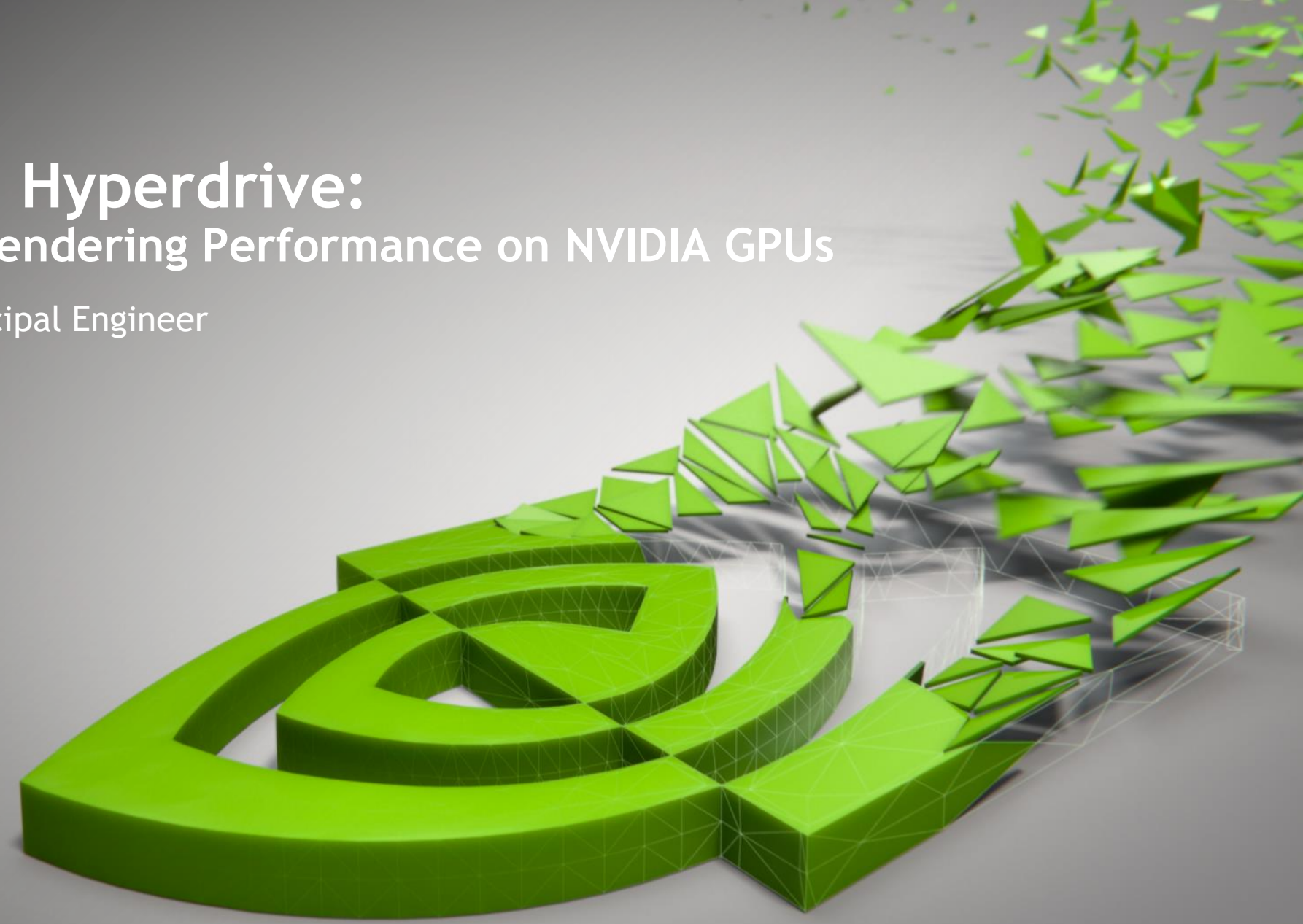
Fixing the Hyperdrive:
Maximizing Rendering Performance on NVIDIA GPUs

