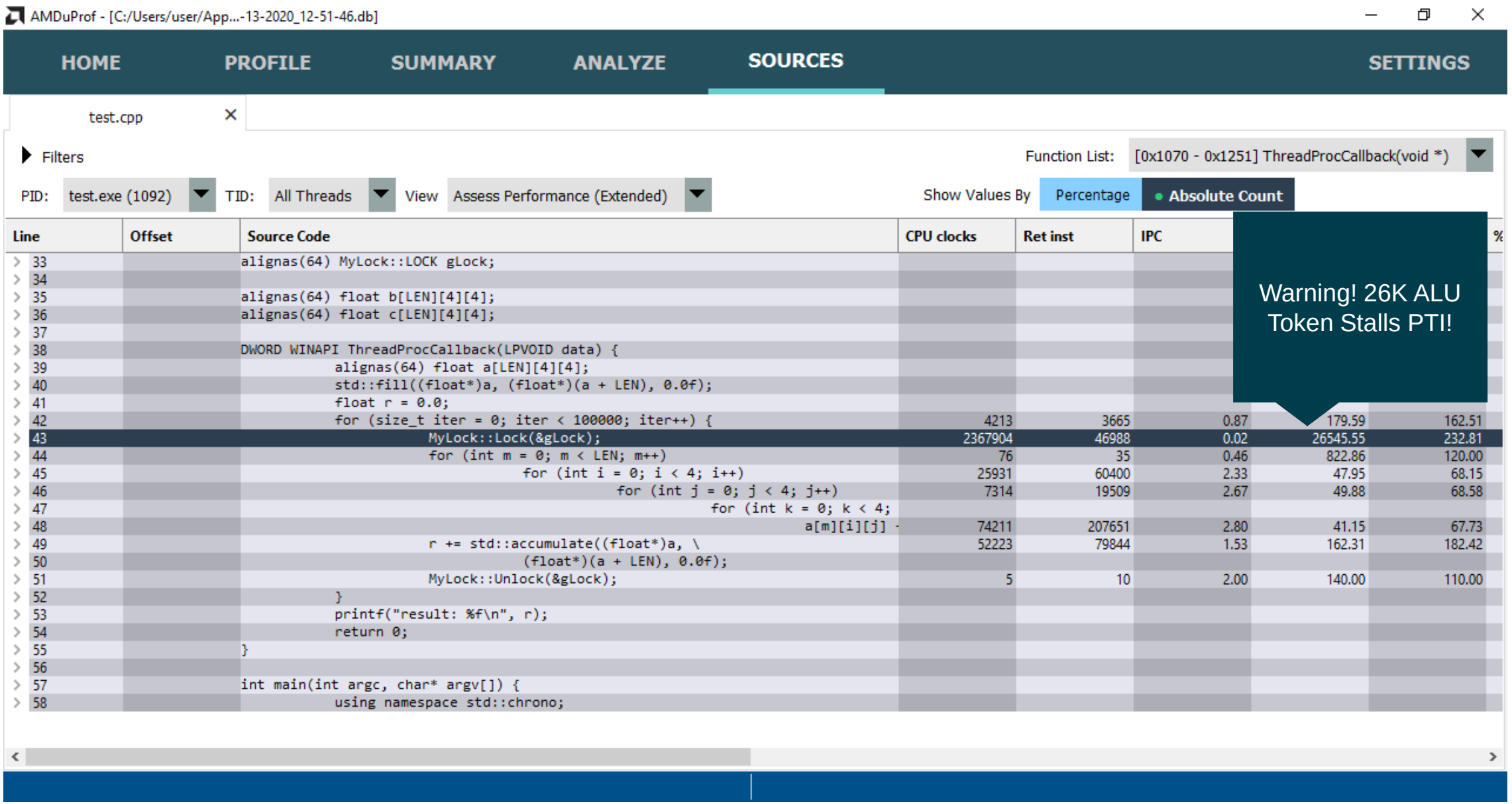
EXAMPLE 1 BAD USER SPIN LOCK

```
namespace MyLock {
    typedef unsigned LOCK, * PLOCK;
    enum { LOCK_IS_FREE = 0, LOCK_IS_TAKEN = 1 };
    void Lock(PLOCK pl) {
        while (LOCK_IS_TAKEN == \
            _InterlockedCompareExchange( \
            pl, LOCK_IS_TAKEN, LOCK_IS_FREE)) {
        }
    }
    void Unlock(PLOCK pl) {
        _InterlockedExchange(pl, LOCK_IS_FREE);
    }
}
```

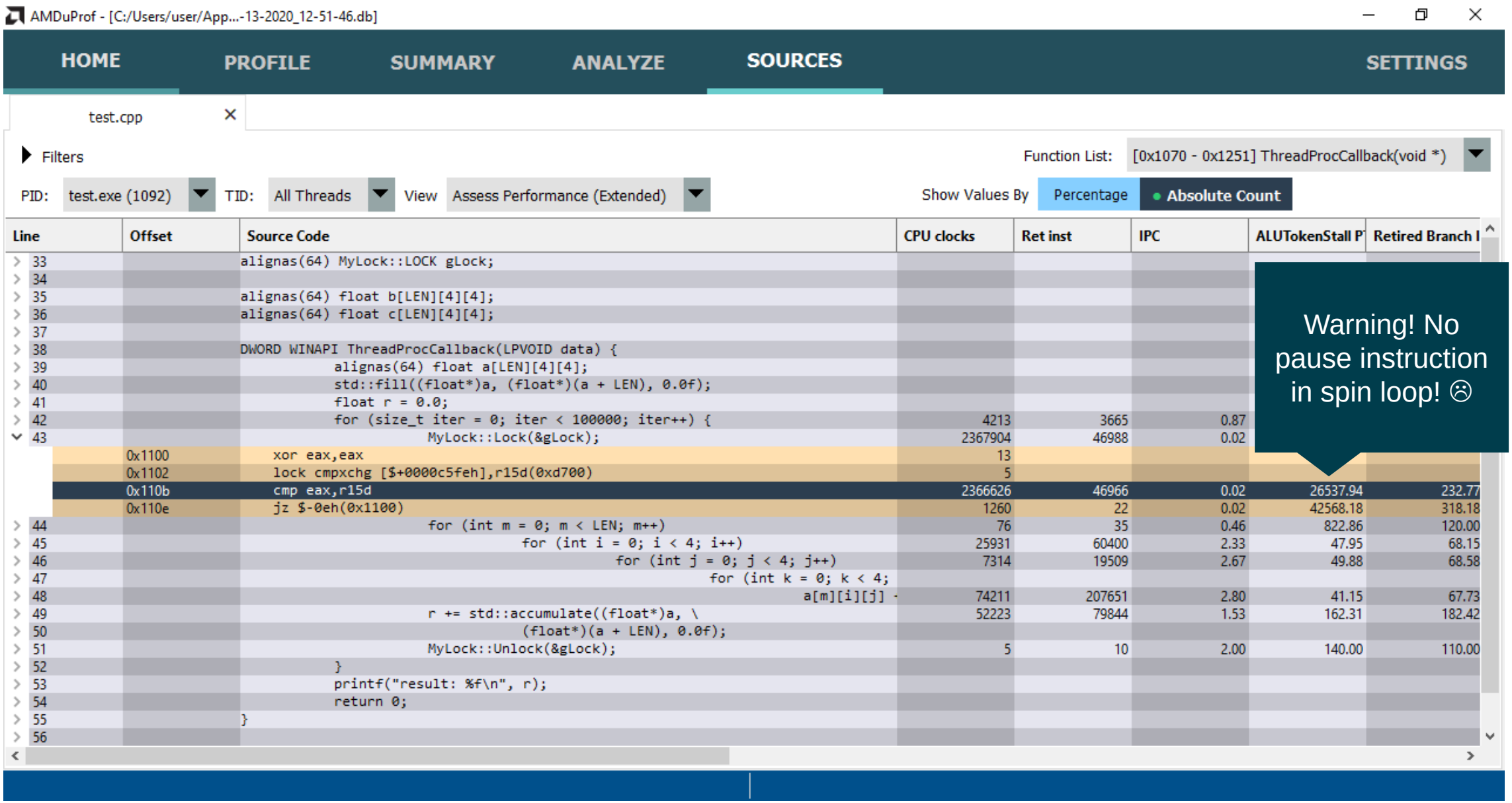
Warning! Not best practices for spin lock.

```
alignas(64) MyLock::LOCK gLock;
DWORD WINAPI ThreadProcCallback(LPVOID data) {
    alignas(64) float a[LEN][4][4];
    std::fill((float*)a, (float*)(a + LEN), 0.0f);
    float r = 0.0;
    for (size_t iter = 0; iter < 100000; iter++) {
        MyLock::Lock(&gLock);
        for (int m = 0; m < LEN; m++)
            for (int i = 0; i < 4; i++)
                for (int j = 0; j < 4; j++)
                    for (int k = 0; k < 4; k++)
                        a[m][i][j] += b[m][i][k] * c[m][k][j];
        r += std::accumulate((float*)a, \
            (float*)(a + LEN), 0.0f);
        MyLock::Unlock(&gLock);
    }
    printf("result: %f\n", r);
    return 0;
}
```


AMDuProf - [C:/Users/user/App...-13-2020_12-51-46.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Metrics
Filters and Options
Load more profile data
Load more functions
View
Assess Performance (Extended)
Group By
Process
Show Values By
Percentage
Absolute Count
System Modules:
Exclude
Include
Process
CPU clocks
Ret inst
IPC
ALUTokenStall P
Retired Branch I
%Retired Branch
Data Cache Acce
%Data Cache Mis
test.exe (PID 1092) 2537686 419293 0.17 3039.07 109.32 0.16 416.55 4.31
conhost.exe (PID 1720) 296 187 0.63 91.44 201.07 6.91 716.58 3.21
Warning! 3K ALU Token Stalls PTI!
Search :
Type function name...
Reset
Go Back
Functions (for test.exe (PID 1092))
CPU clocks
Ret inst
IPC
ALUTokenStall P
Retired Branch I
%Retired Branch
Data Cache Acce
%Data Cache Mis
ThreadProcCallback(void *) 2531877 418102 0.17 3045.63 109.12 0.13 416.04 4.30
KiInsertQueueDpc 1033 310 0.30 172.58 196.13 5.92 564.52 3.31
KiExecuteAllDpcs 904 68 0.08 1526.47 172.06 6.84 1132.35 6.88
KiDpcInterrupt 484 69 0.14 2495.65 118.84 478.26 5.15
KeClockInterruptNotify 464 41 0.09 2970.73 136.59 14.29 1219.51 9.60
KiInterruptSubDispatchNoLockNoEtw 264 46 0.17 2797.83 130.43 391.30 2.78
KiRetireDpcList 226 73 0.32 431.51 176.71 3.88 328.77 19.17
KiDpcInterruptBypass 174 49 0.28 146.94 187.76 1.09 510.20 8.80
KiCheckForTimerExpiration 125 12 0.10 1566.67 91.67 18.18 583.33 4.29
KiTimerWaitTest 113 35 0.31 234.29 248.57 657.14 5.22
KiIpiProcessRequests 84 12 0.14 1550.00 100.00 25.00 833.33 3.00
KiProcessExpiredTimerList 78 20 0.26 215.00 175.00 2.86 200.00 15.00



Line	Offset	Source Code	CPU clocks	Ret inst	IPC		
33		alignas(64) MyLock::LOCK gLock;					
34							
35		alignas(64) float b[LEN][4][4];					
36		alignas(64) float c[LEN][4][4];					
37							
38		DWORD WINAPI ThreadProcCallback(LPVOID data) {					
39		alignas(64) float a[LEN][4][4];					
40		std::fill((float*)a, (float*)(a + LEN), 0.0f);					
41		float r = 0.0;					
42		for (size_t iter = 0; iter < 100000; iter++) {	4213	3665	0.87	179.59	162.51
43		MyLock::Lock(&gLock);	2367904	46988	0.02	26545.55	232.81
44		for (int m = 0; m < LEN; m++)	76	35	0.46	822.86	120.00
45		for (int i = 0; i < 4; i++)	25931	60400	2.33	47.95	68.15
46		for (int j = 0; j < 4; j++)	7314	19509	2.67	49.88	68.58
47		for (int k = 0; k < 4;					
48		a[m][i][j] -	74211	207651	2.80	41.15	67.73
49		r += std::accumulate((float*)a, \	52223	79844	1.53	162.31	182.42
50		(float*)(a + LEN), 0.0f);					
51		MyLock::Unlock(&gLock);	5	10	2.00	140.00	110.00
52		}					
53		printf("result: %f\n", r);					
54		return 0;					
55		}					
56							
57		int main(int argc, char* argv[]) {					
58		using namespace std::chrono;					

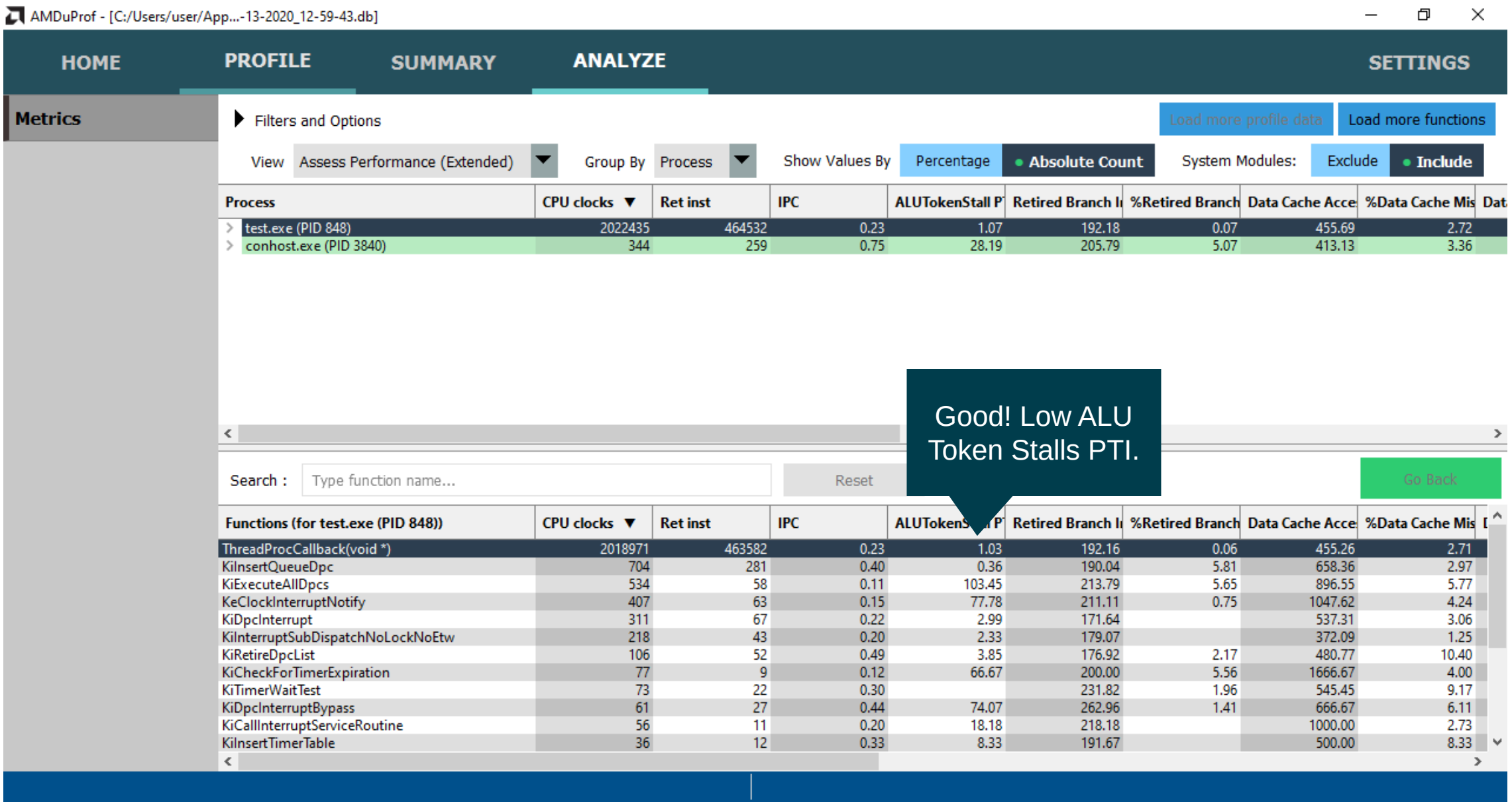

AMDuProf - [C:/Users/user/App...-13-2020_12-51-46.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
test.cpp
Filters
Function List: [0x1070 - 0x1251] ThreadProcCallback(void *)
PID: test.exe (1092)
TID: All Threads
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line | Offset | Source Code | CPU clocks | Ret inst | IPC | ALUTokenStall P | Retired Branch I
33 alignas(64) MyLock::LOCK gLock;
34
35 alignas(64) float b[LEN][4][4];
36 alignas(64) float c[LEN][4][4];
37
38 DWORD WINAPI ThreadProcCallback(LPVOID data) {
39 alignas(64) float a[LEN][4][4];
40 std::fill((float*)a, (float*)(a + LEN), 0.0f);
41 float r = 0.0;
42 for (size_t iter = 0; iter < 100000; iter++) { 4213 3665 0.87
43 MyLock::Lock(&gLock); 2367904 46988 0.02
0x1100 xor eax,eax 13
0x1102 lock cmpxchg [$+0000c5feh],r15d(0xd700) 5
0x110b cmp eax,r15d 2366626 46966 0.02 26537.94 232.77
0x110e jz $-0eh(0x1100) 1260 22 0.02 42568.18 318.18
44 for (int m = 0; m < LEN; m++) 76 35 0.46 822.86 120.00
45 for (int i = 0; i < 4; i++) 25931 60400 2.33 47.95 68.15
46 for (int j = 0; j < 4; j++) 7314 19509 2.67 49.88 68.58
47 for (int k = 0; k < 4;
48 a[m][i][j] - 74211 207651 2.80 41.15 67.73
49 r += std::accumulate((float*)a, \ 52223 79844 1.53 162.31 182.42
50 (float*)(a + LEN), 0.0f);
51 MyLock::Unlock(&gLock); 5 10 2.00 140.00 110.00
52 }
53 printf("result: %f\n", r);
54 return 0;
55 }
56
Warning! No pause instruction in spin loop! ☹

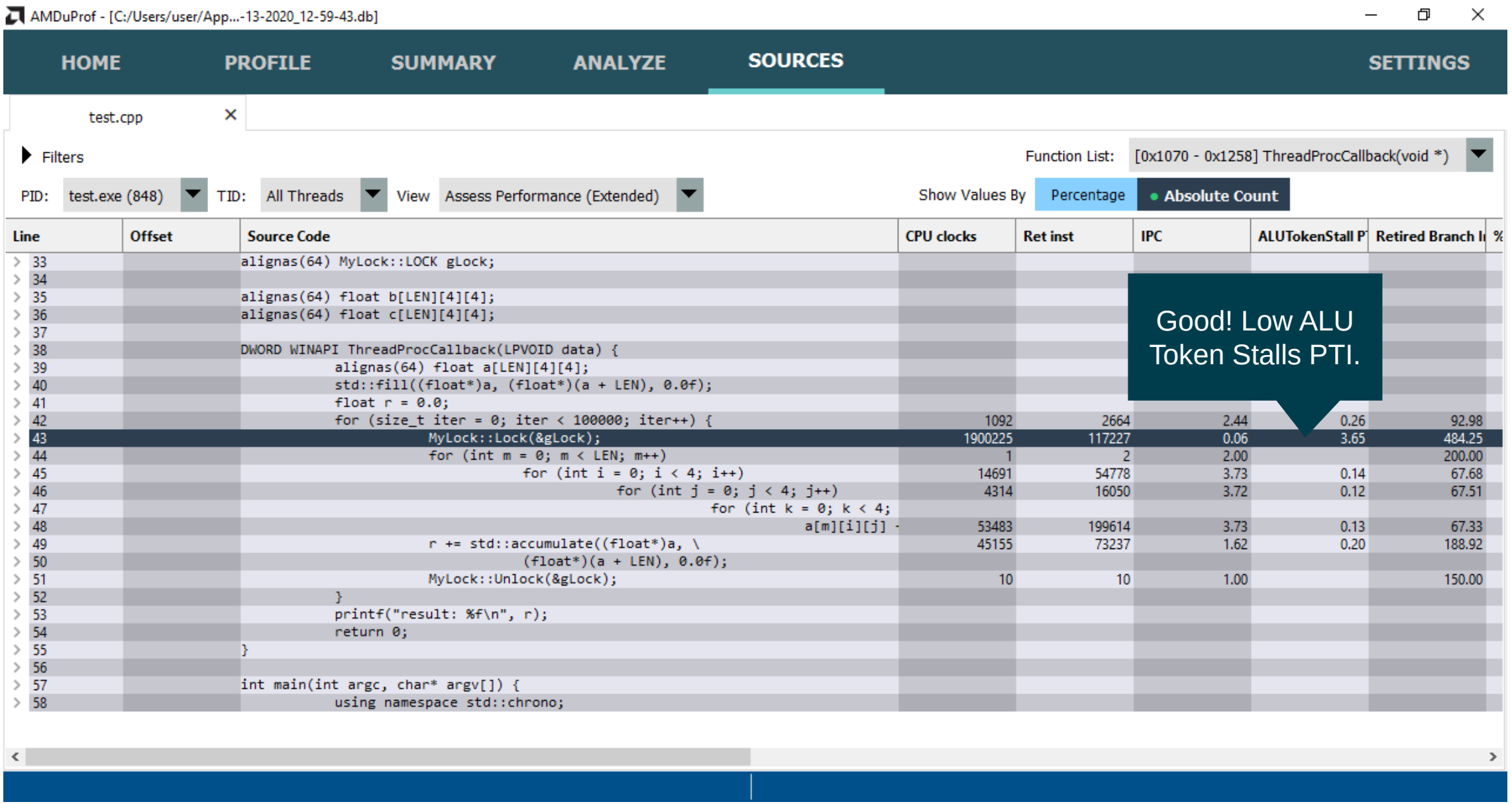
EXAMPLE 2 IMPROVED USER SPIN LOCK

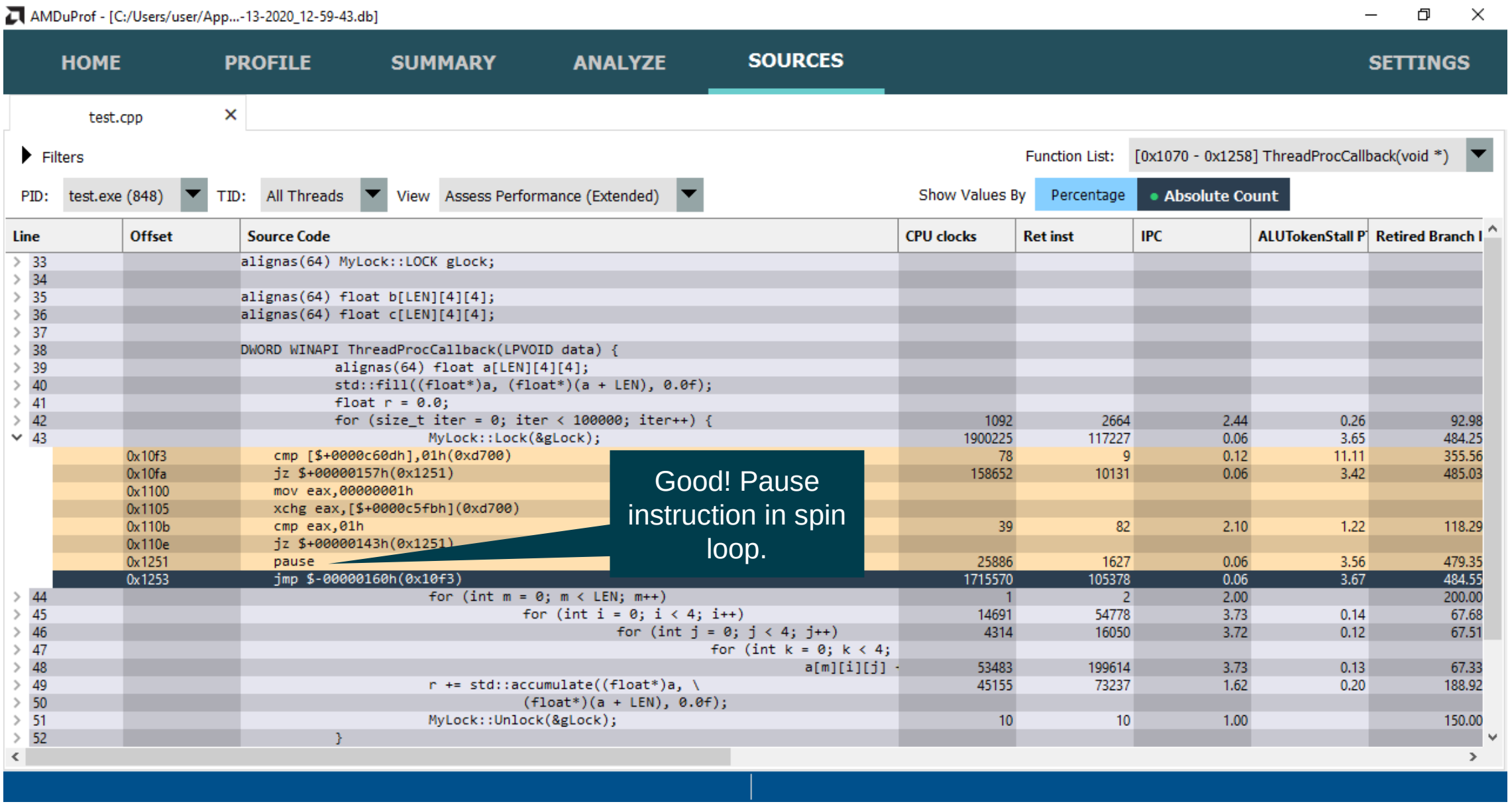
```
namespace MyLock {
    typedef unsigned LOCK, * PLOCK;
    enum { LOCK_IS_FREE = 0, LOCK_IS_TAKEN = 1 };
    void Lock(PLOCK pl) {
        while ((LOCK_IS_TAKEN == *pl) || \
            (LOCK_IS_TAKEN == \
            _InterlockedExchange(pl, LOCK_IS_TAKEN))) {
            _mm_pause();
        }
    }
    void Unlock(PLOCK pl) {
        _InterlockedExchange(pl, LOCK_IS_FREE);
    }
}
```

Good! Applied best practices for spin lock.

```
alignas(64) MyLock::LOCK gLock;
DWORD WINAPI ThreadProcCallback(LPVOID data) {
    alignas(64) float a[LEN][4][4];
    std::fill((float*)a, (float*)(a + LEN), 0.0f);
    float r = 0.0;
    for (size_t iter = 0; iter < 100000; iter++) {
        MyLock::Lock(&gLock);
        for (int m = 0; m < LEN; m++)
            for (int i = 0; i < 4; i++)
                for (int j = 0; j < 4; j++)
                    for (int k = 0; k < 4; k++)
                        a[m][i][j] += b[m][i][k] * c[m][k][j];
        r += std::accumulate((float*)a, \
            (float*)(a + LEN), 0.0f);
        MyLock::Unlock(&gLock);
    }
    printf("result: %f\n", r);
    return 0;
}
```


AMDuProf - [C:/Users/user/App...-13-2020_12-59-43.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Metrics
Filters and Options
Load more profile data
Load more functions
View
Assess Performance (Extended)
Group By
Process
Show Values By
Percentage
Absolute Count
System Modules:
Exclude
Include
Process
CPU clocks
Ret inst
IPC
ALUTokenStall P
Retired Branch I
%Retired Branch
Data Cache Acce
%Data Cache Mis
test.exe (PID 848) 2022435 464532 0.23 1.07 192.18 0.07 455.69 2.72
conhost.exe (PID 3840) 344 259 0.75 28.19 205.79 5.07 413.13 3.36
Good! Low ALU Token Stalls PTI.
Search :
Type function name...
Reset
Go Back
Functions (for test.exe (PID 848))
CPU clocks
Ret inst
IPC
Retired Branch I
%Retired Branch
Data Cache Acce
%Data Cache Mis
ThreadProcCallback(void *) 2018971 463582 0.23 1.03 192.16 0.06 455.26 2.71
KiInsertQueueDpc 704 281 0.40 0.36 190.04 5.81 658.36 2.97
KiExecuteAllDpcs 534 58 0.11 103.45 213.79 5.65 896.55 5.77
KeClockInterruptNotify 407 63 0.15 77.78 211.11 0.75 1047.62 4.24
KiDpcInterrupt 311 67 0.22 2.99 171.64 537.31 3.06
KiInterruptSubDispatchNoLockNoEtw 218 43 0.20 2.33 179.07 372.09 1.25
KiRetireDpcList 106 52 0.49 3.85 176.92 2.17 480.77 10.40
KiCheckForTimerExpiration 77 9 0.12 66.67 200.00 5.56 1666.67 4.00
KiTimerWaitTest 73 22 0.30 231.82 1.96 545.45 9.17
KiDpcInterruptBypass 61 27 0.44 74.07 262.96 1.41 666.67 6.11
KiCallInterruptServiceRoutine 56 11 0.20 18.18 218.18 1000.00 2.73
KiInsertTimerTable 36 12 0.33 8.33 191.67 500.00 8.33

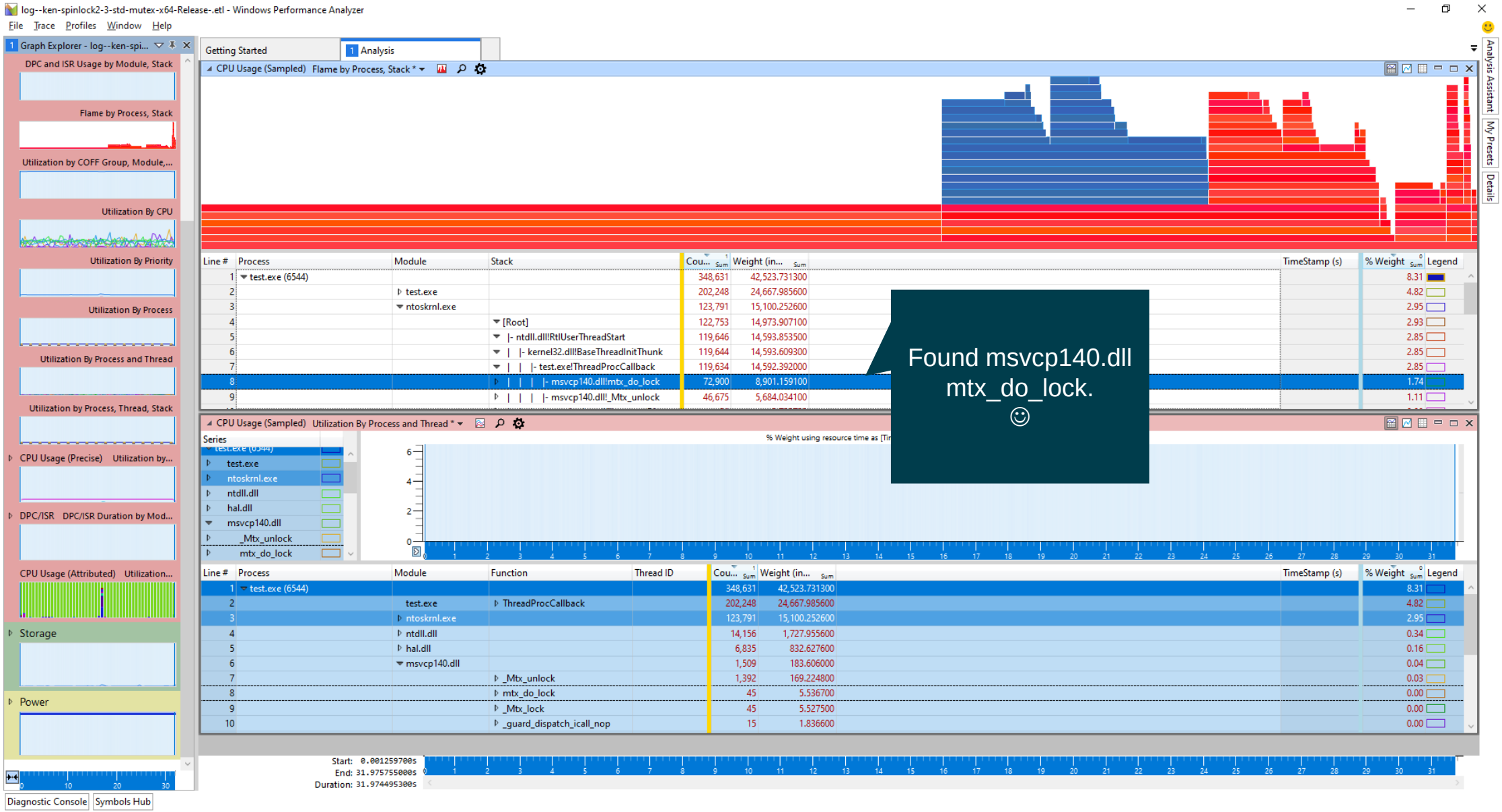

AMDuProf - [C:/Users/user/App...-13-2020_12-59-43.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
test.cpp
Filters
Function List: [0x1070 - 0x1258] ThreadProcCallback(void *)
PID: test.exe (848)
TID: All Threads
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line Offset Source Code CPU clocks Ret inst IPC ALUTokenStall P Retired Branch I %
33 alignas(64) MyLock::LOCK gLock;
34
35 alignas(64) float b[LEN][4][4];
36 alignas(64) float c[LEN][4][4];
37
38 DWORD WINAPI ThreadProcCallback(LPVOID data) {
39 alignas(64) float a[LEN][4][4];
40 std::fill((float*)a, (float*)(a + LEN), 0.0f);
41 float r = 0.0;
42 for (size_t iter = 0; iter < 100000; iter++) { 1092 2664 2.44 0.26 92.98
43 MyLock::Lock(&gLock); 1900225 117227 0.06 3.65 484.25
44 for (int m = 0; m < LEN; m++) 1 2 2.00 200.00
45 for (int i = 0; i < 4; i++) 14691 54778 3.73 0.14 67.68
46 for (int j = 0; j < 4; j++) 4314 16050 3.72 0.12 67.51
47 for (int k = 0; k < 4;
48 a[m][i][j] - 53483 199614 3.73 0.13 67.33
49 r += std::accumulate((float*)a, \ 45155 73237 1.62 0.20 188.92
50 (float*)(a + LEN), 0.0f);
51 MyLock::Unlock(&gLock); 10 10 1.00 150.00
52 }
53 printf("result: %f\n", r);
54 return 0;
55 }
56
57 int main(int argc, char* argv[]) {
58 using namespace std::chrono;
Good! Low ALU Token Stalls PTI.


AMDuProf - [C:/Users/user/App...-13-2020_12-59-43.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
test.cpp
Filters
Function List: [0x1070 - 0x1258] ThreadProcCallback(void *)
PID: test.exe (848)
TID: All Threads
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line Offset Source Code CPU clocks Ret inst IPC ALUTokenStall P Retired Branch I
33 alignas(64) MyLock::LOCK gLock;
34
35 alignas(64) float b[LEN][4][4];
36 alignas(64) float c[LEN][4][4];
37
38 DWORD WINAPI ThreadProcCallback(LPVOID data) {
39 alignas(64) float a[LEN][4][4];
40 std::fill((float*)a, (float*)(a + LEN), 0.0f);
41 float r = 0.0;
42 for (size_t iter = 0; iter < 100000; iter++) { 1092 2664 2.44 0.26 92.98
43 MyLock::Lock(&gLock); 1900225 117227 0.06 3.65 484.25
0x10f3 cmp [$+0000c60dh],01h(0xd700) 78 9 0.12 11.11 355.56
0x10fa jz $+00000157h(0x1251) 158652 10131 0.06 3.42 485.03
0x1100 mov eax,00000001h
0x1105 xchg eax,[$+0000c5fbh](0xd700)
0x110b cmp eax,01h 39 82 2.10 1.22 118.29
0x110e jz $+00000143h(0x1251)
0x1251 pause 25886 1627 0.06 3.56 479.35
0x1253 jmp $-00000160h(0x10f3) 1715570 105378 0.06 3.67 484.55
44 for (int m = 0; m < LEN; m++) 1 2 2.00 200.00
45 for (int i = 0; i < 4; i++) 14691 54778 3.73 0.14 67.68
46 for (int j = 0; j < 4; j++) 4314 16050 3.72 0.12 67.51
47 for (int k = 0; k < 4;
48 a[m][i][j] - 53483 199614 3.73 0.13 67.33
49 r += std::accumulate((float*)a, \ 45155 73237 1.62 0.20 188.92
50 (float*)(a + LEN), 0.0f);
51 MyLock::Unlock(&gLock); 10 10 1.00 150.00
52 }
Good! Pause instruction in spin loop.

EXAMPLE 3 STD::MUTEX

```
// MyLock not required. Let the OS do the work!
```

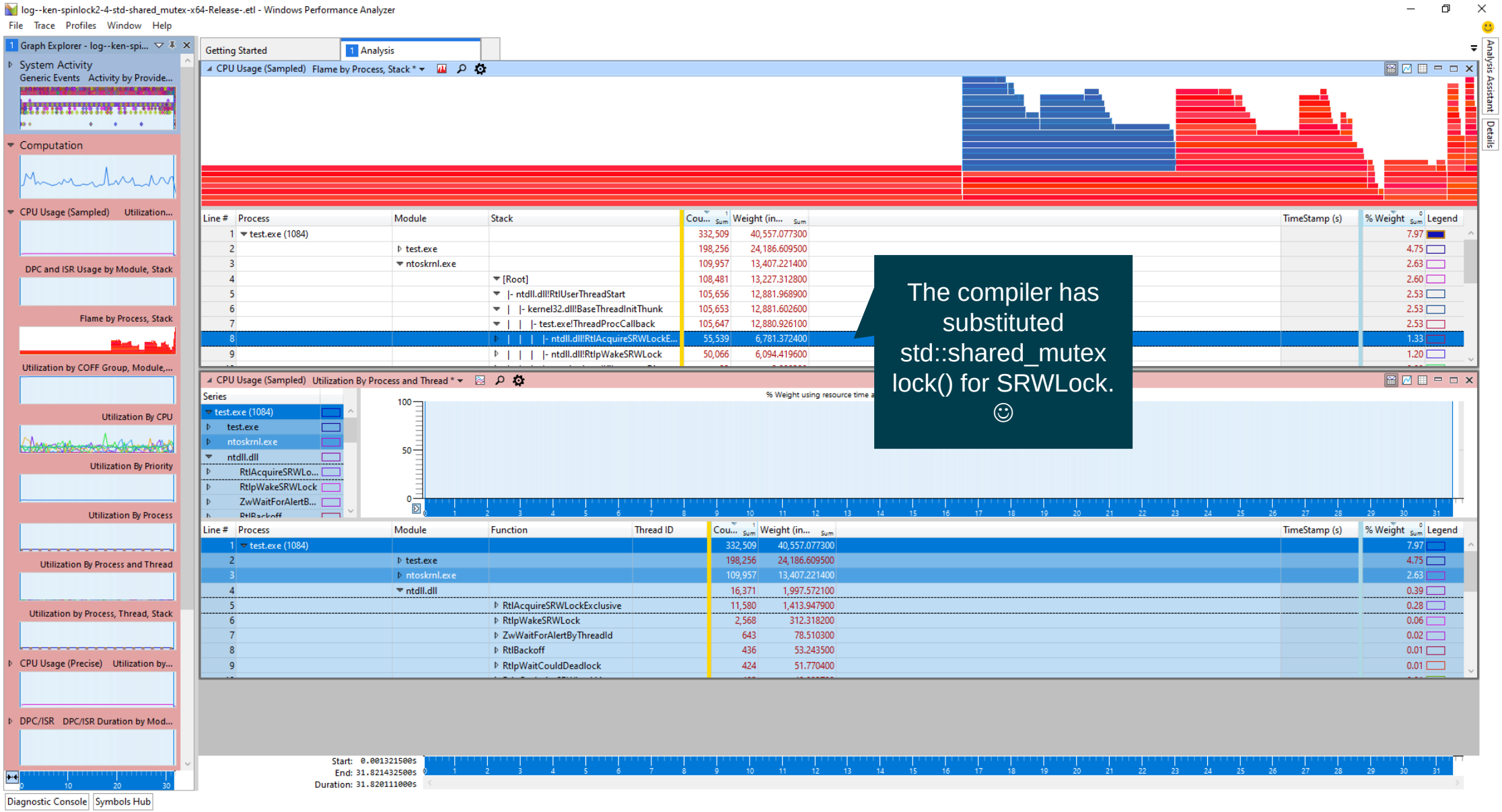
```
std::mutex mutex;
DWORD WINAPI ThreadProcCallback(LPVOID data) {
    alignas(64) float a[LEN][4][4];
    std::fill((float*)a, (float*)(a + LEN), 0.0f);
    float r = 0.0;
    for (size_t iter = 0; iter < 100000; iter++) {
        mutex.lock();
        for (int m = 0; m < LEN; m++)
            for (int i = 0; i < 4; i++)
                for (int j = 0; j < 4; j++)
                    for (int k = 0; k < 4; k++)
                        a[m][i][j] += b[m][i][k] * c[m][k][j];
        r += std::accumulate((float*)a, \
            (float*)(a + LEN), 0.0f);
        mutex.unlock();
    }
    printf("result: %f\n", r);
    return 0;
}
```


log--ken-spinlock2-3-std-mutex-x64-Release-.etl - Windows Performance Analyzer
File Trace Profiles Window Help
Graph Explorer - log--ken-spi...
DPC and ISR Usage by Module, Stack
Flame by Process, Stack
Utilization by COFF Group, Module,...
Utilization By CPU
Utilization By Priority
Utilization By Process
Utilization By Process and Thread
Utilization by Process, Thread, Stack
CPU Usage (Precise) Utilization by...
DPC/ISR DPC/ISR Duration by Mod...
CPU Usage (Attributed) Utilization...
Storage
Power
Diagnostic Console
Symbols Hub
Getting Started
Analysis
CPU Usage (Sampled) Flame by Process, Stack *
Line # Process Module Stack Cou... Weight (in... TimeStamp (s) % Weight Legend
1 test.exe (6544) 348,631 42,523.731300 8.31
2 test.exe 202,248 24,667.985600 4.82
3 ntoskrnl.exe 123,791 15,100.252600 2.95
4 [Root] 122,753 14,973.907100 2.93
5 |- ntdll.dll!RtlUserThreadStart 119,646 14,593.853500 2.85
6 |- kernel32.dll!BaseThreadInitThunk 119,644 14,593.609300 2.85
7 |- test.exe!ThreadProcCallback 119,634 14,592.392000 2.85
8 |- msvcp140.dll!mtx_do_lock 72,900 8,901.159100 1.74
9 |- msvcp140.dll!_Mtx_unlock 46,675 5,684.034100 1.11
Found msvcp140.dll mtx_do_lock.
CPU Usage (Sampled) Utilization By Process and Thread *
Series
test.exe
ntoskrnl.exe
ntdll.dll
hal.dll
msvcp140.dll
_Mtx_unlock
mtx_do_lock
% Weight using resource time as [Ti
Line # Process Module Function Thread ID Cou... Weight (in... TimeStamp (s) % Weight Legend
1 test.exe (6544) 348,631 42,523.731300 8.31
2 test.exe ThreadProcCallback 202,248 24,667.985600 4.82
3 ntoskrnl.exe 123,791 15,100.252600 2.95
4 ntdll.dll 14,156 1,727.955600 0.34
5 hal.dll 6,835 832.627600 0.16
6 msvcp140.dll 1,509 183.606000 0.04
7 _Mtx_unlock 1,392 169.224800 0.03
8 mtx_do_lock 45 5.536700 0.00
9 _Mtx_lock 45 5.527500 0.00
10 _guard_dispatch_icall_nop 15 1.836600 0.00
Start: 0.001259700s
End: 31.975755000s
Duration: 31.974495300s
Analysis Assistant
My Presets
Details

EXAMPLE 4 STD::SHARED_MUTEX

```
// MyLock not required. Let the OS do the work!
```

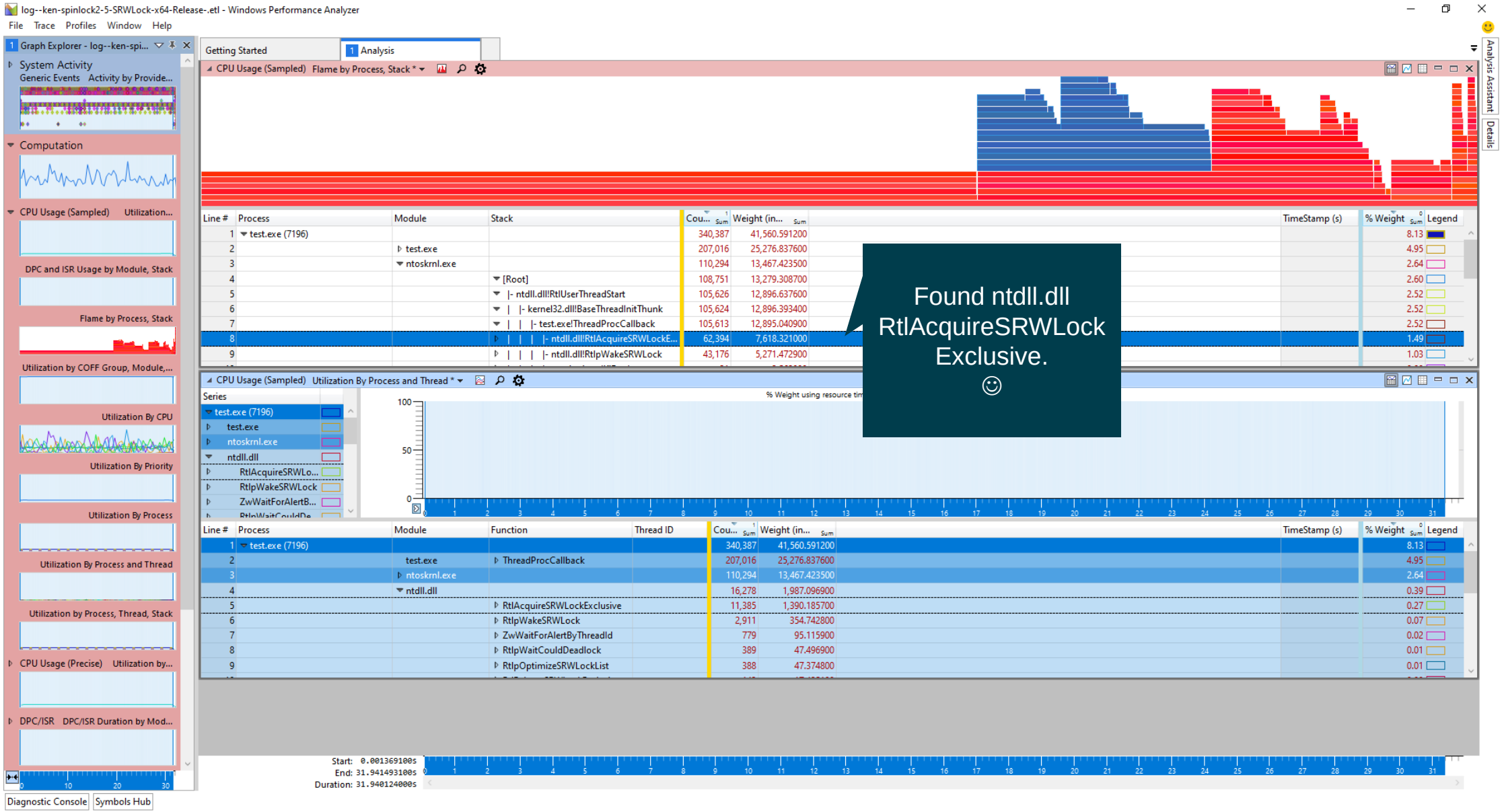
```
std::shared_mutex mutex;
DWORD WINAPI ThreadProcCallback(LPVOID data) {
    alignas(64) float a[LEN][4][4];
    std::fill((float*)a, (float*)(a + LEN), 0.0f);
    float r = 0.0;
    for (size_t iter = 0; iter < 100000; iter++) {
        mutex.lock();
        for (int m = 0; m < LEN; m++)
            for (int i = 0; i < 4; i++)
                for (int j = 0; j < 4; j++)
                    for (int k = 0; k < 4; k++)
                        a[m][i][j] += b[m][i][k] * c[m][k][j];
        r += std::accumulate((float*)a, \
            (float*)(a + LEN), 0.0f);
        mutex.unlock();
    }
    printf("result: %f\n", r);
    return 0;
}
```


log--ken-spinlock2-4-std-shared_mutex-x64-Release-.etl - Windows Performance Analyzer
File Trace Profiles Window Help
Graph Explorer - log--ken-spi...
System Activity
Generic Events Activity by Provide...
Computation
CPU Usage (Sampled) Utilization...
DPC and ISR Usage by Module, Stack
Flame by Process, Stack
Utilization by COFF Group, Module,...
Utilization By CPU
Utilization By Priority
Utilization By Process
Utilization By Process and Thread
Utilization by Process, Thread, Stack
CPU Usage (Precise) Utilization by...
DPC/ISR DPC/ISR Duration by Mod...
Diagnostic Console
Symbols Hub
Getting Started
Analysis
CPU Usage (Sampled) Flame by Process, Stack *
Line # Process Module Stack Cou... Weight (in... TimeStamp (s) % Weight Legend
1 test.exe (1084) 332,509 40,557.077300 7.97
2 test.exe 198,256 24,186.609500 4.75
3 ntoskrnl.exe 109,957 13,407.221400 2.63
4 [Root] 108,481 13,227.312800 2.60
5 |- ntdll.dll!RtlUserThreadStart 105,656 12,881.968900 2.53
6 |- kernel32.dll!BaseThreadInitThunk 105,653 12,881.602600 2.53
7 |- test.exe!ThreadProcCallback 105,647 12,880.926100 2.53
8 |- ntdll.dll!RtlAcquireSRWLockE... 55,539 6,781.372400 1.33
9 |- ntdll.dll!RtlpWakeSRWLock 50,066 6,094.419600 1.20
CPU Usage (Sampled) Utilization By Process and Thread *
Series
test.exe (1084)
test.exe
ntoskrnl.exe
ntdll.dll
RtlAcquireSRWLo...
RtlpWakeSRWLock
ZwWaitForAlertB...
% Weight using resource time
100
50
0
Line # Process Module Function Thread ID Cou... Weight (in... TimeStamp (s) % Weight Legend
1 test.exe (1084) 332,509 40,557.077300 7.97
2 test.exe 198,256 24,186.609500 4.75
3 ntoskrnl.exe 109,957 13,407.221400 2.63
4 ntdll.dll 16,371 1,997.572100 0.39
5 RtlAcquireSRWLockExclusive 11,580 1,413.947900 0.28
6 RtlpWakeSRWLock 2,568 312.318200 0.06
7 ZwWaitForAlertByThreadId 643 78.510300 0.02
8 RtlBackoff 436 53.243500 0.01
9 RtlpWaitCouldDeadlock 424 51.770400 0.01
Start: 0.001321500s
End: 31.821432500s
Duration: 31.820111000s
Analysis Assistant
Details
The compiler has substituted std::shared_mutex lock() for SRWLock. ☺

EXAMPLE 5 SRWLOCK

```
// MyLock not required. Let the OS do the work!
```

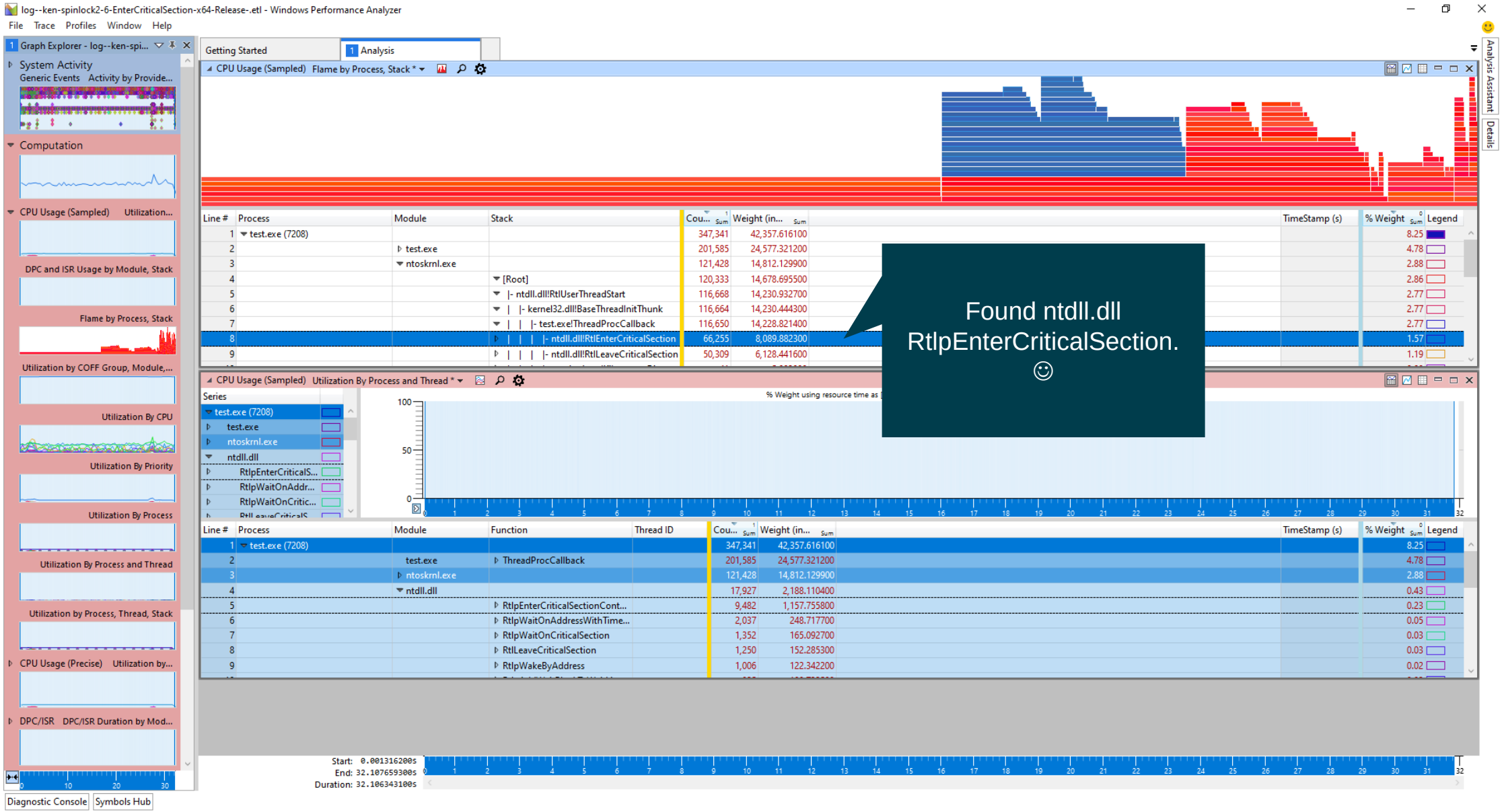
```
SRWLOCK lock;
DWORD WINAPI ThreadProcCallback(LPVOID data) {
    alignas(64) float a[LEN][4][4];
    std::fill((float*)a, (float*)(a + LEN), 0.0f);
    float r = 0.0;
    for (size_t iter = 0; iter < 100000; iter++) {
        AcquireSRWLockExclusive(&lock);
        for (int m = 0; m < LEN; m++)
            for (int i = 0; i < 4; i++)
                for (int j = 0; j < 4; j++)
                    for (int k = 0; k < 4; k++)
                        a[m][i][j] += b[m][i][k] * c[m][k][j];
        r += std::accumulate((float*)a, \
            (float*)(a + LEN), 0.0f);
        ReleaseSRWLockExclusive(&lock);
    }
    printf("result: %f\n", r);
    return 0;
}
int main(int argc, char* argv[]) {
    InitializeSRWLock(&lock);
    // ...
```


log--ken-spinlock2-5-SRWLock-x64-Release-.etl - Windows Performance Analyzer
File Trace Profiles Window Help
Graph Explorer - log--ken-spi...
System Activity
Generic Events Activity by Provide...
Computation
CPU Usage (Sampled) Utilization...
DPC and ISR Usage by Module, Stack
Flame by Process, Stack
Utilization by COFF Group, Module,...
Utilization By CPU
Utilization By Priority
Utilization By Process
Utilization By Process and Thread
Utilization by Process, Thread, Stack
CPU Usage (Precise) Utilization by...
DPC/ISR DPC/ISR Duration by Mod...
Diagnostic Console
Symbols Hub
Getting Started
Analysis
CPU Usage (Sampled) Flame by Process, Stack *
Line # Process Module Stack Cou... Weight (in... TimeStamp (s) % Weight Legend
1 test.exe (7196) 340,387 41,560.591200 8.13
2 test.exe 207,016 25,276.837600 4.95
3 ntoskrnl.exe 110,294 13,467.423500 2.64
4 [Root] 108,751 13,279.308700 2.60
5 |- ntdll.dll!RtlUserThreadStart 105,626 12,896.637600 2.52
6 |- kernel32.dll!BaseThreadInitThunk 105,624 12,896.393400 2.52
7 |- test.exe!ThreadProcCallback 105,613 12,895.040900 2.52
8 |- ntdll.dll!RtlAcquireSRWLockE... 62,394 7,618.321000 1.49
9 |- ntdll.dll!RtlpWakeSRWLock 43,176 5,271.472900 1.03
Found ntdll.dll RtlAcquireSRWLock Exclusive.
CPU Usage (Sampled) Utilization By Process and Thread *
Series
test.exe (7196)
test.exe
ntoskrnl.exe
ntdll.dll
RtlAcquireSRWLo...
RtlpWakeSRWLock
ZwWaitForAlertB...
% Weight using resource tim
Line # Process Module Function Thread ID Cou... Weight (in... TimeStamp (s) % Weight Legend
1 test.exe (7196) 340,387 41,560.591200 8.13
2 test.exe ThreadProcCallback 207,016 25,276.837600 4.95
3 ntoskrnl.exe 110,294 13,467.423500 2.64
4 ntdll.dll 16,278 1,987.096900 0.39
5 RtlAcquireSRWLockExclusive 11,385 1,390.185700 0.27
6 RtlpWakeSRWLock 2,911 354.742800 0.07
7 ZwWaitForAlertByThreadId 779 95.115900 0.02
8 RtlpWaitCouldDeadlock 389 47.496900 0.01
9 RtlpOptimizeSRWLockList 388 47.374800 0.01
Start: 0.001369100s
End: 31.941493100s
Duration: 31.940124000s
Analysis Assistant
Details