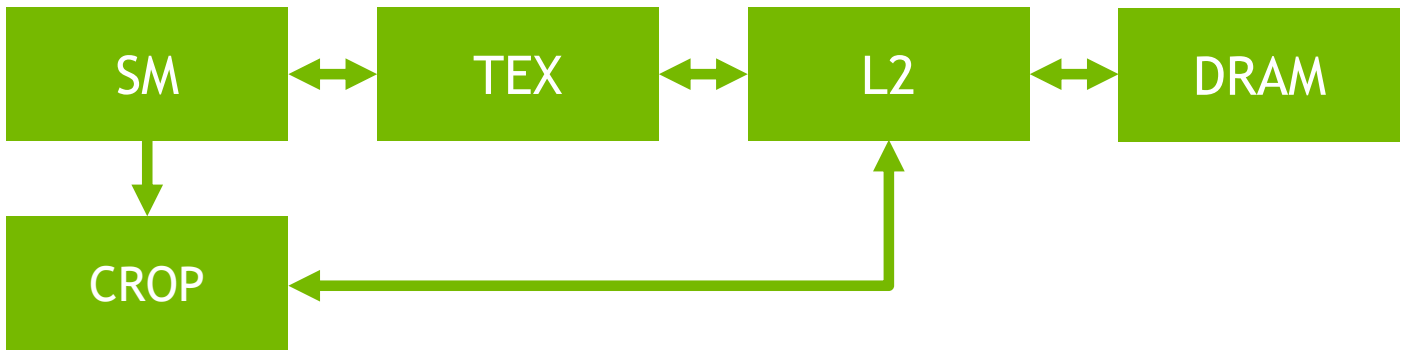
Louis Bavoil, Principal Engineer

NVIDIA

Booth #223 - South Hall
www.nvidia.com/GDC



Full-Screen Pixel Shader



SM = Streaming Multiprocessor