EXAMPLE 6 CRITICAL SECTION

```
// MyLock not required. Let the OS do the work!
```

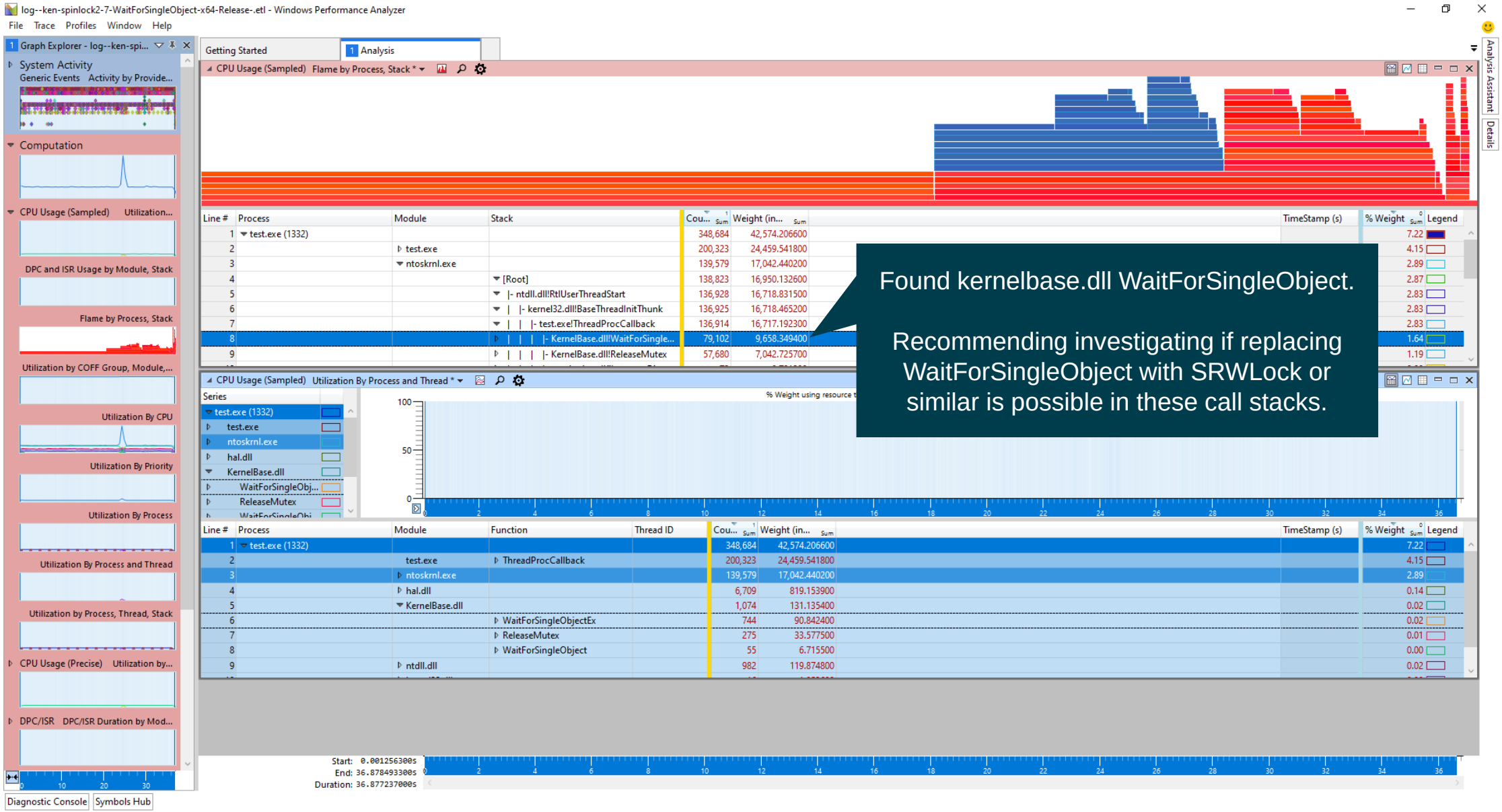
```
CRITICAL_SECTION cs;
DWORD WINAPI ThreadProcCallback(LPVOID data) {
    alignas(64) float a[LEN][4][4];
    std::fill((float*)a, (float*)(a + LEN), 0.0f);
    float r = 0.0;
    for (size_t iter = 0; iter < 100000; iter++) {
        EnterCriticalSection(&cs);
        for (int m = 0; m < LEN; m++)
            for (int i = 0; i < 4; i++)
                for (int j = 0; j < 4; j++)
                    for (int k = 0; k < 4; k++)
                        a[m][i][j] += b[m][i][k] * c[m][k][j];
        r += std::accumulate((float*)a, \
            (float*)(a + LEN), 0.0f);
        LeaveCriticalSection(&cs);
    }
    printf("result: %f\n", r);
    return 0;
}
int main(int argc, char* argv[]) {
    InitializeCriticalSection(&cs);
    // ...
```


log--ken-spinlock2-6-EnterCriticalSection-x64-Release-.etl - Windows Performance Analyzer
File Trace Profiles Window Help
Graph Explorer - log--ken-spi...
System Activity
Generic Events Activity by Provide...
Computation
CPU Usage (Sampled) Utilization...
DPC and ISR Usage by Module, Stack
Flame by Process, Stack
Utilization by COFF Group, Module,...
Utilization By CPU
Utilization By Priority
Utilization By Process
Utilization By Process and Thread
Utilization by Process, Thread, Stack
CPU Usage (Precise) Utilization by...
DPC/ISR DPC/ISR Duration by Mod...
Diagnostic Console
Symbols Hub
Getting Started
Analysis
CPU Usage (Sampled) Flame by Process, Stack *
Line # Process Module Stack Cou... Weight (in... TimeStamp (s) % Weight Legend
1 test.exe (7208) 347,341 42,357.616100 8.25
2 test.exe 201,585 24,577.321200 4.78
3 ntoskrnl.exe 121,428 14,812.129900 2.88
4 [Root] 120,333 14,678.695500 2.86
5 |- ntdll.dll!RtlUserThreadStart 116,668 14,230.932700 2.77
6 |- kernel32.dll!BaseThreadInitThunk 116,664 14,230.444300 2.77
7 |- test.exe!ThreadProcCallback 116,650 14,228.821400 2.77
8 |- ntdll.dll!RtlEnterCriticalSection 66,255 8,089.882300 1.57
9 |- ntdll.dll!RtlLeaveCriticalSection 50,309 6,128.441600 1.19
Found ntdll.dll RtlpEnterCriticalSection. ☺
CPU Usage (Sampled) Utilization By Process and Thread *
Series
% Weight using resource time as
test.exe (7208)
test.exe
ntoskrnl.exe
ntdll.dll
RtlpEnterCriticalS...
RtlpWaitOnAddr...
RtlpWaitOnCritic...
Line # Process Module Function Thread ID Cou... Weight (in... TimeStamp (s) % Weight Legend
1 test.exe (7208) 347,341 42,357.616100 8.25
2 test.exe ThreadProcCallback 201,585 24,577.321200 4.78
3 ntoskrnl.exe 121,428 14,812.129900 2.88
4 ntdll.dll 17,927 2,188.110400 0.43
5 RtlpEnterCriticalSectionCont... 9,482 1,157.755800 0.23
6 RtlpWaitOnAddressWithTime... 2,037 248.717700 0.05
7 RtlpWaitOnCriticalSection 1,352 165.092700 0.03
8 RtlLeaveCriticalSection 1,250 152.285300 0.03
9 RtlpWakeByAddress 1,006 122.342200 0.02
Start: 0.001316200s
End: 32.107659300s
Duration: 32.106343100s
Analysis Assistant
Details

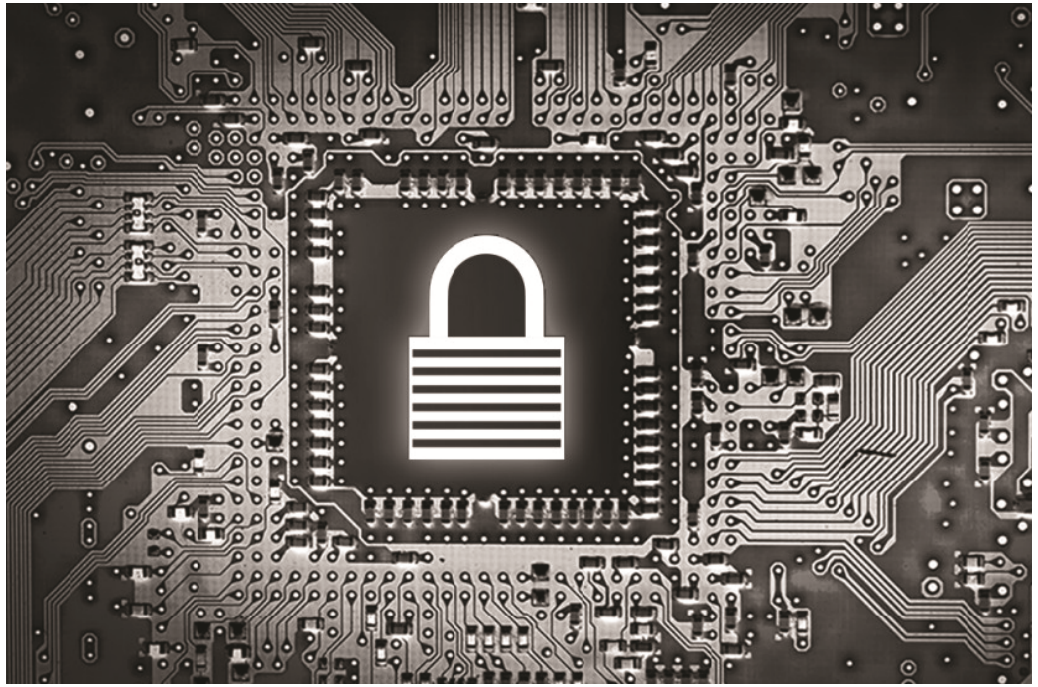
EXAMPLE 7 WAITFORSINGLEOBJECT

```
// MyLock not required. Let the OS do the work!
```

```
HANDLE hMutex;
DWORD WINAPI ThreadProcCallback(LPVOID data) {
    alignas(64) float a[LEN][4][4];
    std::fill((float*)a, (float*)(a + LEN), 0.0f);
    float r = 0.0;
    for (size_t iter = 0; iter < 100000; iter++) {
        WaitForSingleObject(hMutex, INFINITE);
        for (int m = 0; m < LEN; m++)
            for (int i = 0; i < 4; i++)
                for (int j = 0; j < 4; j++)
                    for (int k = 0; k < 4; k++)
                        a[m][i][j] += b[m][i][k] * c[m][k][j];
        r += std::accumulate((float*)a, \
            (float*)(a + LEN), 0.0f);
        ReleaseMutex(hMutex);
    }
    printf("result: %f\n", r);
    return 0;
}
int main(int argc, char* argv[]) {
    hMutex = CreateMutex(NULL,FALSE,NULL);
    // ...
```


log--ken-spinlock2-7-WaitForSingleObject-x64-Release-.etl - Windows Performance Analyzer
File Trace Profiles Window Help
Graph Explorer - log--ken-spi...
System Activity
Generic Events Activity by Provide...
Computation
CPU Usage (Sampled) Utilization...
DPC and ISR Usage by Module, Stack
Flame by Process, Stack
Utilization by COFF Group, Module,...
Utilization By CPU
Utilization By Priority
Utilization By Process
Utilization By Process and Thread
Utilization by Process, Thread, Stack
CPU Usage (Precise) Utilization by...
DPC/ISR DPC/ISR Duration by Mod...
Diagnostic Console
Symbols Hub
Getting Started
Analysis
CPU Usage (Sampled) Flame by Process, Stack *
Line # Process Module Stack Cou... Weight (in... TimeStamp (s) % Weight Legend
1 test.exe (1332) 348,684 42,574.206600 7.22
2 test.exe 200,323 24,459.541800 4.15
3 ntoskrnl.exe 139,579 17,042.440200 2.89
4 [Root] 138,823 16,950.132600 2.87
5 |- ntdll.dll!RtlUserThreadStart 136,928 16,718.831500 2.83
6 |- kernel32.dll!BaseThreadInitThunk 136,925 16,718.465200 2.83
7 |- test.exe!ThreadProcCallback 136,914 16,717.192300 2.83
8 |- KernelBase.dll!WaitForSingle... 79,102 9,658.349400 1.64
9 |- KernelBase.dll!ReleaseMutex 57,680 7,042.725700 1.19
CPU Usage (Sampled) Utilization By Process and Thread *
Series
test.exe (1332)
test.exe
ntoskrnl.exe
hal.dll
KernelBase.dll
WaitForSingleObj...
ReleaseMutex
Line # Process Module Function Thread ID Cou... Weight (in... TimeStamp (s) % Weight Legend
1 test.exe (1332) 348,684 42,574.206600 7.22
2 test.exe ThreadProcCallback 200,323 24,459.541800 4.15
3 ntoskrnl.exe 139,579 17,042.440200 2.89
4 hal.dll 6,709 819.153900 0.14
5 KernelBase.dll 1,074 131.135400 0.02
6 WaitForSingleObjectEx 744 90.842400 0.02
7 ReleaseMutex 275 33.577500 0.01
8 WaitForSingleObject 55 6.715500 0.00
9 ntdll.dll 982 119.874800 0.02
Start: 0.001256300s
End: 36.878493300s
Duration: 36.877237000s
Found kernelbase.dll WaitForSingleObject.
Recommending investigating if replacing WaitForSingleObject with SRWLock or similar is possible in these call stacks.
Analysis Assistant
Details

REMINDER



- Recommended modern OS synchronization APIs:
  - std::mutex
  - std::shared_mutex
  - SRWLock
  - EnterCriticalSection

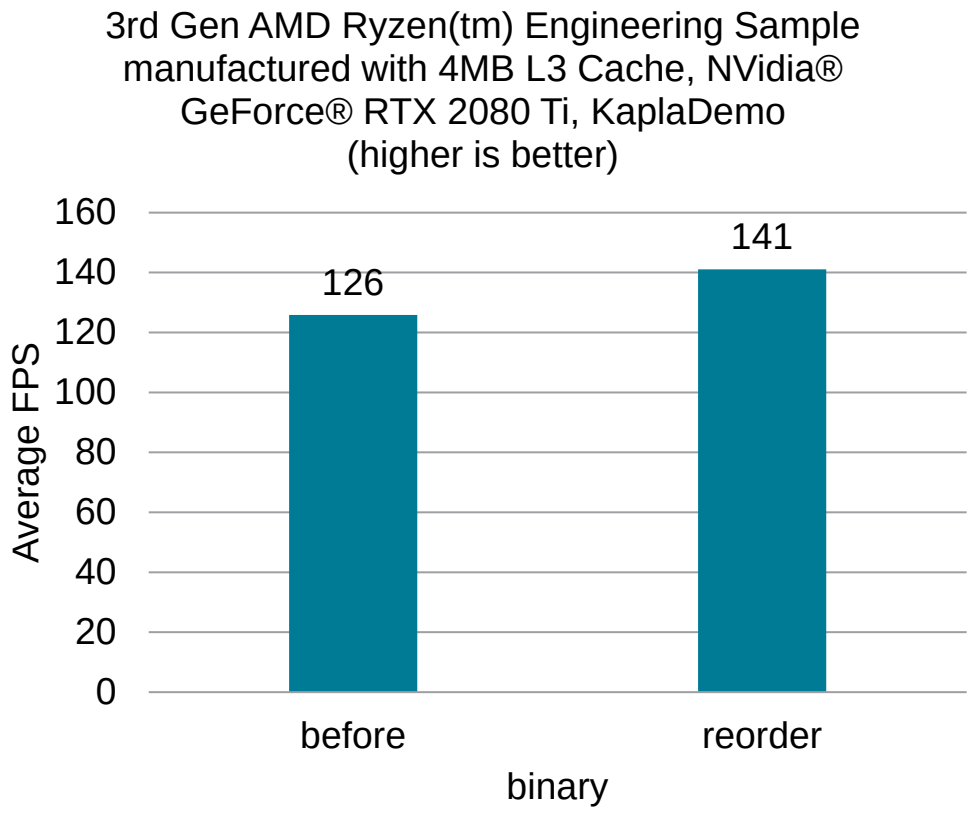
REORDER HOT STRUCT MEMBERS

SUMMARY

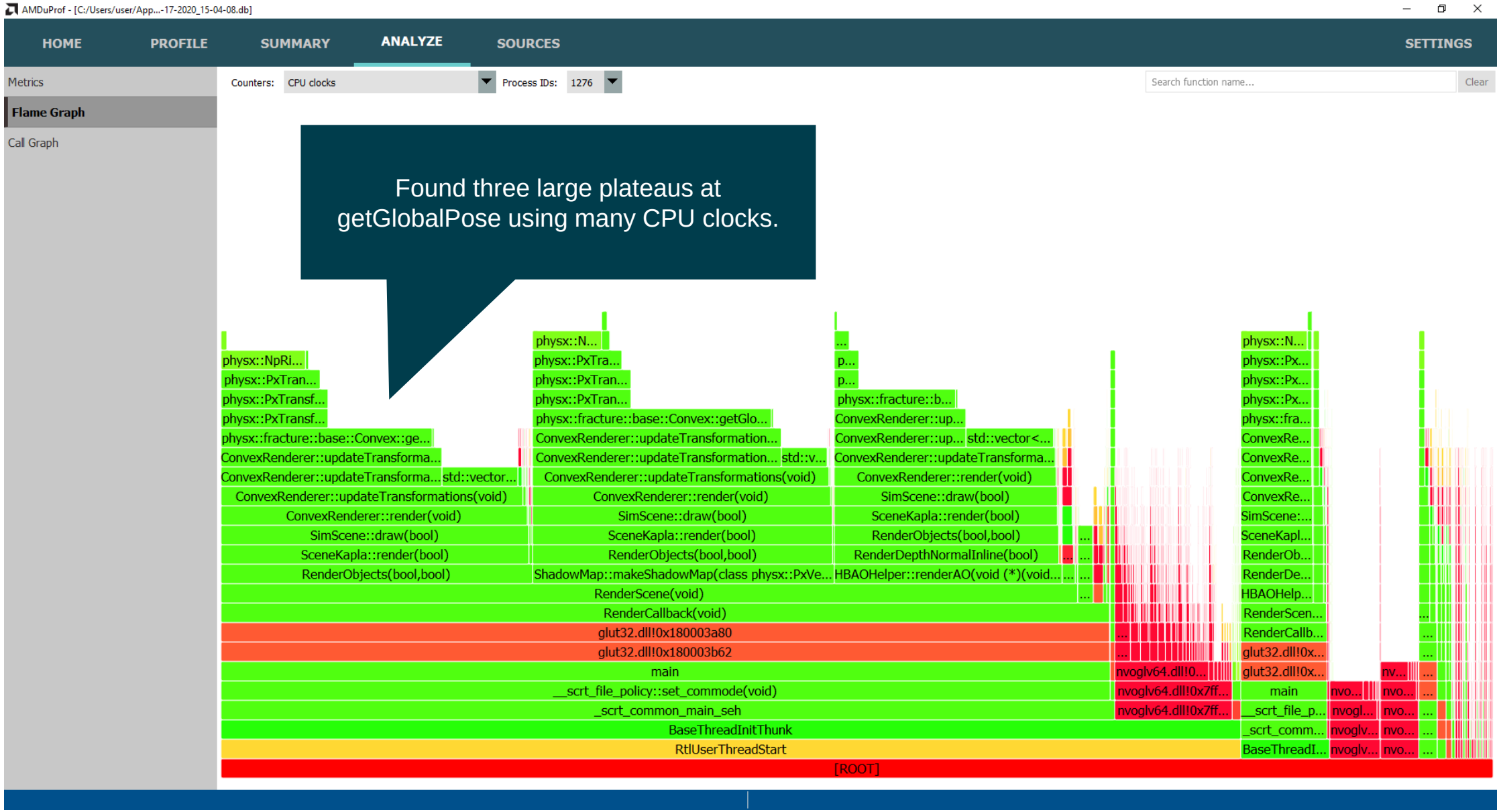
- Use AMDuProf to find plateaus of hot functions where there are many Data Cache refills from DRAM.
- If the hot function includes a loop which indirectly accesses struct data members spread over many cache lines, try reordering the struct.

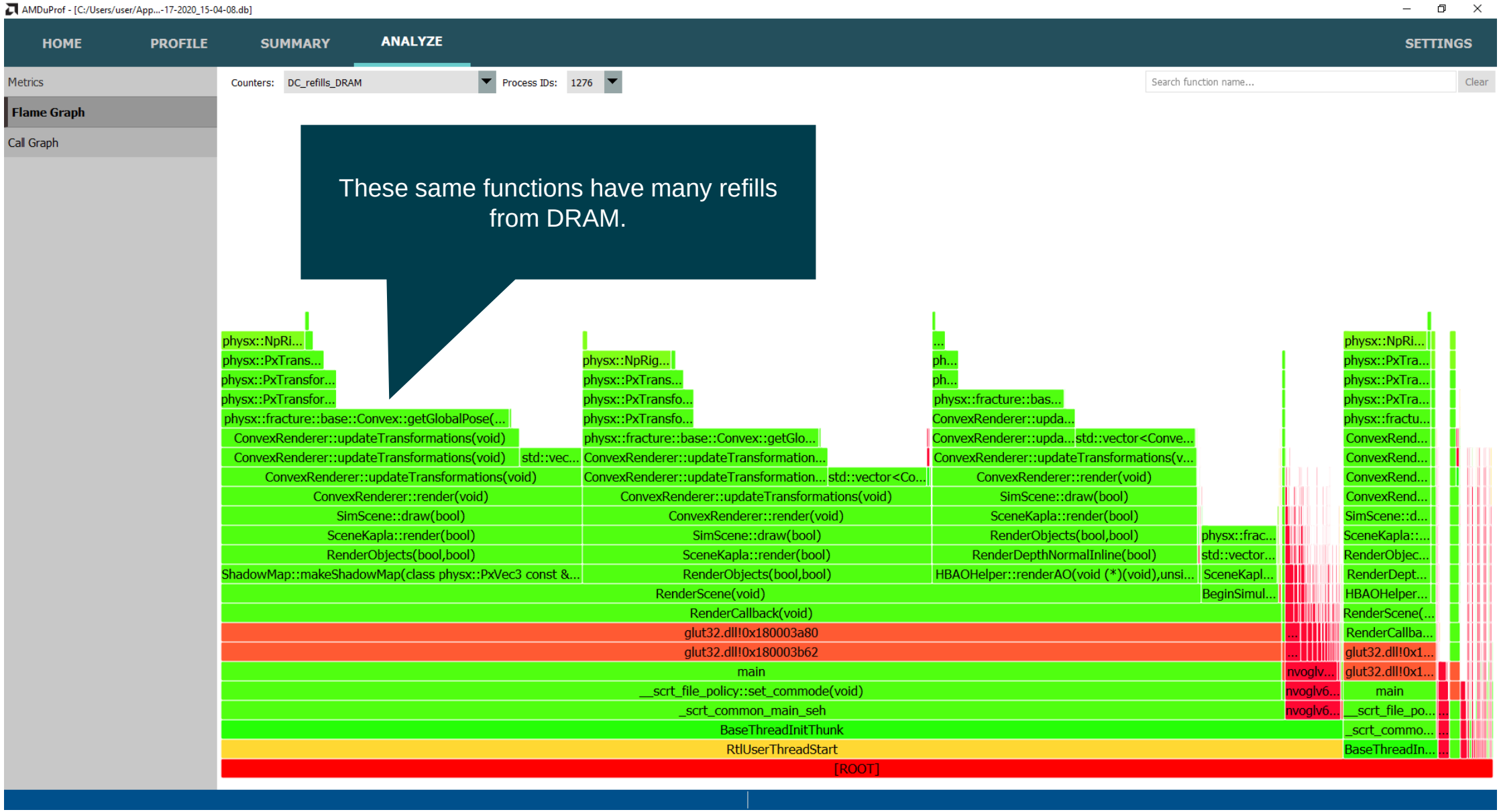
```
#if 0
/* bad */
struct S { double x, y, z, w; char name[256]; double s, t, u, v;};
#else
/* good */
struct S { double x, s, y, z, w; char name[256]; double t, u, v;};
#endif
for (size_t i = 0; i < list.size(); i++) {
   const S* e = list[i];
   foo(e->x);
   bar(e->s);
}
```

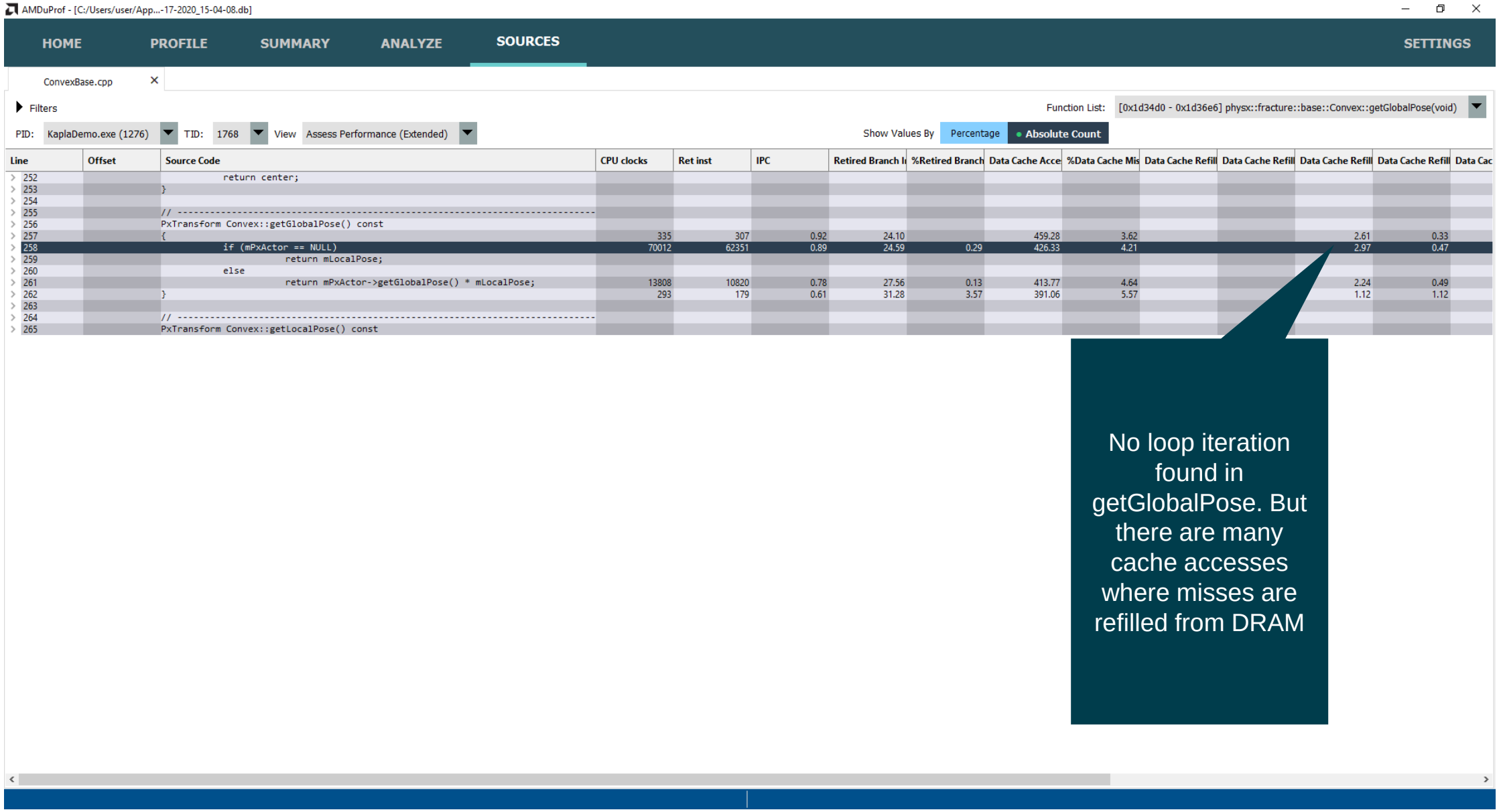
PERFORMANCE

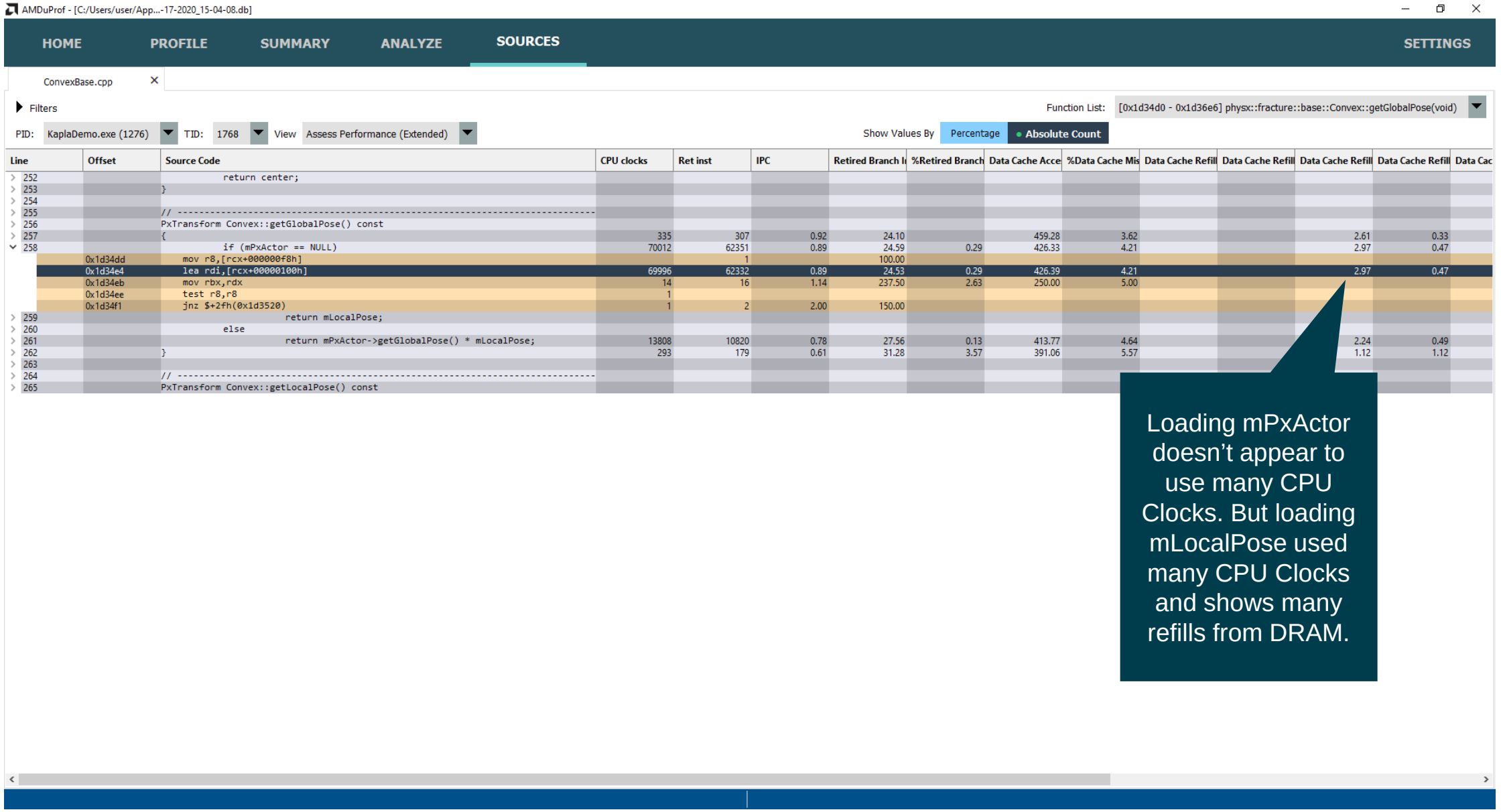


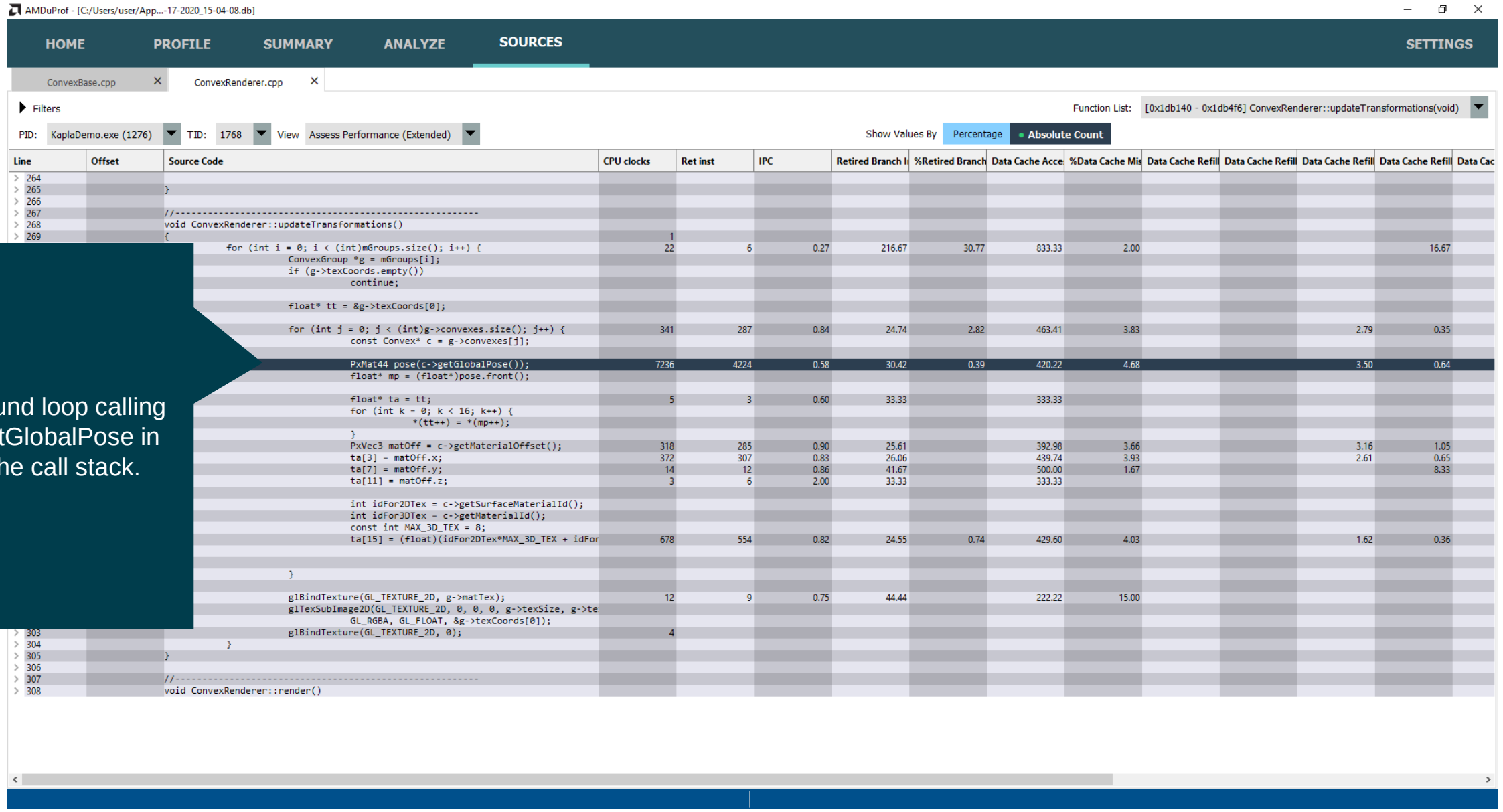
- 12% faster after optimization!
- Performance of binaries compiled with Microsoft Visual Studio 2019 v16.4.5.
- Testing done by AMD technology labs, February 15, 2020 on the following system. Test configuration: 3rd Gen AMD Ryzen™ Engineering Sample manufactured with 4MB L3 Cache, AMD Wraith Prism Cooler, 16GB (2 x 8GB DDR4-3200 at 22-22-22-52) memory, NVidia® GeForce® RTX 2080 Ti GPU with driver 441.87 (December 24, 2019), 2TB M.2 NVME SSD, AMD Ryzen™ Reference Motherboard, Windows® 10 x64 build 1909, 1920x1080 resolution. Actual results may vary.


AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
Metrics
Flame Graph
Call Graph
Counters:
CPU clocks
Process IDs:
1276
Search function name...
Clear
Found three large plateaus at getGlobalPose using many CPU clocks.
physx::NpRi...
physx::PxTran...
physx::PxTransf...
physx::PxTransf...
physx::fracture::base::Convex::ge...
ConvexRenderer::updateTransforma...
ConvexRenderer::updateTransforma...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commode(void)
_scrt_common_main_seh
BaseThreadInitThunk
RtlUserThreadStart
[ROOT]
physx::N...
physx::PxTra...
physx::PxTran...
physx::PxTran...
physx::fracture::base::Convex::getGlo...
ConvexRenderer::updateTransformation...
ConvexRenderer::updateTransformation...
std::v...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
ShadowMap::makeShadowMap(class physx::PxVe...
p...
p...
physx::fracture::b...
ConvexRenderer::up...
ConvexRenderer::up...
std::vector<...
ConvexRenderer::updateTransforma...
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderDepthNormalInline(bool)
HBAOHelper::renderAO(void (*)(void...
physx::N...
physx::Px...
physx::Px...
physx::Px...
physx::fra...
ConvexRe...
ConvexRe...
ConvexRe...
ConvexRe...
SimScene:...
SceneKapl...
RenderOb...
RenderDe...
HBAOHelp...
RenderScen...
RenderCallb...
glut32.dll!0x...
glut32.dll!0x...
nvoglv64.dll!0...
nvoglv64.dll!0x7ff...
nvoglv64.dll!0x7ff...
main
__scrt_file_p...
_scrt_comm...
BaseThreadI...
nv...
nvo...
nvogl...
nvoglv...
nvo...
nvo...
nvo...
nvo...

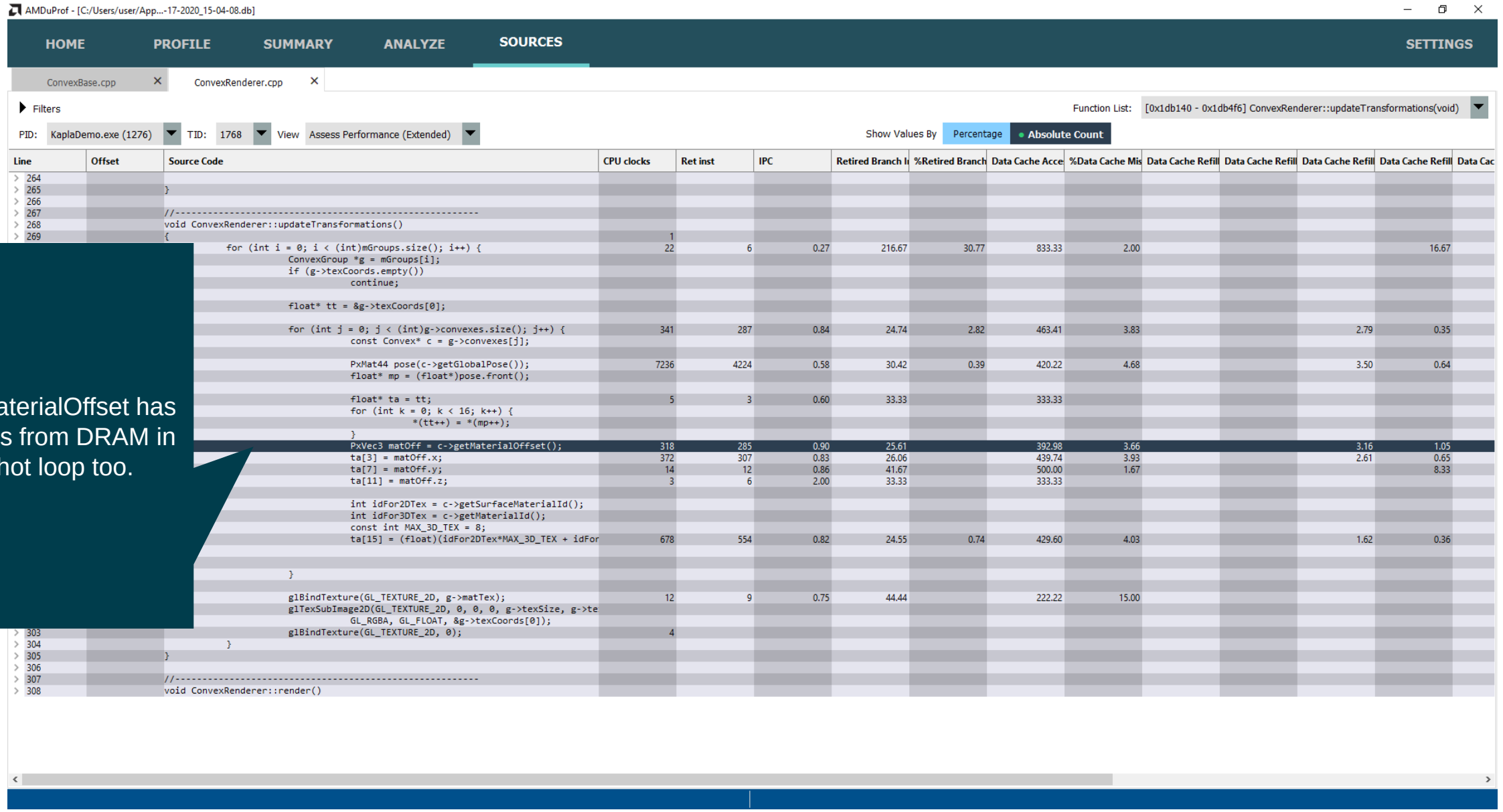

AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Metrics
Flame Graph
Call Graph
Counters:
DC_refills_DRAM
Process IDs:
1276
Search function name...
Clear
These same functions have many refills from DRAM.
physx::NpRi...
physx::PxTrans...
physx::PxTransfor...
physx::PxTransfor...
physx::fracture::base::Convex::getGlobalPose(...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::updateTransformations(void)
std::vec...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
ShadowMap::makeShadowMap(class physx::PxVec3 const &...
physx::NpRig...
physx::PxTrans...
physx::PxTransfo...
physx::PxTransfo...
physx::fracture::base::Convex::getGlo...
ConvexRenderer::updateTransformation...
ConvexRenderer::updateTransformation...
std::vector<Co...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commode(void)
_scrt_common_main_seh
BaseThreadInitThunk
RtlUserThreadStart
[ROOT]
...
ph...
ph...
physx::fracture::bas...
ConvexRenderer::upda...
ConvexRenderer::upda...
std::vector<Conve...
ConvexRenderer::updateTransformations(v...
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderDepthNormalInline(bool)
HBAOHelper::renderAO(void (*)(void),unsi...
physx::frac...
std::vector...
SceneKapl...
BeginSimul...
...
...
nvoglv...
nvoglv6...
nvoglv6...
physx::NpRi...
physx::PxTra...
physx::PxTra...
physx::PxTra...
physx::fractu...
ConvexRend...
ConvexRend...
ConvexRend...
ConvexRend...
SimScene::d...
SceneKapla::...
RenderObjec...
RenderDept...
HBAOHelper...
RenderScene(...
RenderCallba...
glut32.dll!0x1...
glut32.dll!0x1...
main
__scrt_file_po...
_scrt_commo...
BaseThreadIn...
...
...
...
...


AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
ConvexBase.cpp
Filters
Function List: [0x1d34d0 - 0x1d36e6] physx::fracture::base::Convex::getGlobalPose(void)
PID: KaplaDemo.exe (1276)
TID: 1768
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line Offset Source Code CPU clocks Ret inst IPC Retired Branch I %Retired Branch Data Cache Acce %Data Cache Mis Data Cache Refill Data Cache Refill Data Cache Refill Data Cache Refill Data Cac
252 return center;
253 }
254
255 // ----------
256 PxTransform Convex::getGlobalPose() const
257 { 335 307 0.92 24.10 459.28 3.62 2.61 0.33
258 if (mPxActor == NULL) 70012 62351 0.89 24.59 0.29 426.33 4.21 2.97 0.47
259 return mLocalPose;
260 else
261 return mPxActor->getGlobalPose() * mLocalPose; 13808 10820 0.78 27.56 0.13 413.77 4.64 2.24 0.49
262 } 293 179 0.61 31.28 3.57 391.06 5.57 1.12 1.12
263
264 // ----------
265 PxTransform Convex::getLocalPose() const
No loop iteration found in getGlobalPose. But there are many cache accesses where misses are refilled from DRAM

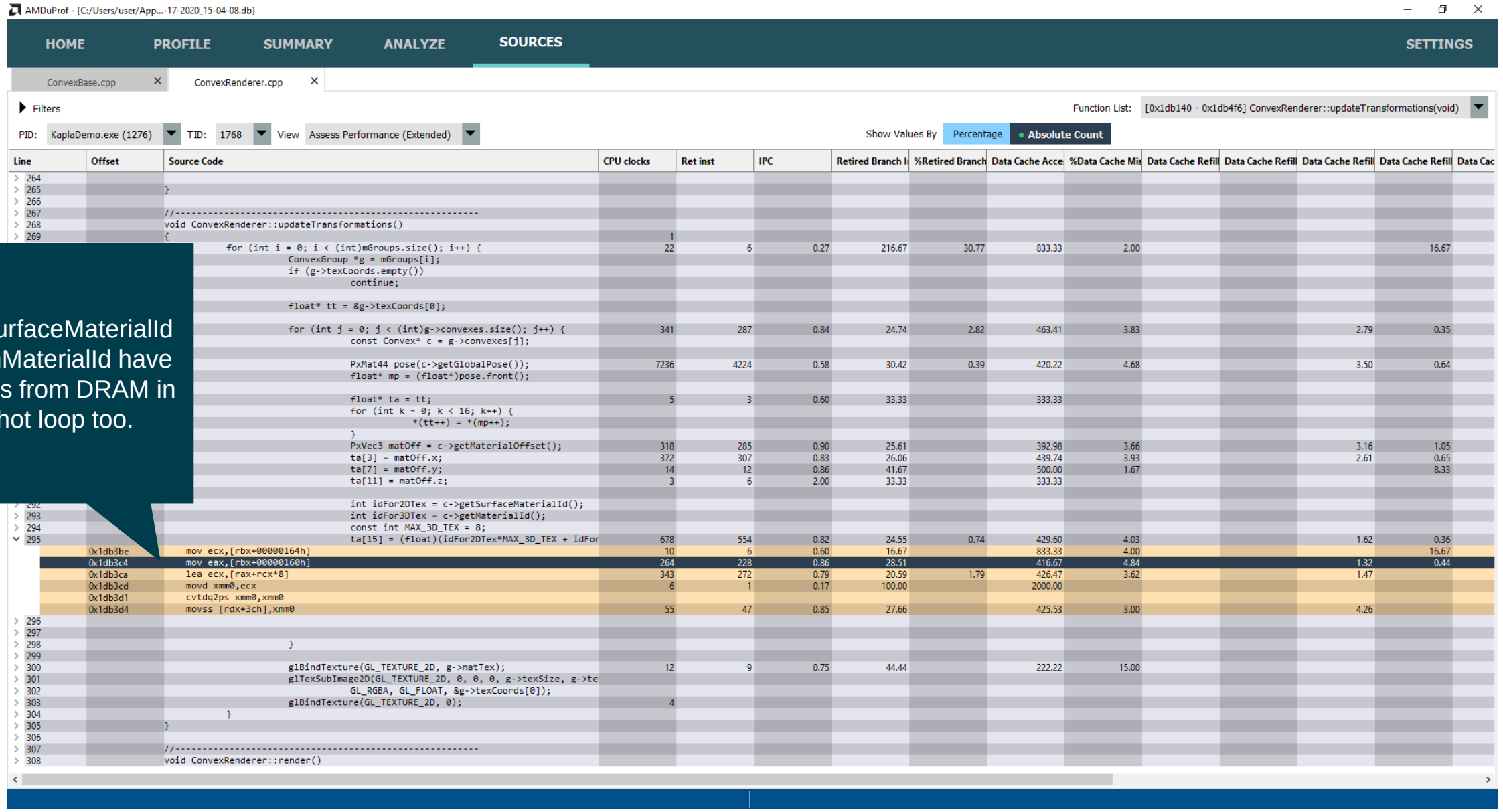

AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
ConvexBase.cpp
Filters
Function List: [0x1d34d0 - 0x1d36e6] physx::fracture::base::Convex::getGlobalPose(void)
PID: KaplaDemo.exe (1276)
TID: 1768
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line | Offset | Source Code | CPU clocks | Ret inst | IPC | Retired Branch I | %Retired Branch | Data Cache Acce | %Data Cache Mis | Data Cache Refill | Data Cache Refill | Data Cache Refill | Data Cache Refill | Data Cac
252 return center;
253 }
254
255 // ----------
256 PxTransform Convex::getGlobalPose() const
257 { 335 307 0.92 24.10 459.28 3.62 2.61 0.33
258 if (mPxActor == NULL) 70012 62351 0.89 24.59 0.29 426.33 4.21 2.97 0.47
0x1d34dd mov r8,[rcx+000000f8h] 1 100.00
0x1d34e4 lea rdi,[rcx+00000100h] 69996 62332 0.89 24.53 0.29 426.39 4.21 2.97 0.47
0x1d34eb mov rbx,rdx 14 16 1.14 237.50 2.63 250.00 5.00
0x1d34ee test r8,r8 1
0x1d34f1 jnz $+2fh(0x1d3520) 1 2 2.00 150.00
259 return mLocalPose;
260 else
261 return mPxActor->getGlobalPose() * mLocalPose; 13808 10820 0.78 27.56 0.13 413.77 4.64 2.24 0.49
262 } 293 179 0.61 31.28 3.57 391.06 5.57 1.12 1.12
263
264 // ----------
265 PxTransform Convex::getLocalPose() const
Loading mPxActor doesn’t appear to use many CPU Clocks. But loading mLocalPose used many CPU Clocks and shows many refills from DRAM.



Found loop calling getGlobalPose in the call stack.



mMaterialOffset has refills from DRAM in hot loop too.



mSurfaceMaterialId & mMaterialId have refills from DRAM in hot loop too.

CODE SAMPLE

https://github.com/NVIDIAGameWorks/PhysX/tree/4.1/physx
ConvexBase.h

```
// Before Reorder Optimization

protected:
    void clear();
…
    Compound *mParent;
    PxRigidActor *mPxActor;
    PxTransform  mLocalPose;
    PxConvexMesh *mPxConvexMesh;

    PxBounds3 mBounds;
    mutable float mVolume;
    mutable bool mVolumeDirty;
    PxVec3 mMaterialOffset;
    float mTexScale;
    int mModelIslandNr;
…
```

```
// After Reorder Optimization

protected:
    // Reorder Begin
    PxRigidActor* mPxActor;
    PxTransform  mLocalPose;
    PxVec3 mMaterialOffset;
    int mSurfaceMaterialId;
    int mMaterialId;
    // Reorder End

    void clear();
    void finalize();
    void updateBounds();
    void updatePlanes();
…
```

WINDBG

https://github.com/NVIDIAGameWorks/PhysX/tree/4.1/physx
ConvexBase.h

Before optimization hot data spread over many cache lines.

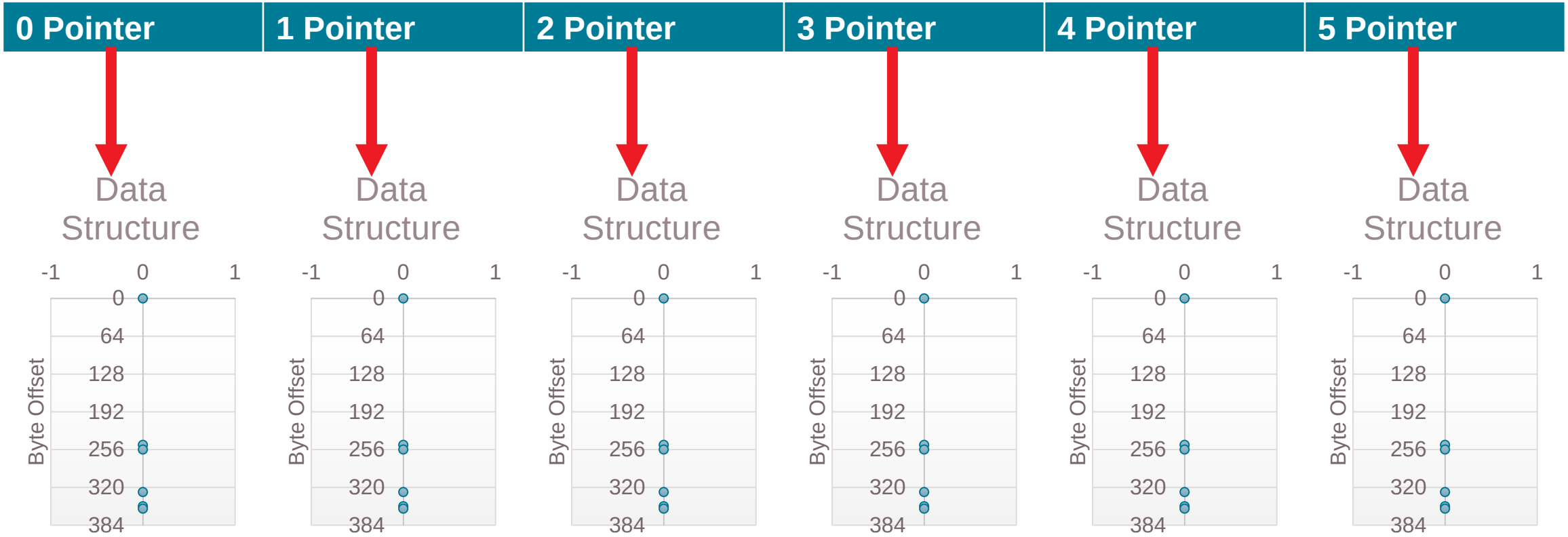
WinDbg
dt KaplaDemo!Convex
  +0x000 __VFN_table : Ptr64
  +0x008 mScene : Ptr64 physx::fracture::base::SimScene
…
  +0x0f8 mPxActor : Ptr64 physx::PxRigidActor
  +0x100 mLocalPose : physx::PxTransform
…
  +0x148 mMaterialOffset : physx::PxVec3
…
  +0x160 mMaterialId : Int4B
  +0x164 mSurfaceMaterialId : Int4B
…

After optimization hot data spread on one cache line if struct aligned.