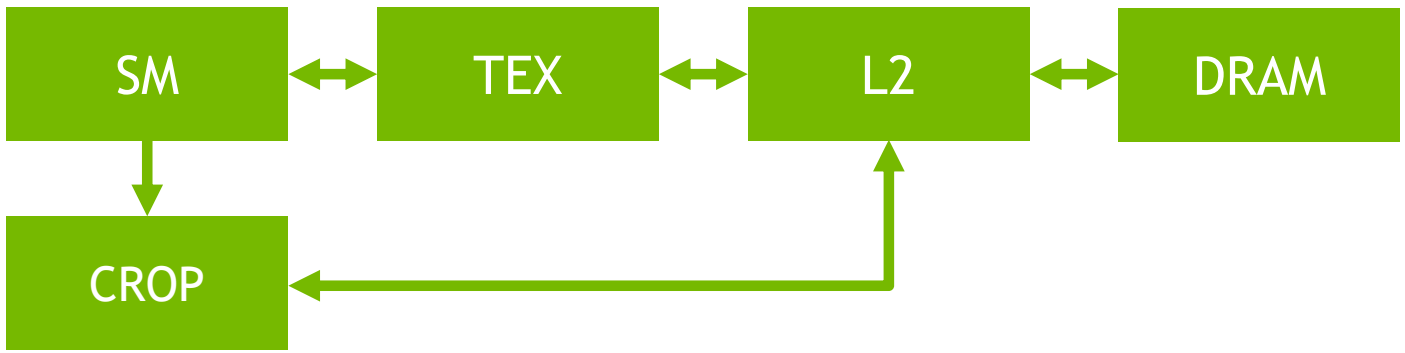
TEX = Texture unit

L2 = Level 2 cache

DRAM = physical video-memory unit

CROP = Color ROP

Speed Of Light (SOL) Metrics