WinDbg
dt KaplaDemo!Convex
  +0x000 __VFN_table : Ptr64
  +0x008 mPxActor : Ptr64 physx::PxRigidActor
  +0x010 mLocalPose : physx::PxTransform
  +0x02c mMaterialOffset : physx::PxVec3
  +0x038 mSurfaceMaterialId : Int4B
  +0x03c mMaterialId : Int4B
  +0x040 mScene : Ptr64 physx::fracture::base::SimScene
…

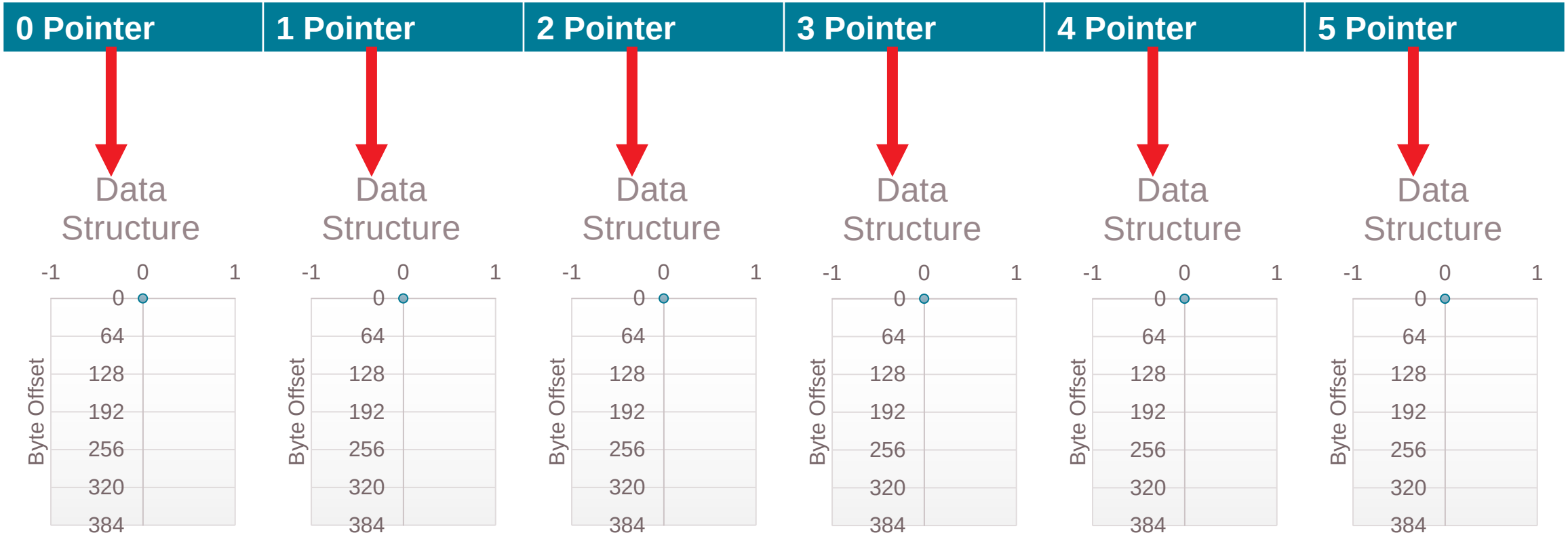
ACCESS PATTERN BEFORE OPTIMIZATION

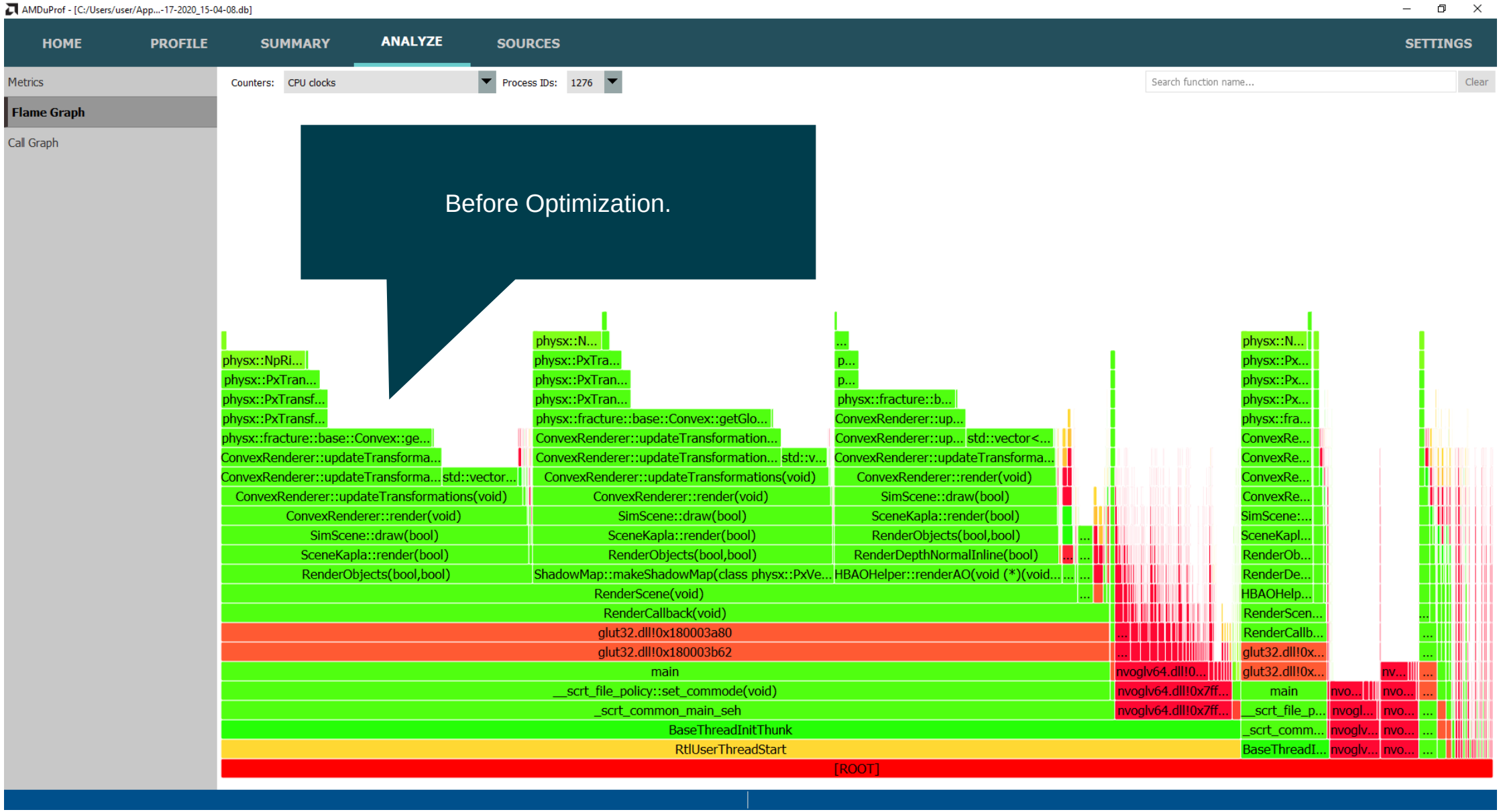
- Contiguous array of pointers (8 Bytes) to data structures (372 Bytes each) with similar byte offset accesses.

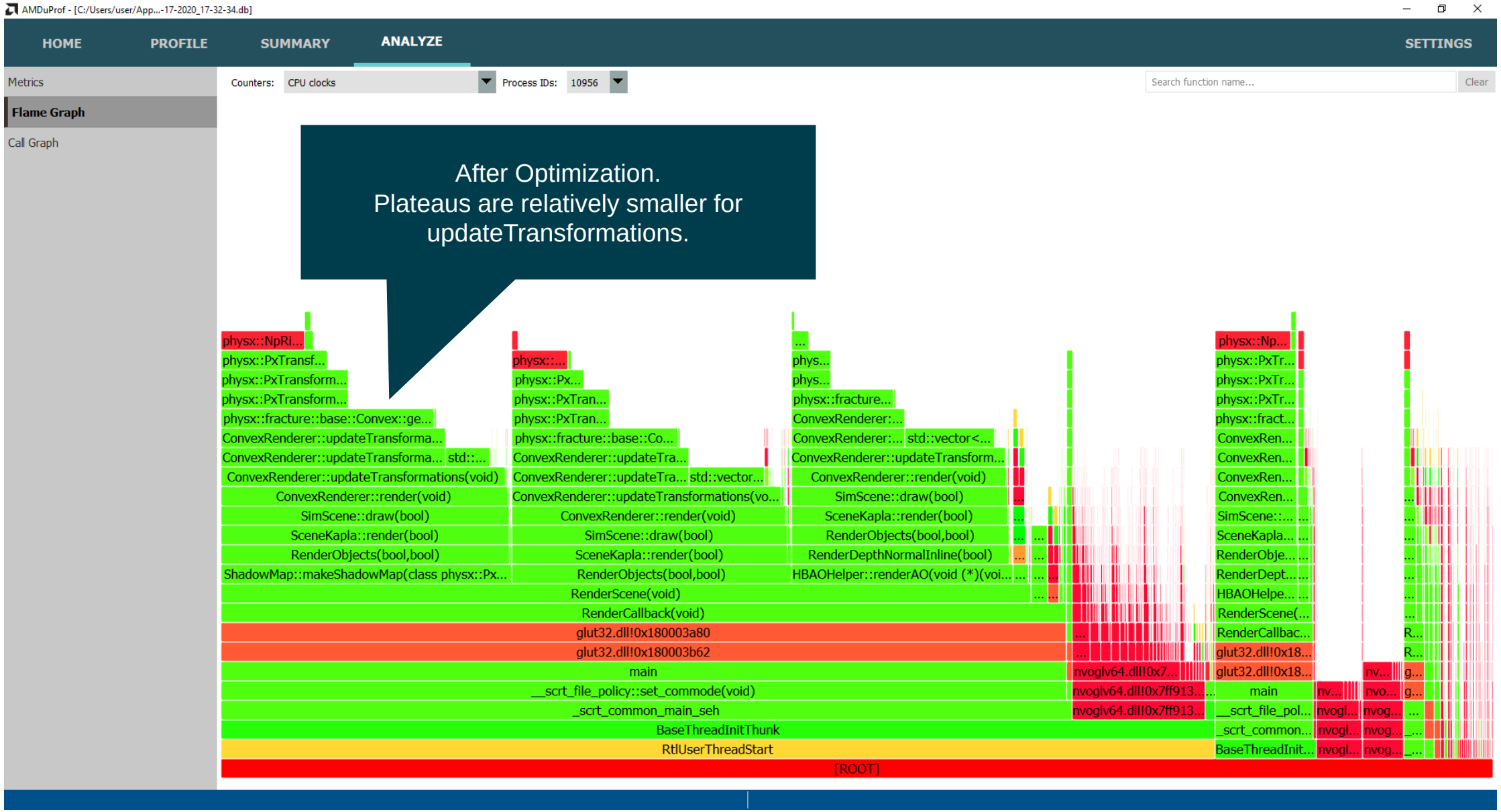


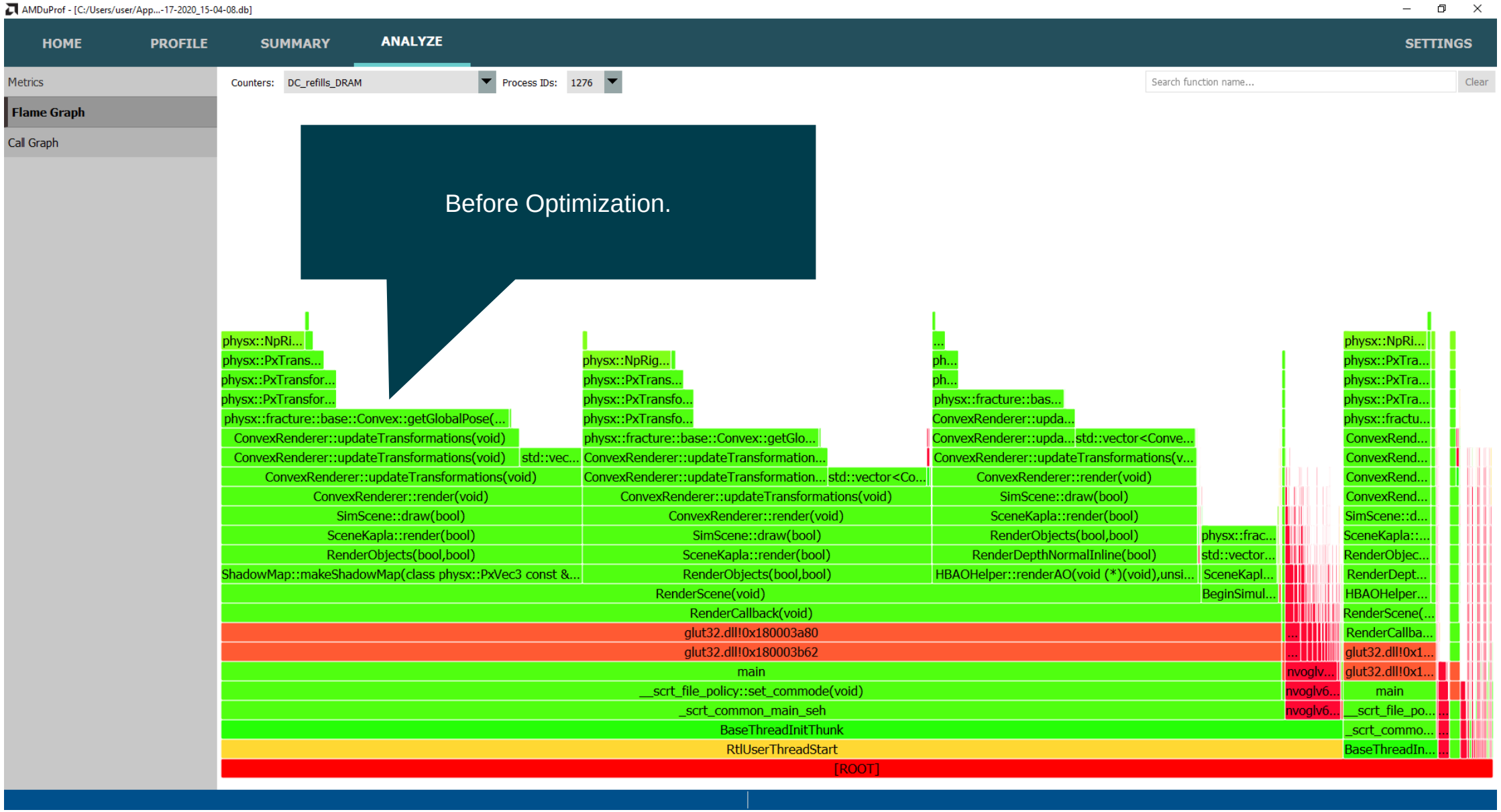
ACCESS PATTERN AFTER OPTIMIZATION

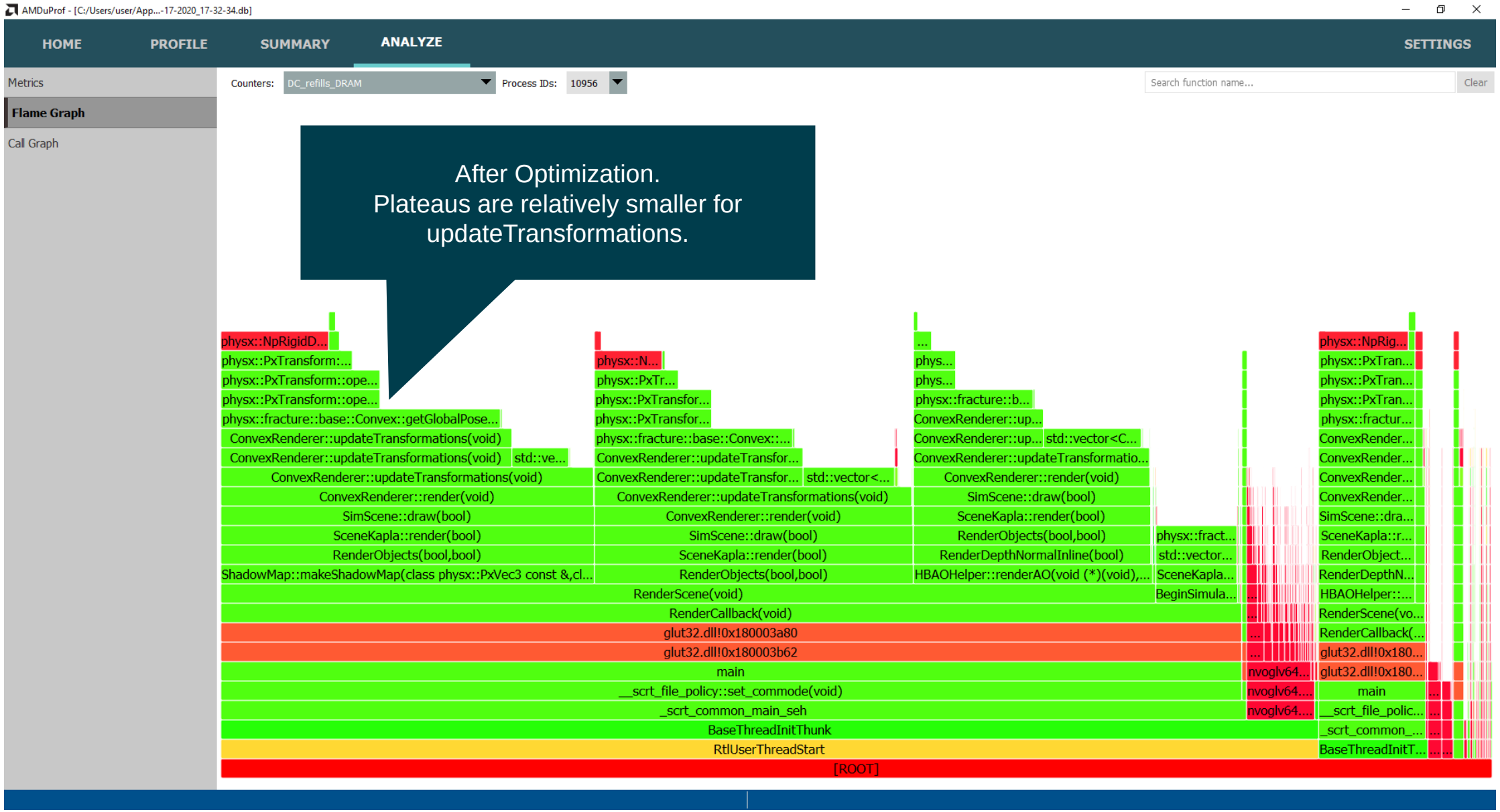
- Contiguous array of pointers (8 Bytes) to data structures (372 Bytes each) with similar byte offset accesses.

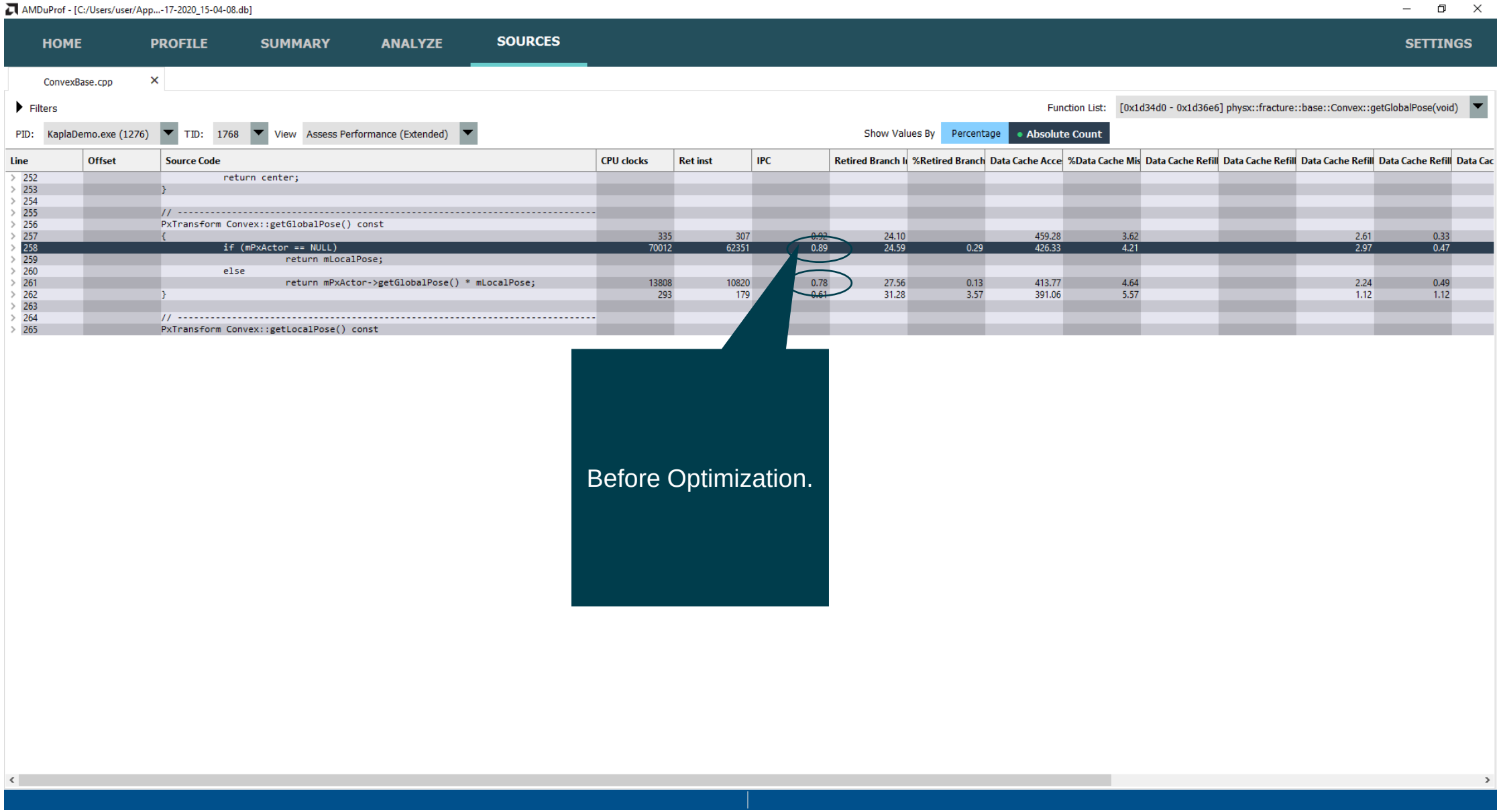


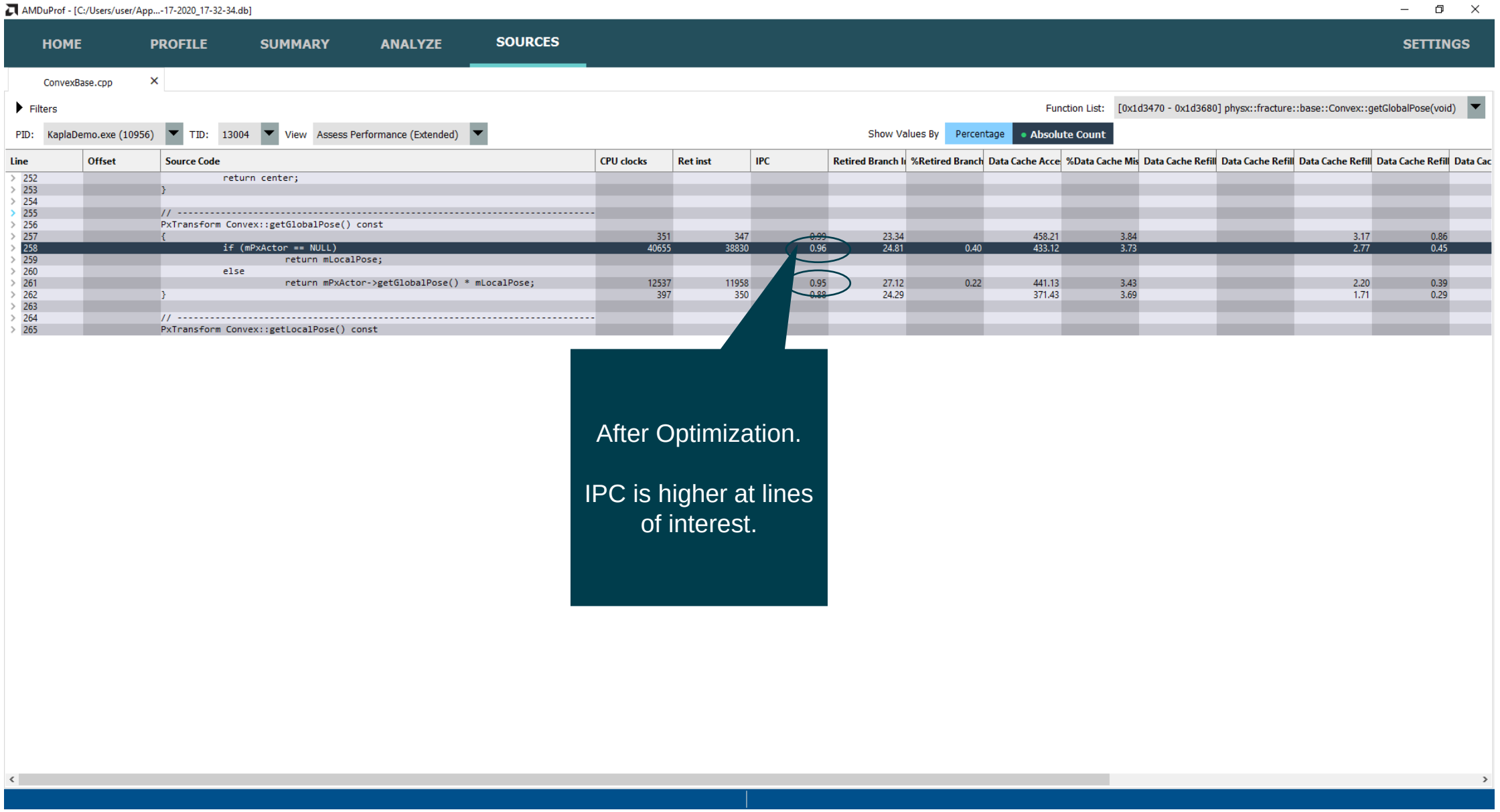

AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
Metrics
Flame Graph
Call Graph
Counters:
CPU clocks
Process IDs:
1276
Search function name...
Clear
Before Optimization.
physx::NpRi...
physx::PxTran...
physx::PxTransf...
physx::PxTransf...
physx::fracture::base::Convex::ge...
ConvexRenderer::updateTransforma...
ConvexRenderer::updateTransforma...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commode(void)
_scrt_common_main_seh
BaseThreadInitThunk
RtlUserThreadStart
[ROOT]
physx::N...
physx::PxTra...
physx::PxTran...
physx::PxTran...
physx::fracture::base::Convex::getGlo...
ConvexRenderer::updateTransformation...
ConvexRenderer::updateTransformation...
std::v...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
ShadowMap::makeShadowMap(class physx::PxVe...
physx::N...
physx::Px...
physx::Px...
physx::Px...
physx::fra...
ConvexRe...
ConvexRe...
ConvexRe...
ConvexRe...
SimScene:...
SceneKapl...
RenderOb...
RenderDe...
HBAOHelp...
RenderScen...
RenderCallb...
glut32.dll!0x...
glut32.dll!0x...
main
_scrt_file_p...
_scrt_comm...
BaseThreadI...
nvoglv64.dll!0...
nvoglv64.dll!0x7ff...
nvoglv64.dll!0x7ff...
nvo...
nvogl...
nvoglv...
nvoglv...
nv...
nvo...
nvo...
nvo...
nvo...

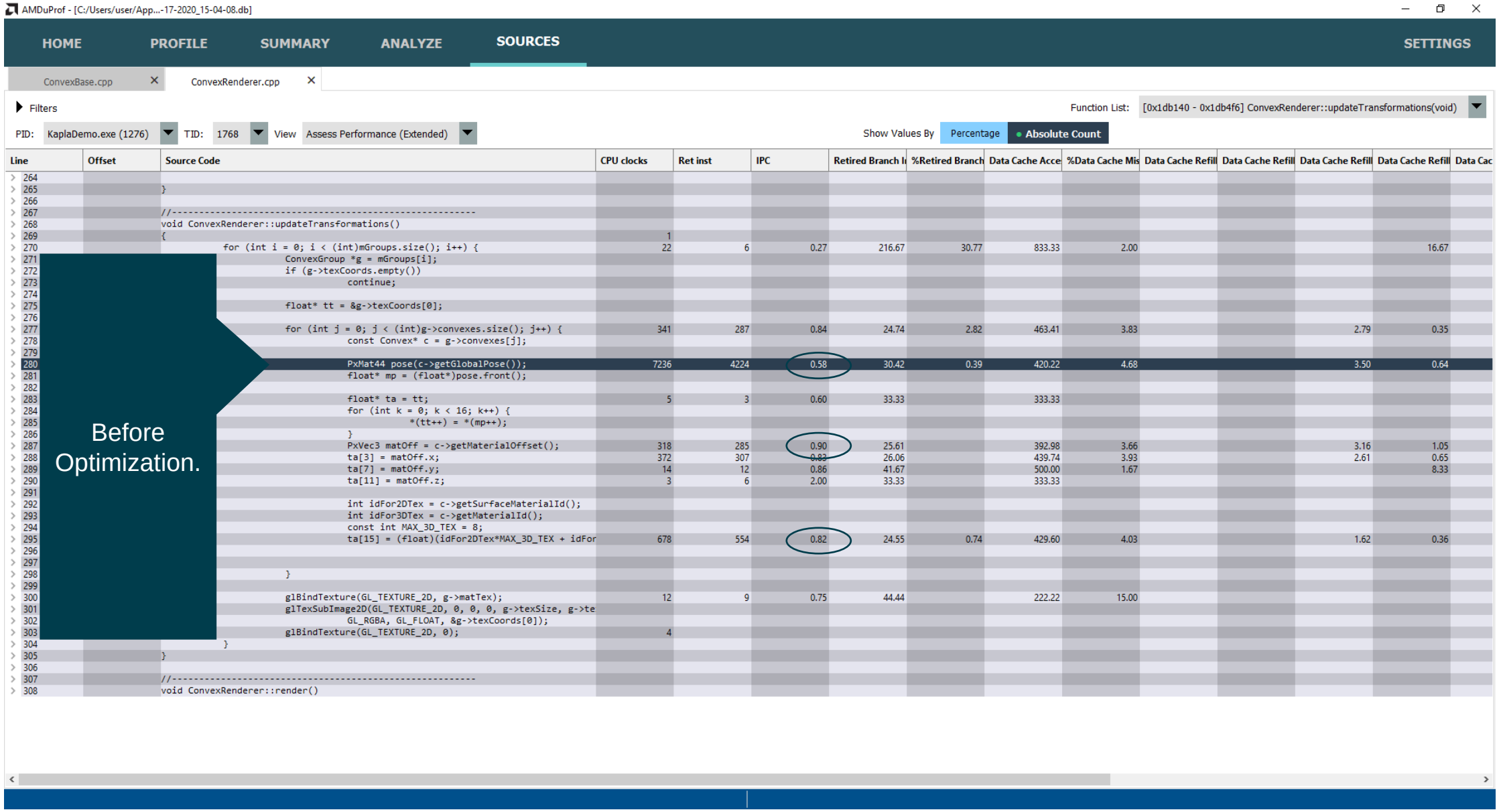

AMDuProf - [C:/Users/user/App...-17-2020_17-32-34.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Metrics
Flame Graph
Call Graph
Counters:
CPU clocks
Process IDs:
10956
Search function name...
Clear
After Optimization.
Plateaus are relatively smaller for
updateTransformations.
physx::NpRi...
physx::PxTransf...
physx::PxTransform...
physx::PxTransform...
physx::fracture::base::Convex::ge...
ConvexRenderer::updateTransforma...
ConvexRenderer::updateTransforma...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
ShadowMap::makeShadowMap(class physx::Px...
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commode(void)
_scrt_common_main_seh
BaseThreadInitThunk
RtlUserThreadStart
[ROOT]
physx::...
physx::Px...
physx::PxTran...
physx::PxTran...
physx::fracture::base::Co...
ConvexRenderer::updateTra...
ConvexRenderer::updateTra...
ConvexRenderer::updateTransformations(vo...
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
HBAOHelper::renderAO(void (*)(voi...
phys...
phys...
physx::fracture...
ConvexRenderer:...
ConvexRenderer:...
std::vector<...
ConvexRenderer::updateTransform...
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderDepthNormalInline(bool)
physx::Np...
physx::PxTr...
physx::PxTr...
physx::PxTr...
physx::fract...
ConvexRen...
ConvexRen...
ConvexRen...
ConvexRen...
SimScene::...
SceneKapla...
RenderObje...
RenderDept...
HBAOHelpe...
RenderScene(...
RenderCallbac...
glut32.dll!0x18...
glut32.dll!0x18...
nvoglv64.dll!0x7...
nvoglv64.dll!0x7ff913...
nvoglv64.dll!0x7ff913...
main
__scrt_file_pol...
_scrt_common...
BaseThreadInit...
nv...
nvo...
nvogl...
nvog...
nvogl...
nvog...
nvogl...
nvog...

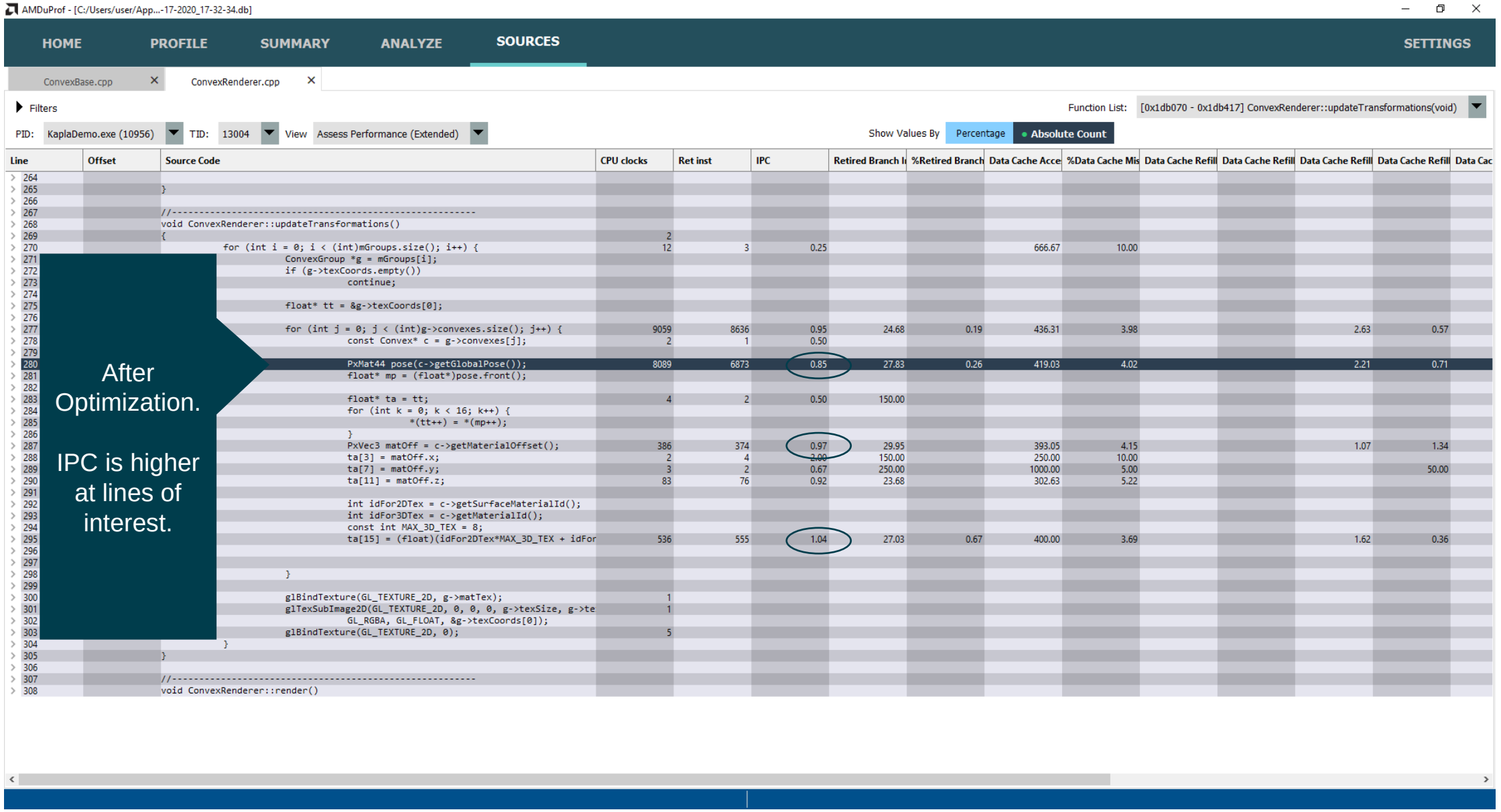

AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Metrics
Flame Graph
Call Graph
Counters:
DC_refills_DRAM
Process IDs:
1276
Search function name...
Clear
Before Optimization.
physx::NpRi...
physx::PxTrans...
physx::PxTransfor...
physx::PxTransfor...
physx::fracture::base::Convex::getGlobalPose(...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::updateTransformations(void)
std::vec...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
ShadowMap::makeShadowMap(class physx::PxVec3 const &...
physx::NpRig...
physx::PxTrans...
physx::PxTransfo...
physx::PxTransfo...
physx::fracture::base::Convex::getGlo...
ConvexRenderer::updateTransformation...
ConvexRenderer::updateTransformation...
std::vector<Co...
ConvexRenderer::updateTransformations(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commode(void)
_scrt_common_main_seh
BaseThreadInitThunk
RtlUserThreadStart
[ROOT]
ph...
physx::fracture::bas...
ConvexRenderer::upda...
ConvexRenderer::upda...
std::vector<Conve...
ConvexRenderer::updateTransformations(v...
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderDepthNormalInline(bool)
HBAOHelper::renderAO(void (*)(void),unsi...
physx::frac...
std::vector...
SceneKapl...
BeginSimul...
nvoglv...
nvoglv6...
nvoglv6...
physx::NpRi...
physx::PxTra...
physx::PxTra...
physx::PxTra...
physx::fractu...
ConvexRend...
ConvexRend...
ConvexRend...
ConvexRend...
SimScene::d...
SceneKapla::...
RenderObjec...
RenderDept...
HBAOHelper...
RenderScene(...
RenderCallba...
glut32.dll!0x1...
glut32.dll!0x1...
main
__scrt_file_po...
_scrt_commo...
BaseThreadIn...


AMDuProf - [C:/Users/user/App...-17-2020_17-32-34.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Metrics
Flame Graph
Call Graph
Counters:
DC_refills_DRAM
Process IDs:
10956
Search function name...
Clear
After Optimization.
Plateaus are relatively smaller for
updateTransformations.
physx::NpRigidD...
physx::PxTransform:...
physx::PxTransform::ope...
physx::PxTransform::ope...
physx::fracture::base::Convex::getGlobalPose...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::updateTransformations(void)
std::ve...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
ShadowMap::makeShadowMap(class physx::PxVec3 const &,cl...
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commode(void)
_scrt_common_main_seh
BaseThreadInitThunk
RtlUserThreadStart
[ROOT]
physx::N...
physx::PxTr...
physx::PxTransfor...
physx::PxTransfor...
physx::fracture::base::Convex::...
ConvexRenderer::updateTransfor...
ConvexRenderer::updateTransfor...
std::vector<...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
HBAOHelper::renderAO(void (*)(void),...
...
phys...
phys...
physx::fracture::b...
ConvexRenderer::up...
ConvexRenderer::up...
std::vector<C...
ConvexRenderer::updateTransformatio...
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderDepthNormalInline(bool)
physx::fract...
std::vector...
SceneKapla...
BeginSimula...
...
physx::NpRig...
physx::PxTran...
physx::PxTran...
physx::PxTran...
physx::fractur...
ConvexRender...
ConvexRender...
ConvexRender...
ConvexRender...
SimScene::dra...
SceneKapla::r...
RenderObject...
RenderDepthN...
HBAOHelper::...
RenderScene(vo...
RenderCallback(...
glut32.dll!0x180...
glut32.dll!0x180...
nvoglv64...
nvoglv64...
nvoglv64...
main
__scrt_file_polic...
_scrt_common_...
BaseThreadInitT...


AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
ConvexBase.cpp
Filters
Function List: [0x1d34d0 - 0x1d36e6] physx::fracture::base::Convex::getGlobalPose(void)
PID: KaplaDemo.exe (1276)
TID: 1768
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line Offset Source Code CPU clocks Ret inst IPC Retired Branch I %Retired Branch Data Cache Acce %Data Cache Mis Data Cache Refill Data Cache Refill Data Cache Refill Data Cache Refill Data Cac
252 return center;
253 }
254
255 // ----------
256 PxTransform Convex::getGlobalPose() const
257 { 335 307 0.92 24.10 459.28 3.62 2.61 0.33
258 if (mPxActor == NULL) 70012 62351 0.89 24.59 0.29 426.33 4.21 2.97 0.47
259 return mLocalPose;
260 else
261 return mPxActor->getGlobalPose() * mLocalPose; 13808 10820 0.78 27.56 0.13 413.77 4.64 2.24 0.49
262 } 293 179 0.61 31.28 3.57 391.06 5.57 1.12 1.12
263
264 // ----------
265 PxTransform Convex::getLocalPose() const
Before Optimization.


AMDuProf - [C:/Users/user/App...-17-2020_17-32-34.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
ConvexBase.cpp
Filters
Function List: [0x1d3470 - 0x1d3680] physx::fracture::base::Convex::getGlobalPose(void)
PID: KaplaDemo.exe (10956)
TID: 13004
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line Offset Source Code CPU clocks Ret inst IPC Retired Branch I %Retired Branch Data Cache Acce %Data Cache Mis Data Cache Refill Data Cache Refill Data Cache Refill Data Cache Refill Data Cac
252 return center;
253 }
254
255 // ----------
256 PxTransform Convex::getGlobalPose() const
257 { 351 347 0.99 23.34 458.21 3.84 3.17 0.86
258 if (mPxActor == NULL) 40655 38830 0.96 24.81 0.40 433.12 3.73 2.77 0.45
259 return mLocalPose;
260 else
261 return mPxActor->getGlobalPose() * mLocalPose; 12537 11958 0.95 27.12 0.22 441.13 3.43 2.20 0.39
262 } 397 350 0.88 24.29 371.43 3.69 1.71 0.29
263
264 // ----------
265 PxTransform Convex::getLocalPose() const
After Optimization.
IPC is higher at lines of interest.


AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
ConvexBase.cpp
ConvexRenderer.cpp
Filters
Function List: [0x1db140 - 0x1db4f6] ConvexRenderer::updateTransformations(void)
PID: KaplaDemo.exe (1276)
TID: 1768
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line Offset Source Code CPU clocks Ret inst IPC Retired Branch I %Retired Branch Data Cache Acce %Data Cache Mis Data Cache Refill Data Cache Refill Data Cache Refill Data Cache Refill Data Cac
264
265 }
266
267 //----------
268 void ConvexRenderer::updateTransformations()
269 { 1
270 for (int i = 0; i < (int)mGroups.size(); i++) { 22 6 0.27 216.67 30.77 833.33 2.00 16.67
271 ConvexGroup *g = mGroups[i];
272 if (g->texCoords.empty())
273 continue;
274
275 float* tt = &g->texCoords[0];
276
277 for (int j = 0; j < (int)g->convexes.size(); j++) { 341 287 0.84 24.74 2.82 463.41 3.83 2.79 0.35
278 const Convex* c = g->convexes[j];
279
280 PxMat44 pose(c->getGlobalPose()); 7236 4224 0.58 30.42 0.39 420.22 4.68 3.50 0.64
281 float* mp = (float*)pose.front();
282
283 float* ta = tt; 5 3 0.60 33.33 333.33
284 for (int k = 0; k < 16; k++) {
285 *(tt++) = *(mp++);
286 }
287 PxVec3 matOff = c->getMaterialOffset(); 318 285 0.90 25.61 392.98 3.66 3.16 1.05
288 ta[3] = matOff.x; 372 307 0.83 26.06 439.74 3.93 2.61 0.65
289 ta[7] = matOff.y; 14 12 0.86 41.67 500.00 1.67 8.33
290 ta[11] = matOff.z; 3 6 2.00 33.33 333.33
291
292 int idFor2DTex = c->getSurfaceMaterialId();
293 int idFor3DTex = c->getMaterialId();
294 const int MAX_3D_TEX = 8;
295 ta[15] = (float)(idFor2DTex*MAX_3D_TEX + idFor 678 554 0.82 24.55 0.74 429.60 4.03 1.62 0.36
296
297
298 }
299
300 glBindTexture(GL_TEXTURE_2D, g->matTex); 12 9 0.75 44.44 222.22 15.00
301 glTexSubImage2D(GL_TEXTURE_2D, 0, 0, 0, g->texSize, g->te
302 GL_RGBA, GL_FLOAT, &g->texCoords[0]);
303 glBindTexture(GL_TEXTURE_2D, 0); 4
304 }
305 }
306
307 //----------
308 void ConvexRenderer::render()
Before Optimization.



Line	Offset	Source Code	CPU clocks	Ret inst	IPC	Retired Branch I	%Retired Branch	Data Cache Acce	%Data Cache Mis	Data Cache Refill	Data Cache Refill	Data Cache Refill	Data Cache Refill	Data Cac
264														
265		}												
266														
267		//----------												
268		void ConvexRenderer::updateTransformations()												
269		{	2											
270		for (int i = 0; i < (int)mGroups.size(); i++) {	12	3	0.25			666.67	10.00					
271		ConvexGroup *g = mGroups[i];												
272		if (g->texCoords.empty())												
273		continue;												
274														
275		float* tt = &g->texCoords[0];												
276														
277		for (int j = 0; j < (int)g->convexes.size(); j++) {	9059	8636	0.95	24.68	0.19	436.31	3.98			2.63	0.57	
278		const Convex* c = g->convexes[j];	2	1	0.50									
279														
280		PxMat44 pose(c->getGlobalPose());	8089	6873	0.85	27.83	0.26	419.03	4.02			2.21	0.71	
281		float* mp = (float*)pose.front();												
282														
283		float* ta = tt;	4	2	0.50	150.00								
284		for (int k = 0; k < 16; k++) {												
285		*(tt++) = *(mp++);												
286		}												
287		PxVec3 matOff = c->getMaterialOffset();	386	374	0.97	29.95		393.05	4.15			1.07	1.34	
288		ta[3] = matOff.x;	2	4	2.00	150.00		250.00	10.00					
289		ta[7] = matOff.y;	3	2	0.67	250.00		1000.00	5.00				50.00	
290		ta[11] = matOff.z;	83	76	0.92	23.68		302.63	5.22					
291														
292		int idFor2DTex = c->getSurfaceMaterialId();												
293		int idFor3DTex = c->getMaterialId();												
294		const int MAX_3D_TEX = 8;												
295		ta[15] = (float)(idFor2DTex*MAX_3D_TEX + idFor	536	555	1.04	27.03	0.67	400.00	3.69			1.62	0.36	
296														
297														
298		}												
299														
300		glBindTexture(GL_TEXTURE_2D, g->matTex);	1											
301		glTexSubImage2D(GL_TEXTURE_2D, 0, 0, 0, g->texSize, g->te	1											
302		GL_RGBA, GL_FLOAT, &g->texCoords[0]);												
303		glBindTexture(GL_TEXTURE_2D, 0);	5											
304		}												
305		}												
306														
307		//----------												
308		void ConvexRenderer::render()												

BUT CAN WE DO BETTER?

YES!

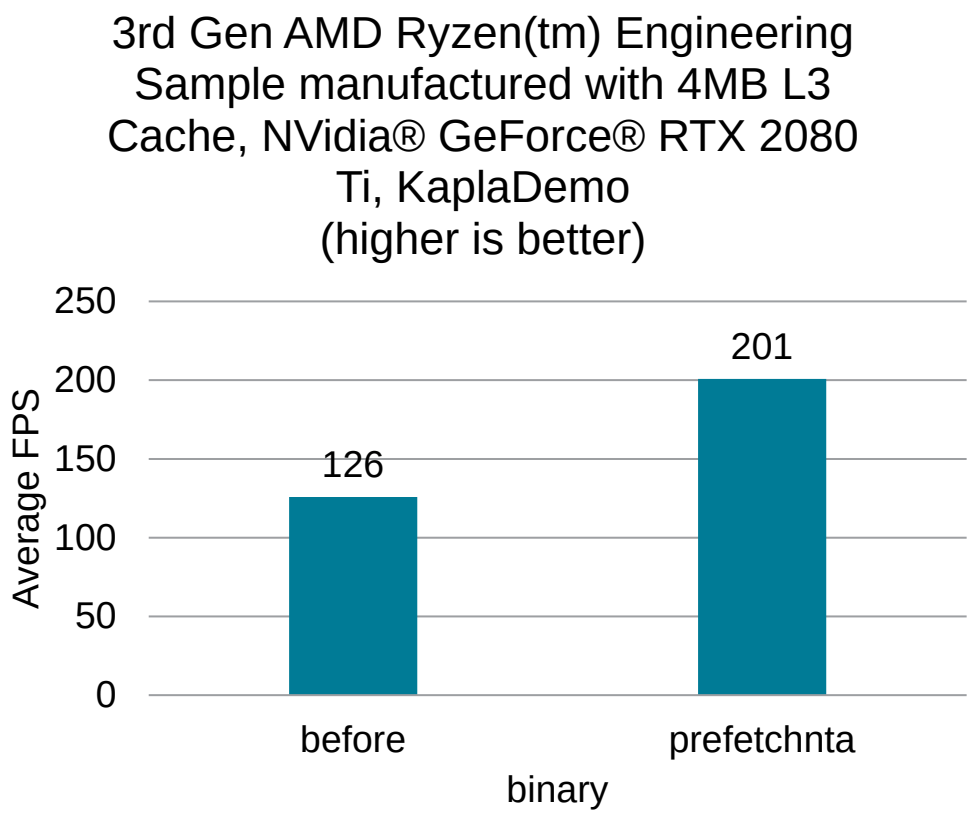
USE PREFETCH LEVEL WHILE ITERATING STD::VECTOR<T*>

SUMMARY

- Use AMDuProf to find hot functions where there are many Data Cache refills from DRAM.
- If many refills from DRAM are while iterating std::vector<T*>, try using the _mm_prefetch intrinsic to improve performance.
- The distance to prefetch and the prefetch level <NTA|T0> may require some tuning.
- Try prefetch distance 4 and prefetch level NTA.
- For public & friend private data, try using _mm_prefetch((char*)(&v[future]->data), _MM_HINT_NTA);
- For private data, use WinDbg "dt <T>" to determine the offset of the hot data then try using _mm_prefetch(offset + (char*)(v[future]), _MM_HINT_NTA);

```
// Example
struct S { double x, y, z, w; char name[256]; double s,
t, u, v;};

int main() {
  std::vector<S*> v;
  // initialize v
  for (size_t i = 0; i < v.size(); i++) {
    size_t distance = 4; // TODO find ideal number
    size_t future = (i + distance) % v.size();
    _mm_prefetch((char*)&(v[future]->x), _MM_HINT_NTA);
    _mm_prefetch((char*)&(v[future]->s), _MM_HINT_NTA);
    foo(v[i]->x);
    bar(v[i]->s);
  }
}
```

PERFORMANCE



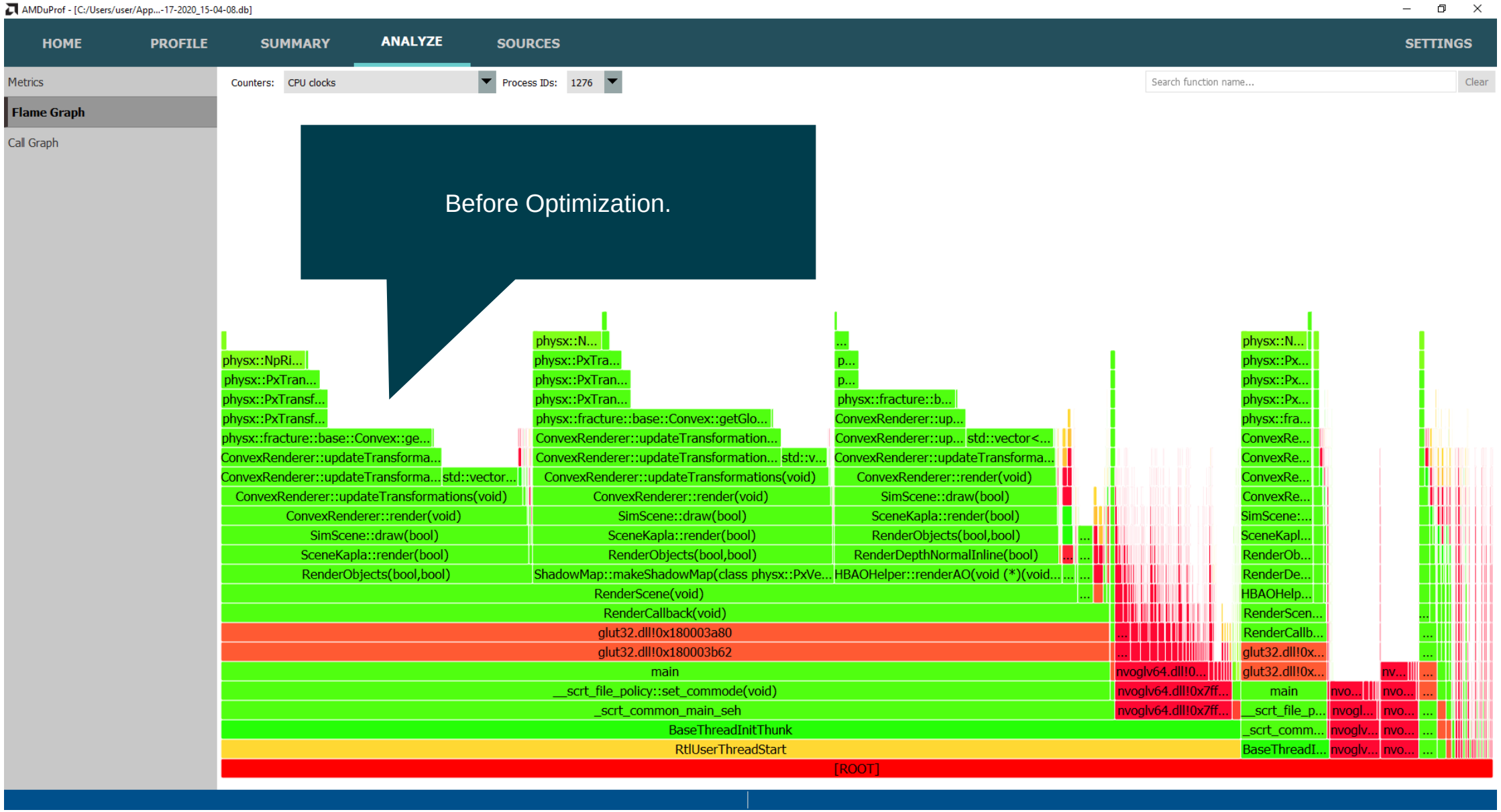
- 60% faster after optimization!
- Performance of binaries compiled with Microsoft Visual Studio 2019 v16.4.5.
- Testing done by AMD technology labs, February 15, 2020 on the following system. Test configuration: 3rd Gen AMD Ryzen™ Engineering Sample manufactured with 4MB L3 Cache, AMD Wraith Prism Cooler, 16GB (2 x 8GB DDR4-3200 at 22-22-22-52) memory, NVidia® GeForce® RTX 2080 Ti GPU with driver 441.87 (December 24, 2019), 2TB M.2 NVME SSD, AMD Ryzen™ Reference Motherboard, Windows® 10 x64 build 1909, 1920x1080 resolution. Actual results may vary.