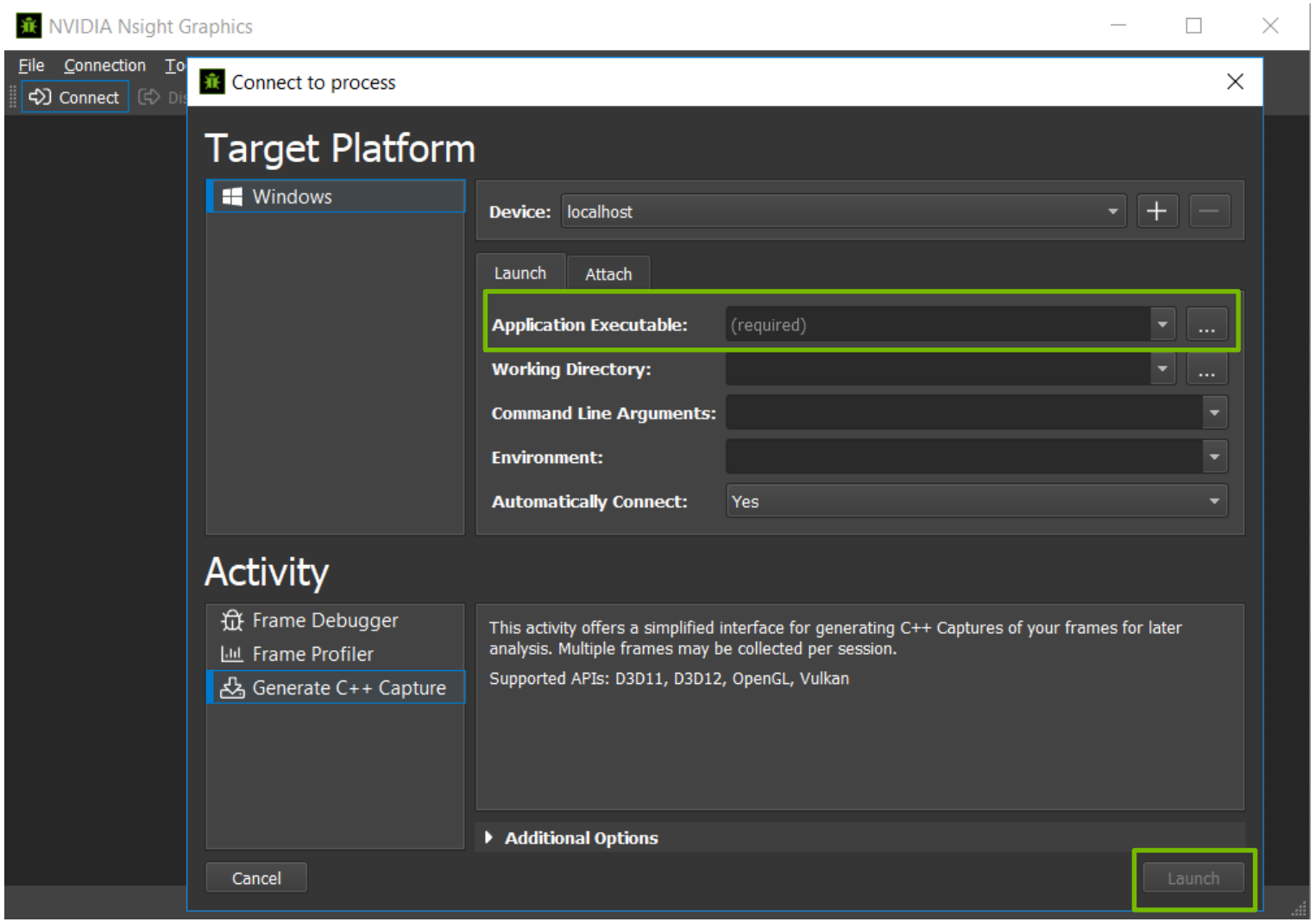


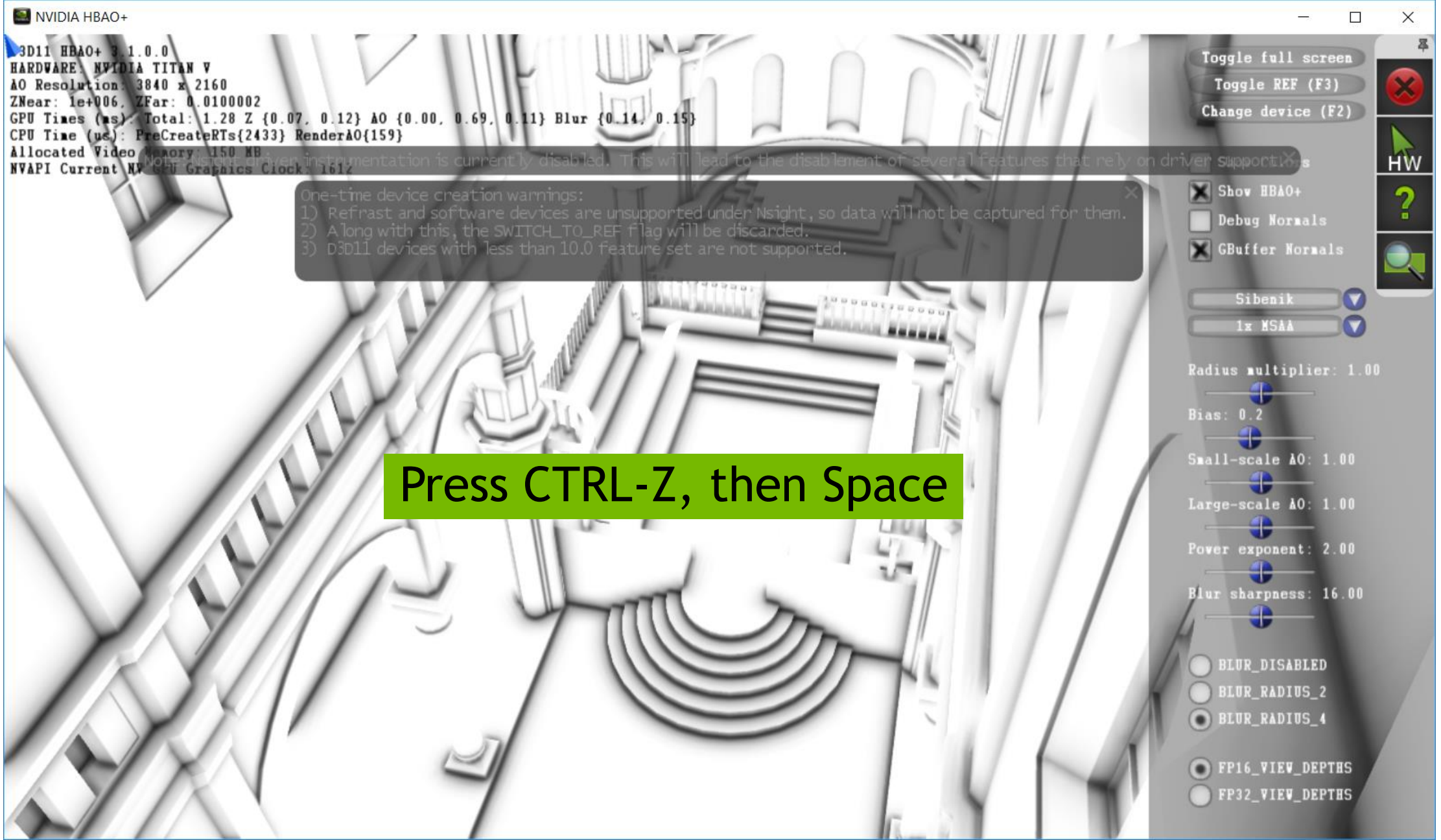
"SOL%" = % of Peak Performance

Top SOL%s [**SM**:95% | **TEX**:72% | **L2**:72% | **DRAM**:34% | **CROP**:5%]

Capturing a Frame from a DX App

Using Nsight Graphics 1.0




NVIDIA HBAO+
D3D11 HBAO+ 3.1.0.0
HARDWARE: NVIDIA TITAN V
AO Resolution: 3840 x 2160
ZNear: 1e+006, ZFar: 0.0100002
GPU Times (ms): Total: 1.28 Z {0.07, 0.12} AO {0.00, 0.69, 0.11} Blur {0.14, 0.15}
CPU Time (us): PreCreateRTs{2433} RenderAO{159}
Allocated Video Memory: 150 MB
NVAPI Current NV GPU Graphics Clock: 1612
Nsight GPU trace activity and instrumentation is currently disabled. This will lead to the disablement of several features that rely on driver support.
One-time device creation warnings:
1) Refrast and software devices are unsupported under Nsight, so data will not be captured for them.
2) Along with this, the SWITCH_TO_REF flag will be discarded.
3) D3D11 devices with less than 10.0 feature set are not supported.
Press CTRL-Z, then Space
Toggle full screen
Toggle REF (F3)
Change device (F2)
HW
Show HBAO+
Debug Normals
GBuffer Normals
Sibenik
1x MSAA
Radius multiplier: 1.00
Bias: 0.2
Small-scale AO: 1.00
Large-scale AO: 1.00
Power exponent: 2.00
Blur sharpness: 16.00
BLUR_DISABLED
BLUR_RADIUS_2
BLUR_RADIUS_4
FP16_VIEW_DEPTHS
FP32_VIEW_DEPTHS

HBAO+ - NVIDIA Nsight Graphics

File Connection C++ Capture Tools Window Help

Connect Disconnect Terminate Generate C++ Capture

Project Explorer

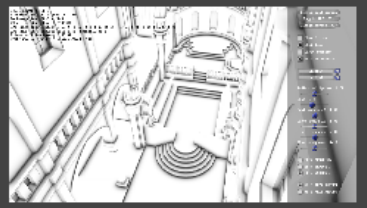
Search project...

HBAO+

Viewer__2018_03_21__15_05_20.nsight-gfxcppcap

Viewer__2018_03_21__15_05_20.nsight-gfxcppcap

Output Image



Application Details

Application: Viewer.exe
Arguments:
Host Name: localhost
Capture Date: 3/21/2018 3:05 PM
GPU(s): GeForce GTX 1080
Driver Version: 391.1

Capture Statistics

API: D3D11.3
 # Events: 797
 # Actions: 95

Build

Tool Chain: vs2017

Tools: Build Clean Open Folder

Build Log:

Comments

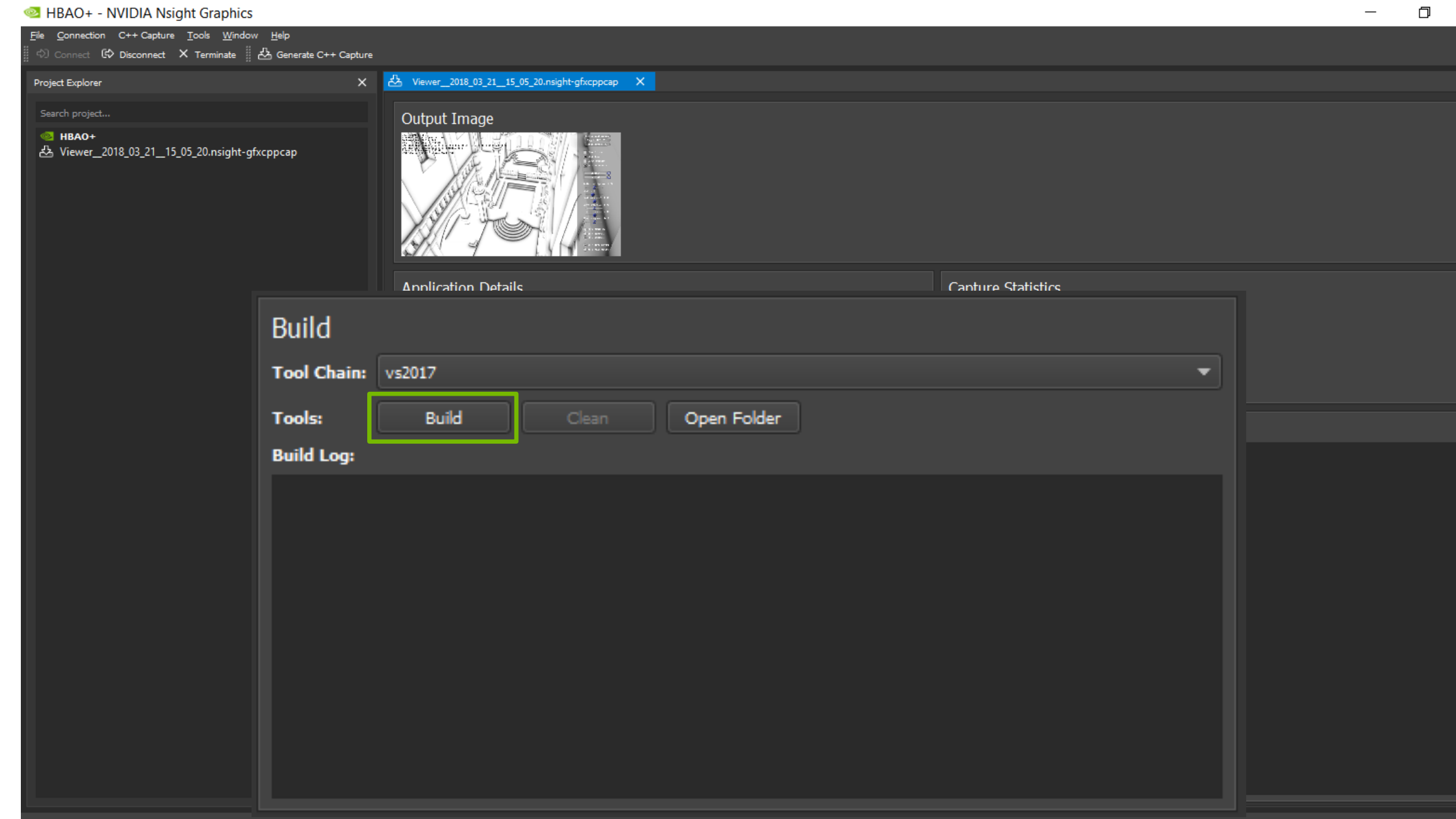
Add comments...

Run

Status: Not Yet Built

Options: ✓ Reset state between frames

Run: Execute Connect...


HBAO+ - NVIDIA Nsight Graphics
File Connection C++ Capture Tools Window Help
Connect Disconnect Terminate Generate C++ Capture
Project Explorer
Search project...
HBAO+
Viewer__2018_03_21__15_05_20.nsight-gfxcppcap
Viewer__2018_03_21__15_05_20.nsight-gfxcppcap
Output Image
Application Details
Capture Statistics
Build
Tool Chain: vs2017
Tools: Build Clean Open Folder
Build Log:

HBAO+ - NVIDIA Nsight Graphics

File Connection C++ Capture Tools Window Help

Connect Disconnect Terminate Generate C++ Capture

Project Explorer

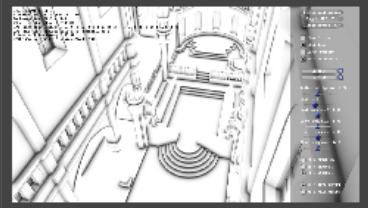
Search project...