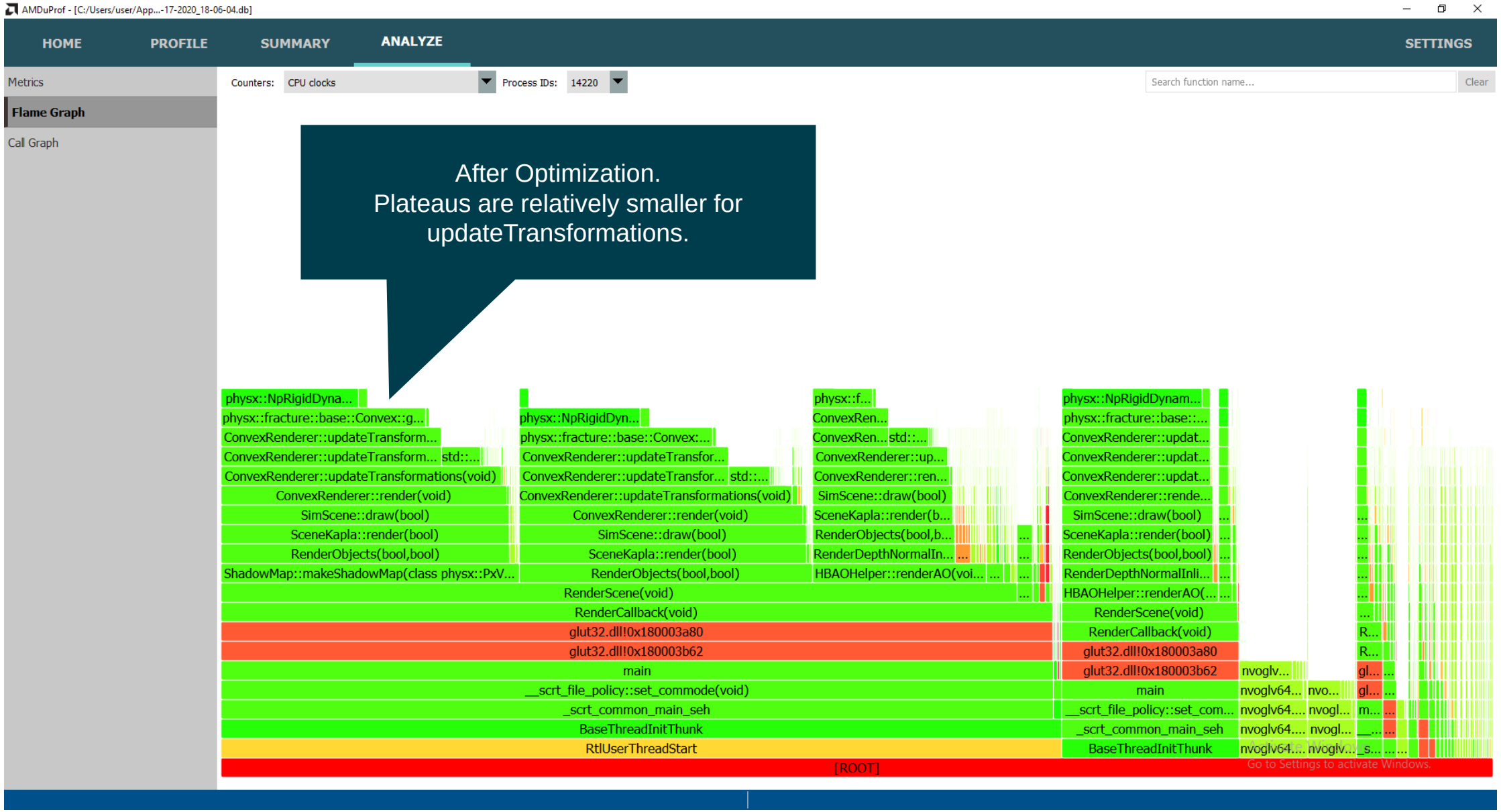
CODE SAMPLE

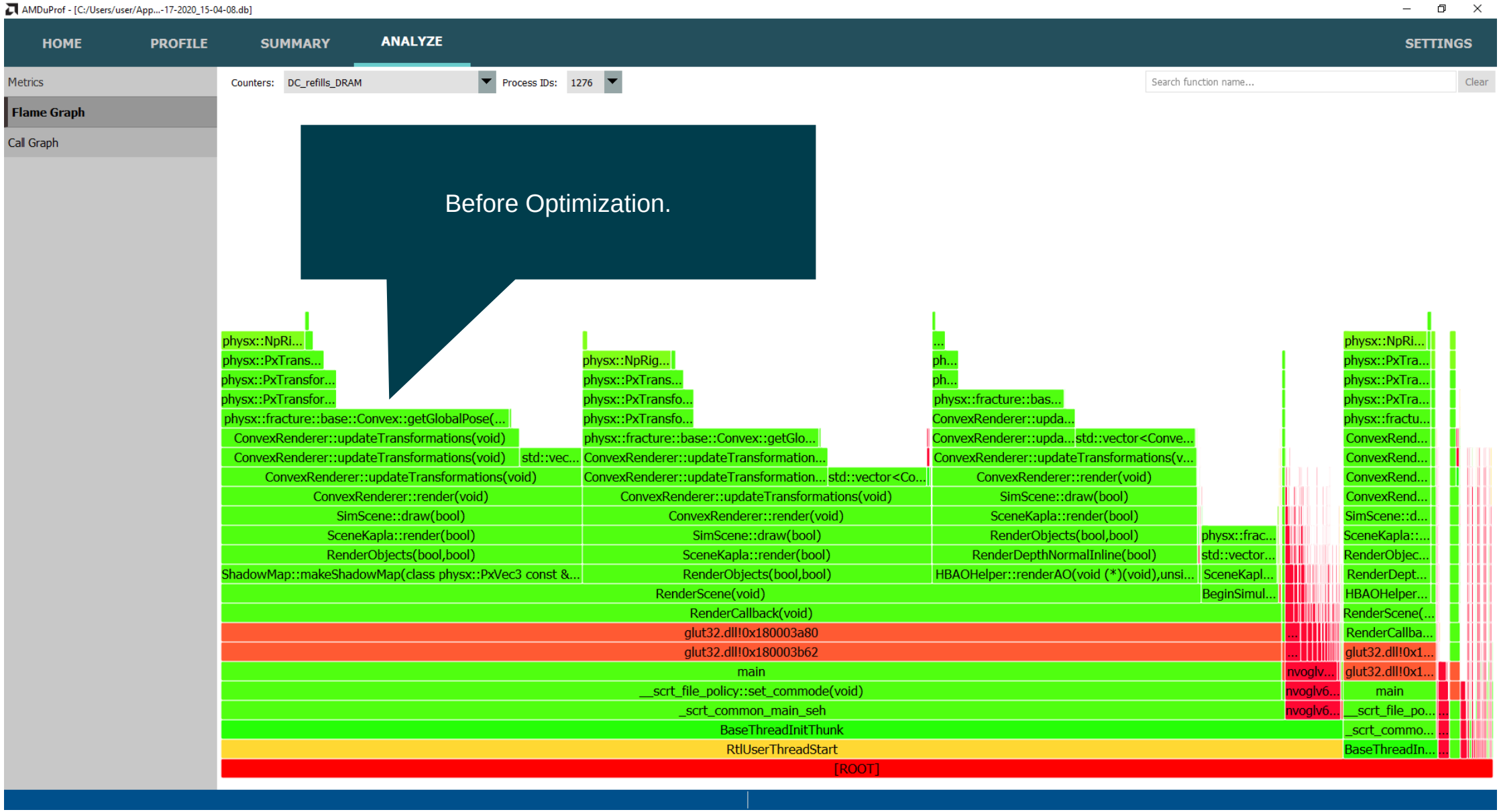
https://github.com/NVIDIAGameWorks/PhysX/tree/4.1/physx
ConvexRenderer.cpp

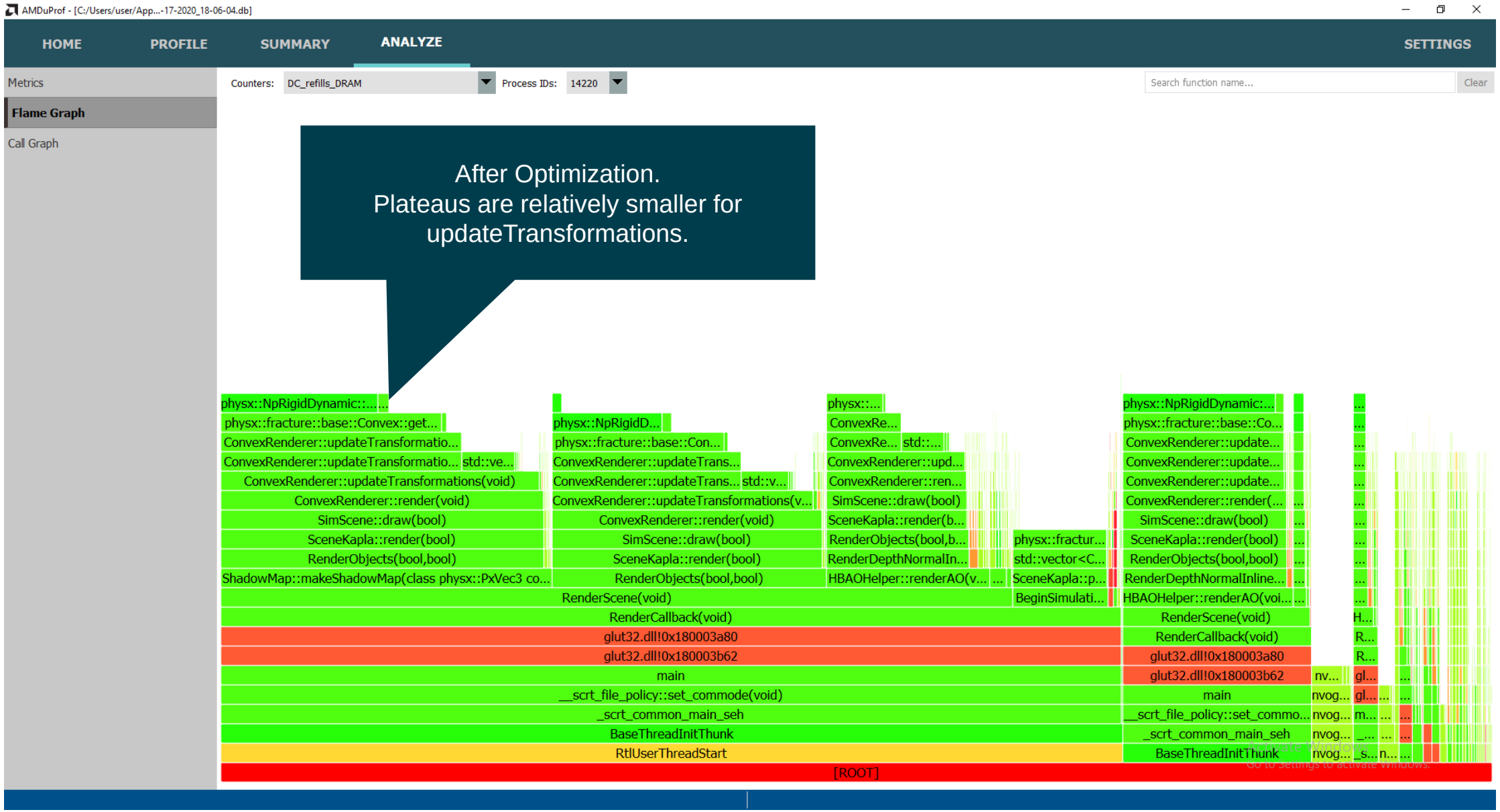
```
void ConvexRenderer::updateTransformations() {
 for (int i = 0; i < (int)mGroups.size(); i++) {
  ConvexGroup *g = mGroups[i];
  if (g->texCoords.empty())
   continue;
  float* tt = &g->texCoords[0];
  for (int j = 0; j < (int)g->convexes.size(); j++) {
   const Convex* c = g->convexes[j];
#if 1
   int distance = 4; // TODO find ideal number
   size_t future = (j + distance) % g->convexes.size();
   _mm_prefetch(0x0F8 + (char*)(g->convexes[future]), _MM_HINT_NTA); // mPxActor
   _mm_prefetch(0x100 + (char*)(g->convexes[future]), _MM_HINT_NTA); // mLocalPose
   _mm_prefetch(0x148 + (char*)(g->convexes[future]), _MM_HINT_NTA); // mMaterialOffset.x
   _mm_prefetch(0x14C + (char*)(g->convexes[future]), _MM_HINT_NTA); // mMaterialOffset.y
   _mm_prefetch(0x150 + (char*)(g->convexes[future]), _MM_HINT_NTA); // mMaterialOffset.z
   _mm_prefetch(0x164 + (char*)(g->convexes[future]), _MM_HINT_NTA); //mSurfaceMaterialId
   _mm_prefetch(0x160 + (char*)(g->convexes[future]), _MM_HINT_NTA); // mMaterialId
#endif
 PxMat44 pose(c->getGlobalPose());
  float* mp = (float*)pose.front();
  float* ta = tt;
  for (int k = 0; k < 16; k++) {
   *(tt++) = *(mp++);
  }
  PxVec3 matOff = c->getMaterialOffset();
  ta[3] = matOff.x;
  ta[7] = matOff.y;
  ta[11] = matOff.z;
  int idFor2DTex = c->getSurfaceMaterialId();
  int idFor3DTex = c->getMaterialId();
  const int MAX_3D_TEX = 8;
  ta[15] = (float)(idFor2DTex*MAX_3D_TEX + idFor3DTex);
 }
 glBindTexture(GL_TEXTURE_2D, g->matTex);
 glTexSubImage2D(GL_TEXTURE_2D, 0, 0, 0, g->texSize, g->texSize, GL_RGBA,
GL_FLOAT, &g->texCoords[0]);
 glBindTexture(GL_TEXTURE_2D, 0);
 }
}
```

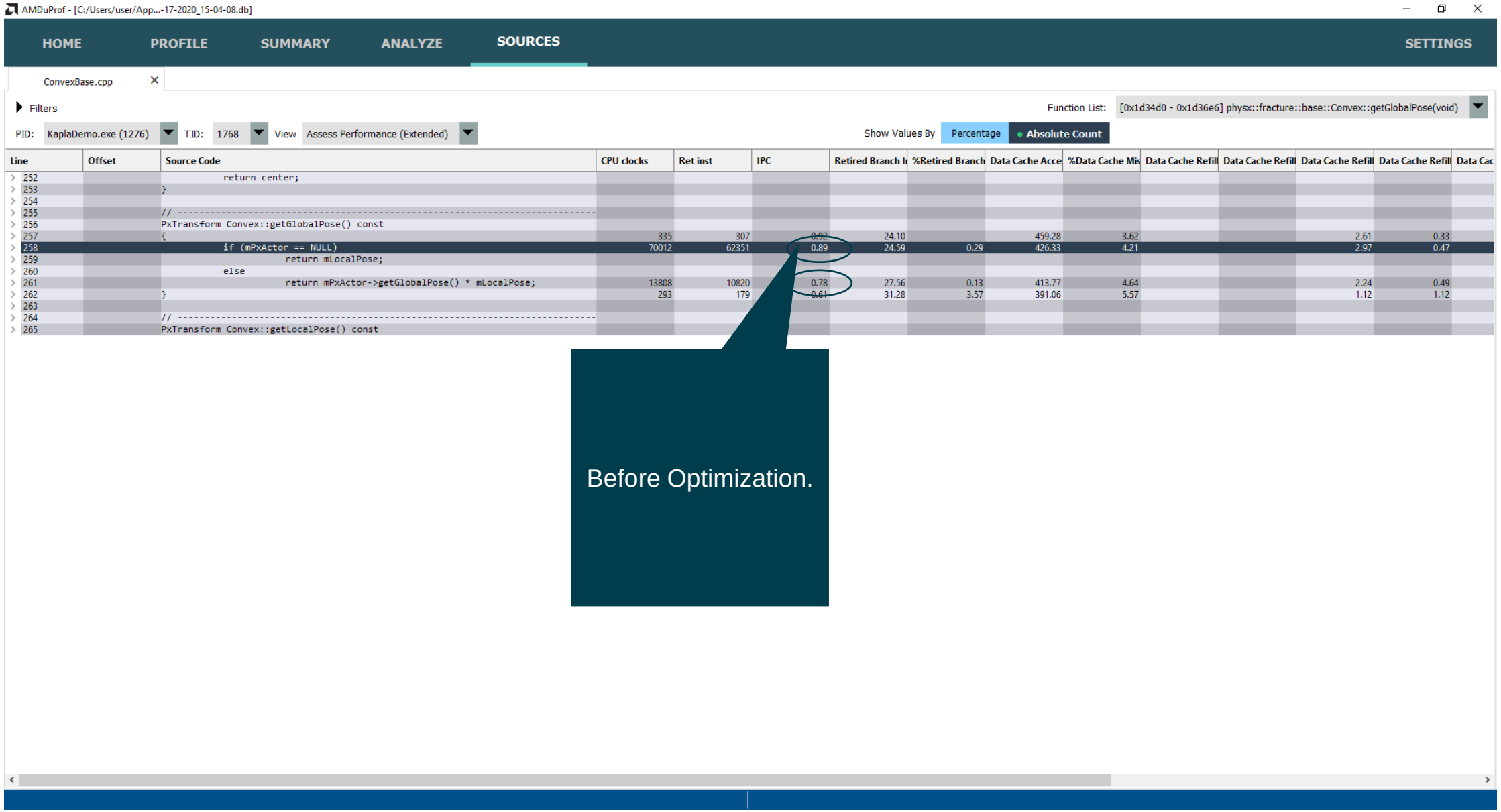
Used offsets due to protected data

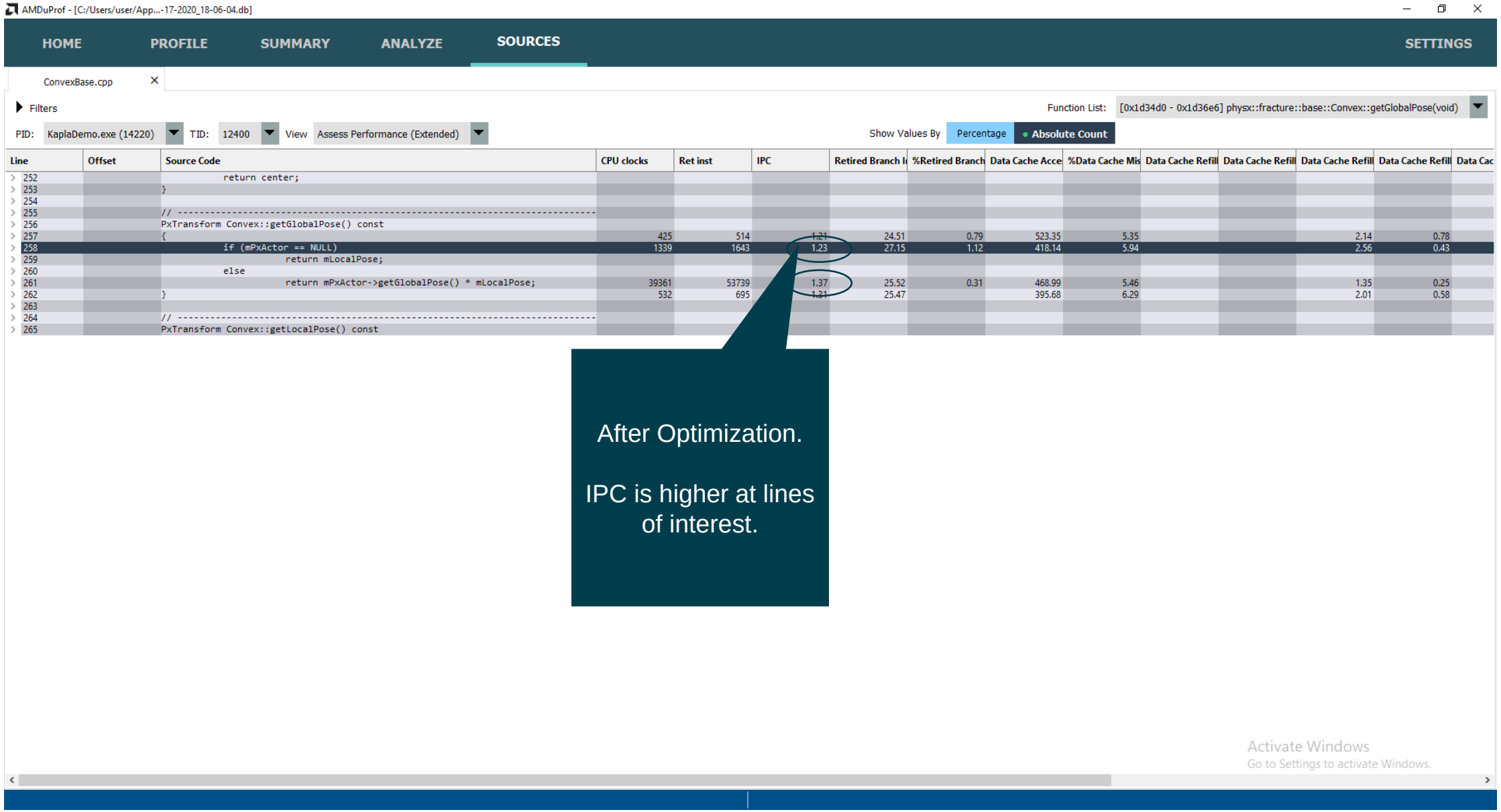

AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
Metrics
Flame Graph
Call Graph
Counters:
CPU clocks
Process IDs:
1276
Search function name...
Clear
Before Optimization.
physx::NpRi...
physx::PxTran...
physx::PxTransf...
physx::PxTransf...
physx::fracture::base::Convex::ge...
ConvexRenderer::updateTransforma...
ConvexRenderer::updateTransforma...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commode(void)
_scrt_common_main_seh
BaseThreadInitThunk
RtlUserThreadStart
[ROOT]
physx::N...
physx::PxTra...
physx::PxTran...
physx::PxTran...
physx::fracture::base::Convex::getGlo...
ConvexRenderer::updateTransformation...
ConvexRenderer::updateTransformation...
std::v...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
ShadowMap::makeShadowMap(class physx::PxVe...
physx::N...
physx::Px...
physx::Px...
physx::Px...
physx::fra...
ConvexRe...
ConvexRe...
ConvexRe...
ConvexRe...
SimScene:...
SceneKapl...
RenderOb...
RenderDe...
HBAOHelp...
RenderScen...
RenderCallb...
glut32.dll!0x...
glut32.dll!0x...
main
_scrt_file_p...
_scrt_comm...
BaseThreadI...
p...
p...
physx::fracture::b...
ConvexRenderer::up...
ConvexRenderer::up...
std::vector<...
ConvexRenderer::updateTransforma...
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderDepthNormalInline(bool)
HBAOHelper::renderAO(void (*)(void...
nvoglv64.dll!0...
nvoglv64.dll!0x7ff...
nvoglv64.dll!0x7ff...
nvo...
nvogl...
nvoglv...
nv...
nvo...
nvo...
nvo...
nvo...

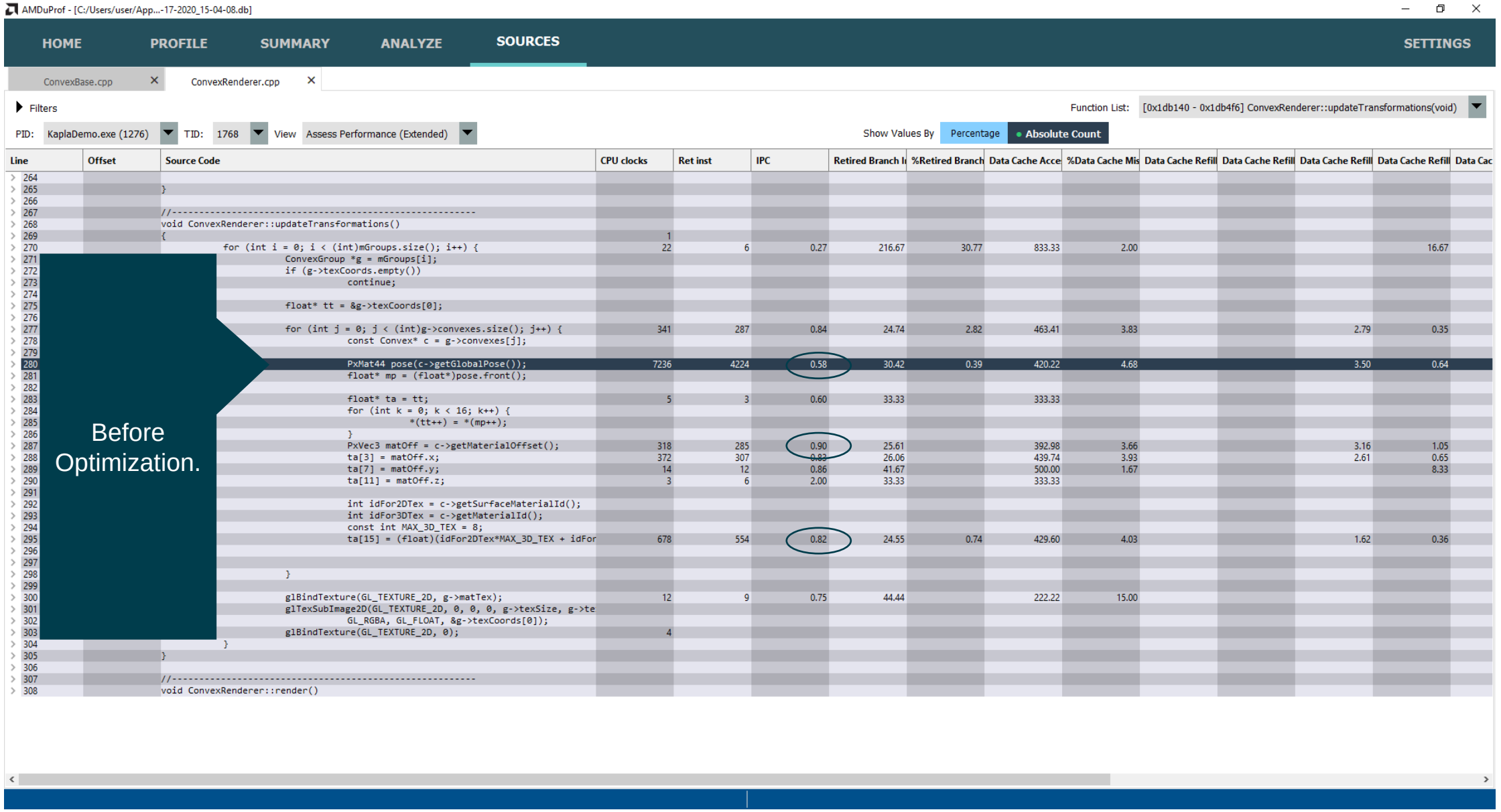

AMDuProf - [C:/Users/user/App...-17-2020_18-06-04.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Metrics
Flame Graph
Call Graph
Counters:
CPU clocks
Process IDs:
14220
Search function name...
Clear
After Optimization.
Plateaus are relatively smaller for updateTransformations.
physx::NpRigidDyna...
physx::fracture::base::Convex::g...
ConvexRenderer::updateTransform...
ConvexRenderer::updateTransform... std::...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
ShadowMap::makeShadowMap(class physx::PxV...
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commode(void)
_scrt_common_main_seh
BaseThreadInitThunk
RtlUserThreadStart
[ROOT]
physx::NpRigidDyn...
physx::fracture::base::Convex:...
ConvexRenderer::updateTransfor...
ConvexRenderer::updateTransfor... std::...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
HBAOHelper::renderAO(voi...
physx::f...
ConvexRen...
ConvexRen... std::...
ConvexRenderer::up...
ConvexRenderer::ren...
SimScene::draw(bool)
SceneKapla::render(b...
RenderObjects(bool,b...
RenderDepthNormalIn...
HBAOHelper::renderAO(...
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_com...
_scrt_common_main_seh
BaseThreadInitThunk
physx::NpRigidDynam...
physx::fracture::base::...
ConvexRenderer::updat...
ConvexRenderer::updat...
ConvexRenderer::updat...
ConvexRenderer::rende...
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderDepthNormalInli...
HBAOHelper::renderAO(...
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
nvoglv...
nvoglv64...
nvo...
nvoglv64....
nvogl...
nvoglv64....
nvogl...
nvoglv64....
nvoglv...
Go to Settings to activate Windows.

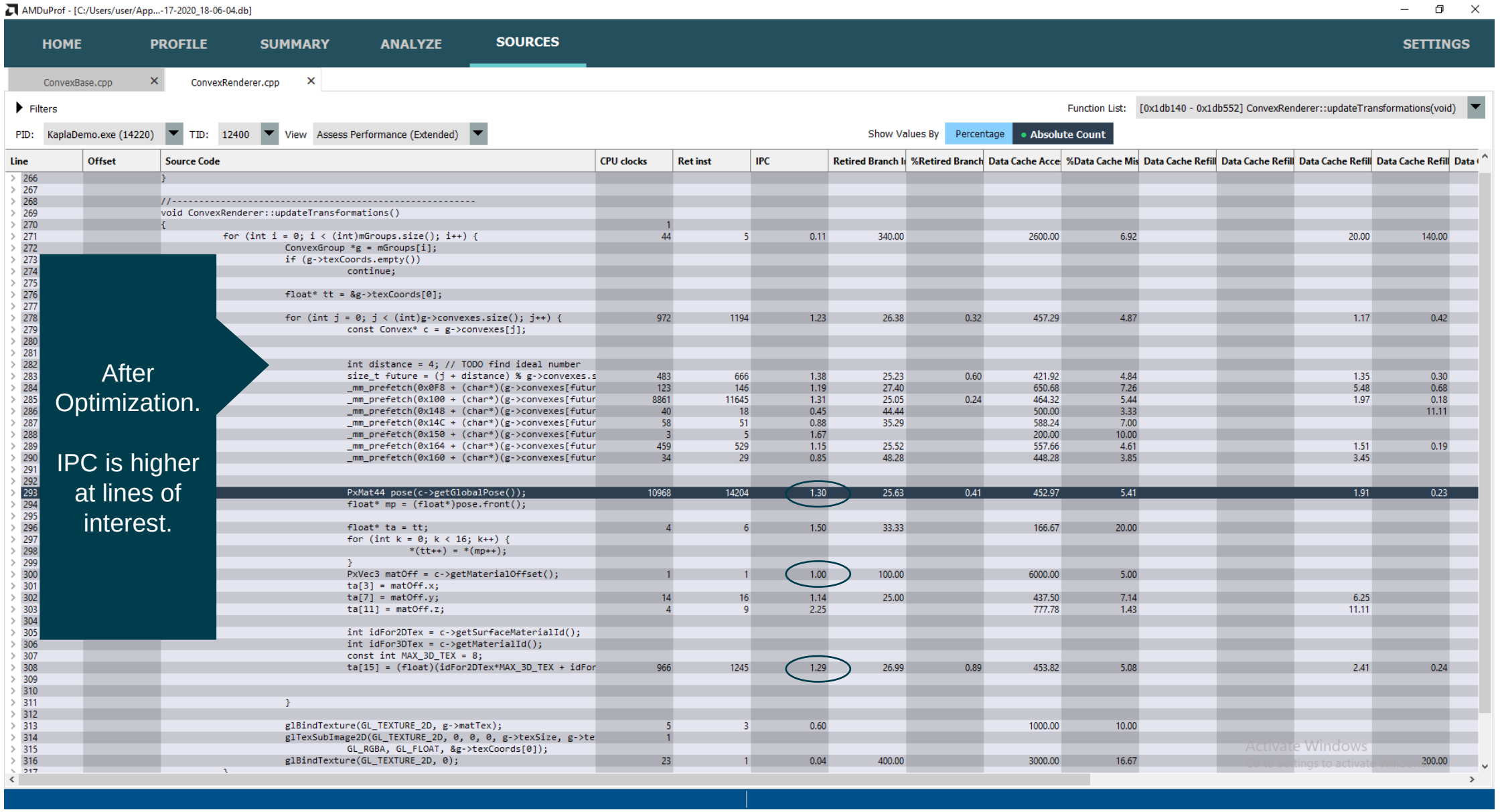

AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Metrics
Flame Graph
Call Graph
Counters:
DC_refills_DRAM
Process IDs:
1276
Search function name...
Clear
Before Optimization.
physx::NpRi...
physx::PxTrans...
physx::PxTransfor...
physx::PxTransfor...
physx::fracture::base::Convex::getGlobalPose(...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::updateTransformations(void)
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
ShadowMap::makeShadowMap(class physx::PxVec3 const &...
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commode(void)
_scrt_common_main_seh
BaseThreadInitThunk
RtlUserThreadStart
[ROOT]
physx::NpRig...
physx::PxTrans...
physx::PxTransfo...
physx::PxTransfo...
physx::fracture::base::Convex::getGlo...
ConvexRenderer::updateTransformation...
ConvexRenderer::updateTransformation...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderDepthNormalInline(bool)
HBAOHelper::renderAO(void (*)(void),unsi...
ph...
ph...
physx::fracture::bas...
ConvexRenderer::upda...
ConvexRenderer::upda...
ConvexRenderer::updateTransformations(v...
std::vector<Conve...
physx::frac...
std::vector...
SceneKapl...
BeginSimul...
nvoglv...
nvoglv6...
nvoglv6...
physx::NpRi...
physx::PxTra...
physx::PxTra...
physx::PxTra...
physx::fractu...
ConvexRend...
ConvexRend...
ConvexRend...
ConvexRend...
SimScene::d...
SceneKapla::...
RenderObjec...
RenderDept...
HBAOHelper...
RenderScene(...
RenderCallba...
glut32.dll!0x1...
glut32.dll!0x1...
main
__scrt_file_po...
_scrt_commo...
BaseThreadIn...


AMDuProf - [C:/Users/user/App...-17-2020_18-06-04.db]
HOME
PROFILE
SUMMARY
ANALYZE
SETTINGS
Metrics
Flame Graph
Call Graph
Counters:
DC_refills_DRAM
Process IDs:
14220
Search function name...
Clear
After Optimization.
Plateaus are relatively smaller for updateTransformations.
physx::NpRigidDynamic::...
physx::fracture::base::Convex::get...
ConvexRenderer::updateTransformatio...
ConvexRenderer::updateTransformatio...
std::ve...
ConvexRenderer::updateTransformations(void)
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
ShadowMap::makeShadowMap(class physx::PxVec3 co...
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commode(void)
_scrt_common_main_seh
BaseThreadInitThunk
RtlUserThreadStart
[ROOT]
physx::NpRigidD...
physx::fracture::base::Con...
ConvexRenderer::updateTrans...
ConvexRenderer::updateTrans...
std::v...
ConvexRenderer::updateTransformations(v...
ConvexRenderer::render(void)
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
HBAOHelper::renderAO(v...
physx::...
ConvexRe...
ConvexRe...
std::...
ConvexRenderer::upd...
ConvexRenderer::ren...
SimScene::draw(bool)
SceneKapla::render(b...
RenderObjects(bool,b...
RenderDepthNormalIn...
SceneKapla::p...
BeginSimulati...
physx::fractur...
std::vector<C...
physx::NpRigidDynamic:...
physx::fracture::base::Co...
ConvexRenderer::update...
ConvexRenderer::update...
ConvexRenderer::update...
ConvexRenderer::render(...
SimScene::draw(bool)
SceneKapla::render(bool)
RenderObjects(bool,bool)
RenderDepthNormalInline...
HBAOHelper::renderAO(voi...
RenderScene(void)
RenderCallback(void)
glut32.dll!0x180003a80
glut32.dll!0x180003b62
main
__scrt_file_policy::set_commo...
_scrt_common_main_seh
BaseThreadInitThunk
nv...
nvog...
nvog...
nvog...
nvog...
H...
R...
R...
gl...
gl...
m...
_...
_s...
n...

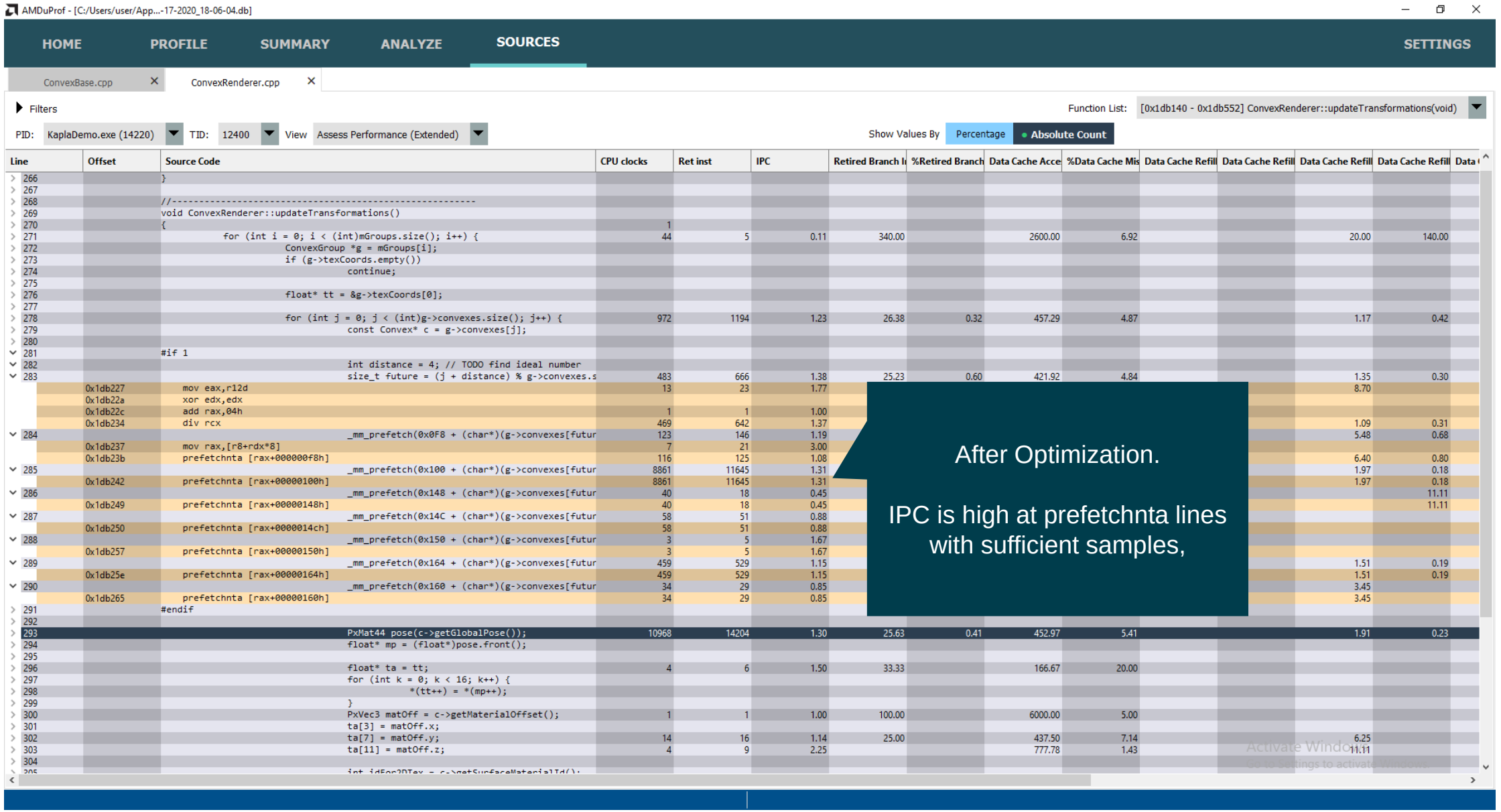

AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
ConvexBase.cpp
Filters
Function List: [0x1d34d0 - 0x1d36e6] physx::fracture::base::Convex::getGlobalPose(void)
PID: KaplaDemo.exe (1276)
TID: 1768
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line Offset Source Code CPU clocks Ret inst IPC Retired Branch I %Retired Branch Data Cache Acce %Data Cache Mis Data Cache Refill Data Cache Refill Data Cache Refill Data Cache Refill Data Cac
252 return center;
253 }
254
255 // ----------
256 PxTransform Convex::getGlobalPose() const
257 { 335 307 0.92 24.10 459.28 3.62 2.61 0.33
258 if (mPxActor == NULL) 70012 62351 0.89 24.59 0.29 426.33 4.21 2.97 0.47
259 return mLocalPose;
260 else
261 return mPxActor->getGlobalPose() * mLocalPose; 13808 10820 0.78 27.56 0.13 413.77 4.64 2.24 0.49
262 } 293 179 0.61 31.28 3.57 391.06 5.57 1.12 1.12
263
264 // ----------
265 PxTransform Convex::getLocalPose() const
Before Optimization.


AMDuProf - [C:/Users/user/App...-17-2020_18-06-04.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
ConvexBase.cpp
Filters
Function List: [0x1d34d0 - 0x1d36e6] physx::fracture::base::Convex::getGlobalPose(void)
PID: KaplaDemo.exe (14220)
TID: 12400
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line Offset Source Code CPU clocks Ret inst IPC Retired Branch I %Retired Branch Data Cache Acce %Data Cache Mis Data Cache Refill Data Cache Refill Data Cache Refill Data Cache Refill Data Cac
252 return center;
253 }
254
255 // ----------
256 PxTransform Convex::getGlobalPose() const
257 { 425 514 1.21 24.51 0.79 523.35 5.35 2.14 0.78
258 if (mPxActor == NULL) 1339 1643 1.23 27.15 1.12 418.14 5.94 2.56 0.43
259 return mLocalPose;
260 else
261 return mPxActor->getGlobalPose() * mLocalPose; 39361 53739 1.37 25.52 0.31 468.99 5.46 1.35 0.25
262 } 532 695 1.21 25.47 395.68 6.29 2.01 0.58
263
264 // ----------
265 PxTransform Convex::getLocalPose() const
After Optimization.
IPC is higher at lines of interest.
Activate Windows
Go to Settings to activate Windows.


AMDuProf - [C:/Users/user/App...-17-2020_15-04-08.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
ConvexBase.cpp
ConvexRenderer.cpp
Filters
Function List: [0x1db140 - 0x1db4f6] ConvexRenderer::updateTransformations(void)
PID: KaplaDemo.exe (1276)
TID: 1768
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line Offset Source Code CPU clocks Ret inst IPC Retired Branch I %Retired Branch Data Cache Acce %Data Cache Mis Data Cache Refill Data Cache Refill Data Cache Refill Data Cache Refill Data Cac
264
265 }
266
267 //----------
268 void ConvexRenderer::updateTransformations()
269 { 1
270 for (int i = 0; i < (int)mGroups.size(); i++) { 22 6 0.27 216.67 30.77 833.33 2.00 16.67
271 ConvexGroup *g = mGroups[i];
272 if (g->texCoords.empty())
273 continue;
274
275 float* tt = &g->texCoords[0];
276
277 for (int j = 0; j < (int)g->convexes.size(); j++) { 341 287 0.84 24.74 2.82 463.41 3.83 2.79 0.35
278 const Convex* c = g->convexes[j];
279
280 PxMat44 pose(c->getGlobalPose()); 7236 4224 0.58 30.42 0.39 420.22 4.68 3.50 0.64
281 float* mp = (float*)pose.front();
282
283 float* ta = tt; 5 3 0.60 33.33 333.33
284 for (int k = 0; k < 16; k++) {
285 *(tt++) = *(mp++);
286 }
287 PxVec3 matOff = c->getMaterialOffset(); 318 285 0.90 25.61 392.98 3.66 3.16 1.05
288 ta[3] = matOff.x; 372 307 0.83 26.06 439.74 3.93 2.61 0.65
289 ta[7] = matOff.y; 14 12 0.86 41.67 500.00 1.67 8.33
290 ta[11] = matOff.z; 3 6 2.00 33.33 333.33
291
292 int idFor2DTex = c->getSurfaceMaterialId();
293 int idFor3DTex = c->getMaterialId();
294 const int MAX_3D_TEX = 8;
295 ta[15] = (float)(idFor2DTex*MAX_3D_TEX + idFor 678 554 0.82 24.55 0.74 429.60 4.03 1.62 0.36
296
297
298 }
299
300 glBindTexture(GL_TEXTURE_2D, g->matTex); 12 9 0.75 44.44 222.22 15.00
301 glTexSubImage2D(GL_TEXTURE_2D, 0, 0, 0, g->texSize, g->te
302 GL_RGBA, GL_FLOAT, &g->texCoords[0]);
303 glBindTexture(GL_TEXTURE_2D, 0); 4
304 }
305 }
306
307 //----------
308 void ConvexRenderer::render()
Before Optimization.



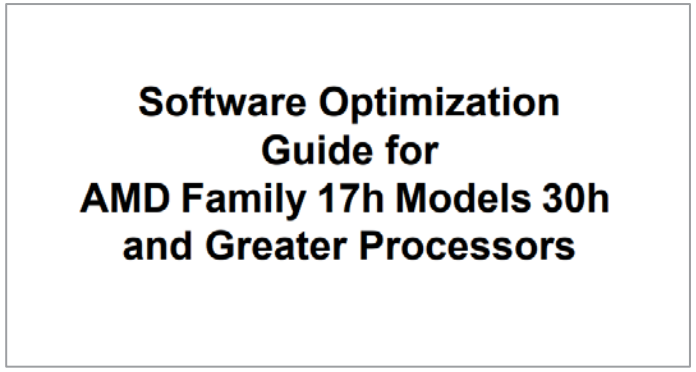
Line	Offset	Source Code	CPU clocks	Ret inst	IPC	Retired Branch I	%Retired Branch	Data Cache Acce	%Data Cache Mis	Data Cache Refill	Data Cache Refill	Data Cache Refill	Data Cache Refill	Data
266		}												
267														
268		//----------												
269		void ConvexRenderer::updateTransformations()												
270		{	1											
271		for (int i = 0; i < (int)mGroups.size(); i++) {	44	5	0.11	340.00		2600.00	6.92			20.00	140.00	
272		ConvexGroup *g = mGroups[i];												
273		if (g->texCoords.empty())												
274		continue;												
275														
276		float* tt = &g->texCoords[0];												
277														
278		for (int j = 0; j < (int)g->convexes.size(); j++) {	972	1194	1.23	26.38	0.32	457.29	4.87			1.17	0.42	
279		const Convex* c = g->convexes[j];												
280														
281														
282		int distance = 4; // TODO find ideal number												
283		size_t future = (j + distance) % g->convexes.s	483	666	1.38	25.23	0.60	421.92	4.84			1.35	0.30	
284		_mm_prefetch(0x0F8 + (char*)(g->convexes[futur	123	146	1.19	27.40		650.68	7.26			5.48	0.68	
285		_mm_prefetch(0x100 + (char*)(g->convexes[futur	8861	11645	1.31	25.05	0.24	464.32	5.44			1.97	0.18	
286		_mm_prefetch(0x148 + (char*)(g->convexes[futur	40	18	0.45	44.44		500.00	3.33				11.11	
287		_mm_prefetch(0x14C + (char*)(g->convexes[futur	58	51	0.88	35.29		588.24	7.00					
288		_mm_prefetch(0x150 + (char*)(g->convexes[futur	3	5	1.67			200.00	10.00					
289		_mm_prefetch(0x164 + (char*)(g->convexes[futur	459	529	1.15	25.52		557.66	4.61			1.51	0.19	
290		_mm_prefetch(0x160 + (char*)(g->convexes[futur	34	29	0.85	48.28		448.28	3.85			3.45		
291														
292														
293		PxMat44 pose(c->getGlobalPose());	10968	14204	1.30	25.63	0.41	452.97	5.41			1.91	0.23	
294		float* mp = (float*)pose.front();												
295														
296		float* ta = tt;	4	6	1.50	33.33		166.67	20.00					
297		for (int k = 0; k < 16; k++) {												
298		*(tt++) = *(mp++);												
299		}												
300		PxVec3 matOff = c->getMaterialOffset();	1	1	1.00	100.00		6000.00	5.00					
301		ta[3] = matOff.x;												
302		ta[7] = matOff.y;	14	16	1.14	25.00		437.50	7.14			6.25		
303		ta[11] = matOff.z;	4	9	2.25			777.78	1.43			11.11		
304														
305		int idFor2DTex = c->getSurfaceMaterialId();												
306		int idFor3DTex = c->getMaterialId();												
307		const int MAX_3D_TEX = 8;												
308		ta[15] = (float)(idFor2DTex*MAX_3D_TEX + idFor	966	1245	1.29	26.99	0.89	453.82	5.08			2.41	0.24	
309														
310														
311		}												
312														
313		glBindTexture(GL_TEXTURE_2D, g->matTex);	5	3	0.60			1000.00	10.00					
314		glTexSubImage2D(GL_TEXTURE_2D, 0, 0, 0, g->texSize, g->te	1											
315		GL_RGBA, GL_FLOAT, &g->texCoords[0]);												
316		glBindTexture(GL_TEXTURE_2D, 0);	23	1	0.04	400.00		3000.00	16.67				200.00	


AMDuProf - [C:/Users/user/App...-17-2020_18-06-04.db]
HOME
PROFILE
SUMMARY
ANALYZE
SOURCES
SETTINGS
ConvexBase.cpp
ConvexRenderer.cpp
Filters
Function List: [0x1db140 - 0x1db552] ConvexRenderer::updateTransformations(void)
PID: KaplaDemo.exe (14220)
TID: 12400
View Assess Performance (Extended)
Show Values By Percentage Absolute Count
Line Offset Source Code CPU clocks Ret inst IPC Retired Branch I %Retired Branch Data Cache Acce %Data Cache Mis Data Cache Refill Data Cache Refill Data Cache Refill Data Cache Refill Data
266 }
267
268 //----------
269 void ConvexRenderer::updateTransformations()
270 { 1
271 for (int i = 0; i < (int)mGroups.size(); i++) { 44 5 0.11 340.00 2600.00 6.92 20.00 140.00
272 ConvexGroup *g = mGroups[i];
273 if (g->texCoords.empty())
274 continue;
275
276 float* tt = &g->texCoords[0];
277
278 for (int j = 0; j < (int)g->convexes.size(); j++) { 972 1194 1.23 26.38 0.32 457.29 4.87 1.17 0.42
279 const Convex* c = g->convexes[j];
280
281 #if 1
282 int distance = 4; // TODO find ideal number
283 size_t future = (j + distance) % g->convexes.s 483 666 1.38 25.23 0.60 421.92 4.84 1.35 0.30
0x1db227 mov eax,r12d 13 23 1.77 8.70
0x1db22a xor edx,edx
0x1db22c add rax,04h 1 1 1.00
0x1db234 div rcx 469 642 1.37 1.09 0.31
284 _mm_prefetch(0x0F8 + (char*)(g->convexes[futur 123 146 1.19 5.48 0.68
0x1db237 mov rax,[r8+rdx*8] 7 21 3.00
0x1db23b prefetchnta [rax+000000f8h] 116 125 1.08 6.40 0.80
285 _mm_prefetch(0x100 + (char*)(g->convexes[futur 8861 11645 1.31 1.97 0.18
0x1db242 prefetchnta [rax+00000100h] 8861 11645 1.31 1.97 0.18
286 _mm_prefetch(0x148 + (char*)(g->convexes[futur 40 18 0.45 11.11
0x1db249 prefetchnta [rax+00000148h] 40 18 0.45 11.11
287 _mm_prefetch(0x14C + (char*)(g->convexes[futur 58 51 0.88
0x1db250 prefetchnta [rax+0000014ch] 58 51 0.88
288 _mm_prefetch(0x150 + (char*)(g->convexes[futur 3 5 1.67
0x1db257 prefetchnta [rax+00000150h] 3 5 1.67
289 _mm_prefetch(0x164 + (char*)(g->convexes[futur 459 529 1.15 1.51 0.19
0x1db25e prefetchnta [rax+00000164h] 459 529 1.15 1.51 0.19
290 _mm_prefetch(0x160 + (char*)(g->convexes[futur 34 29 0.85 3.45
0x1db265 prefetchnta [rax+00000160h] 34 29 0.85 3.45
291 #endif
292
293 PxMat44 pose(c->getGlobalPose()); 10968 14204 1.30 25.63 0.41 452.97 5.41 1.91 0.23
294 float* mp = (float*)pose.front();
295
296 float* ta = tt; 4 6 1.50 33.33 166.67 20.00
297 for (int k = 0; k < 16; k++) {
298 *(tt++) = *(mp++);
299 }
300 PxVec3 matOff = c->getMaterialOffset(); 1 1 1.00 100.00 6000.00 5.00
301 ta[3] = matOff.x;
302 ta[7] = matOff.y; 14 16 1.14 25.00 437.50 7.14 6.25
303 ta[11] = matOff.z; 4 9 2.25 777.78 1.43 11.11
304
After Optimization.
IPC is high at prefetchnta lines with sufficient samples,
Activate Windows
Go to Settings to activate Windows.

THANK YOU

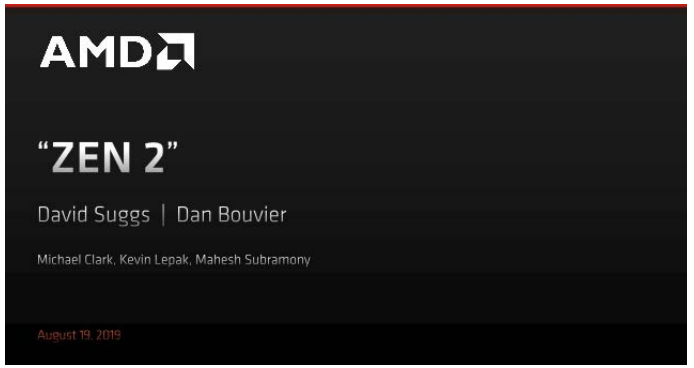
FURTHER READING

OPTIMIZATION GUIDE



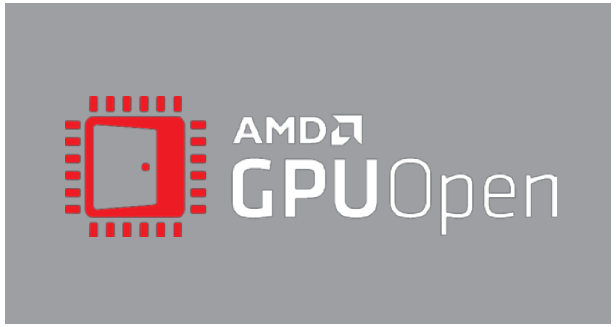
https://developer.amd.com/resources/developer-guides-manuals/

THE PATH TO “ZEN 2”



https://www.slideshare.net/AMD/the-path-to-zen-2

AGS SDK 5.4



https://gpuopen.com/ags-sdk-5-4-improves-handling-video-memory-reporting-apus/

CONTACT INFORMATION

- Front Row:
  - Ken Mitchell
    - Team Lead/Platform Specialist
    - Kenneth.Mitchell@amd.com
  - John Hartwig
    - Engine/Scalability Specialist
    - John.Hartwig@amd.com
- Back Row:
  - Elliot Kim
    - Physics/Simulation Specialist
    - Elliot.Kim@amd.com

DISCLAIMER AND ATTRIBUTION

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions and typographical errors.

The information contained herein is subject to change and may be rendered inaccurate for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product releases, product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION.

AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY DIRECT, INDIRECT, SPECIAL OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

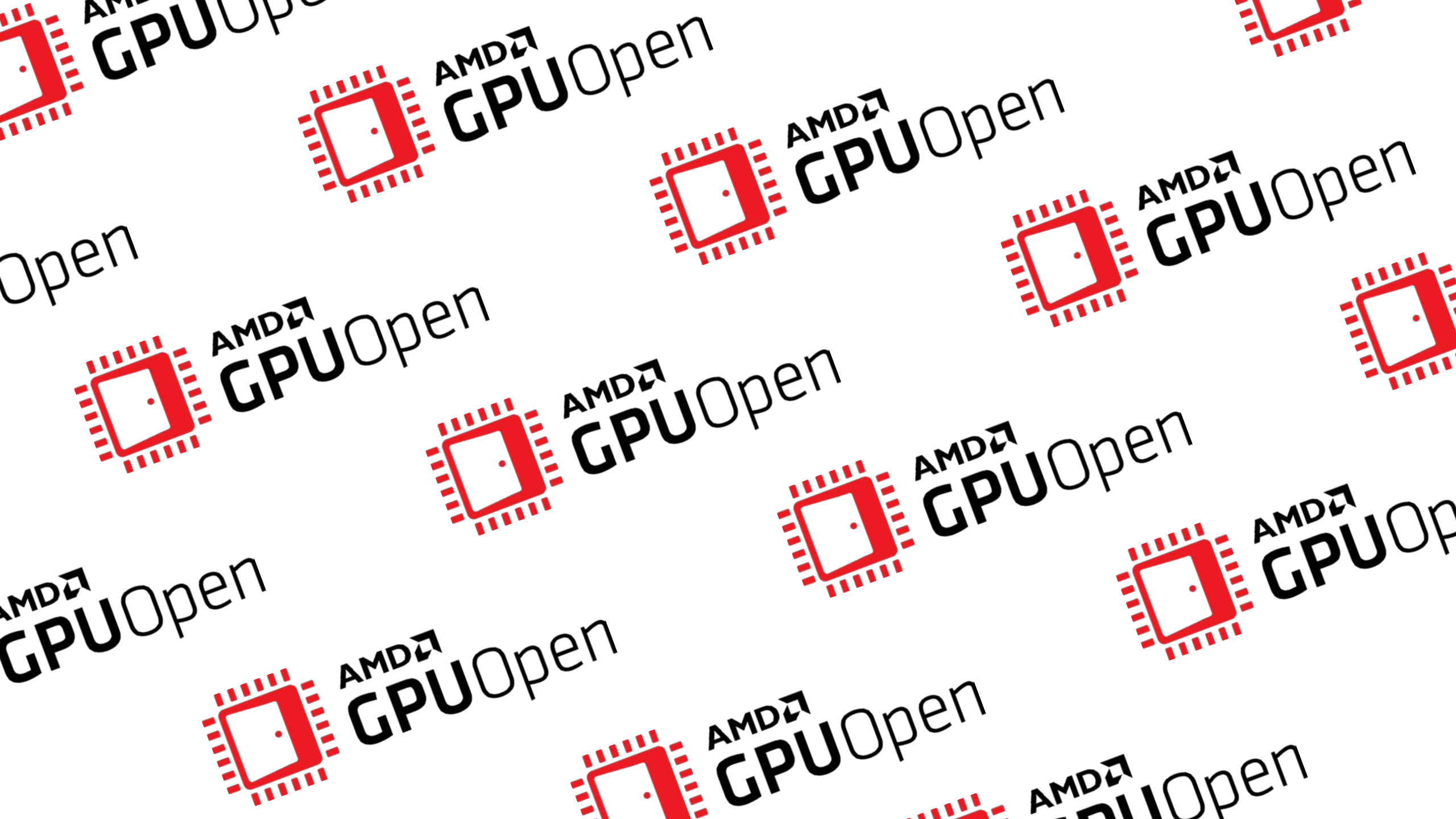
ATTRIBUTION

DISCLAIMER AND ATTRIBUTION (MORE)

“Code Sample excerpted on slides 92 and 109 are modifications Copyright (c) 2020 Advanced Micro Devices, Inc. All Rights Reserved. Copyright (c) 2019 NVIDIA Corporation. All rights reserved. Code Sample is licensed subject to the following: “Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of NVIDIA CORPORATION nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.”

DISCLAIMER AND ATTRIBUTION (MORE)

- Slide 10: Claim: 15% IPC Improvement
  - AMD “Zen 2” CPU-based system scored an estimated 15% higher than previous generation AMD “Zen” based system using estimated SPECInt®_base2006 results. SPEC and SPECint are registered trademarks of the Standard Performance Evaluation Corporation. See www.spec.org. GD-141
- Slides 47, 48, 50, 51, 53, 54, 72, 74, 76, 78, and 80 Windows Performance Analyzer used with permission from Microsoft


AMD
GPUOpen