HBAO+

Viewer__2018_03_21__15_05_20.nsight-gfxcppcap

Viewer__2018_03_21__15_05_20.nsight-gfxcppcap

Output Image



Application Details

Capture Statistics

Build

Tool Chain: vs2017

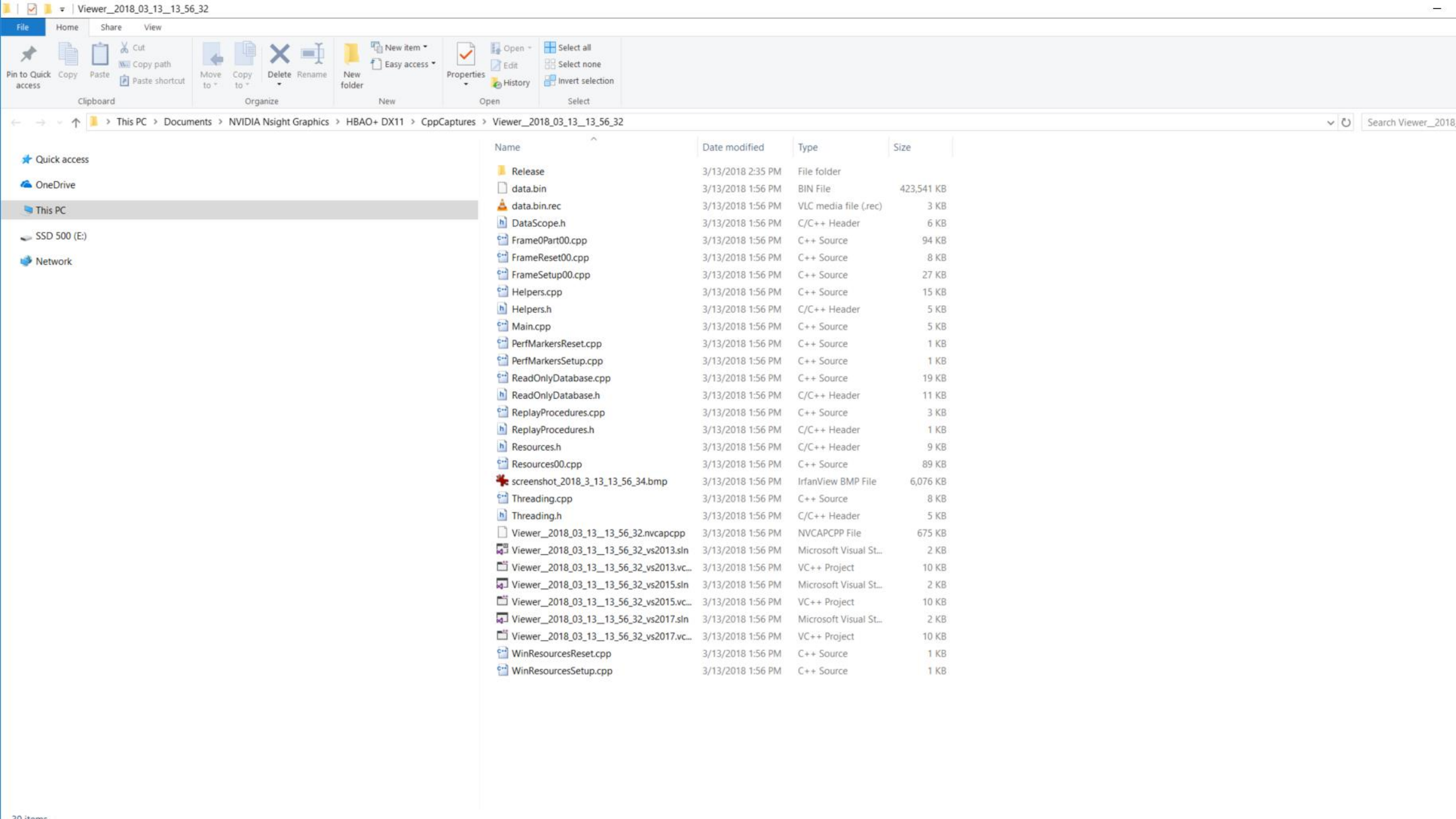
Tools: Build | Clean | Open Folder

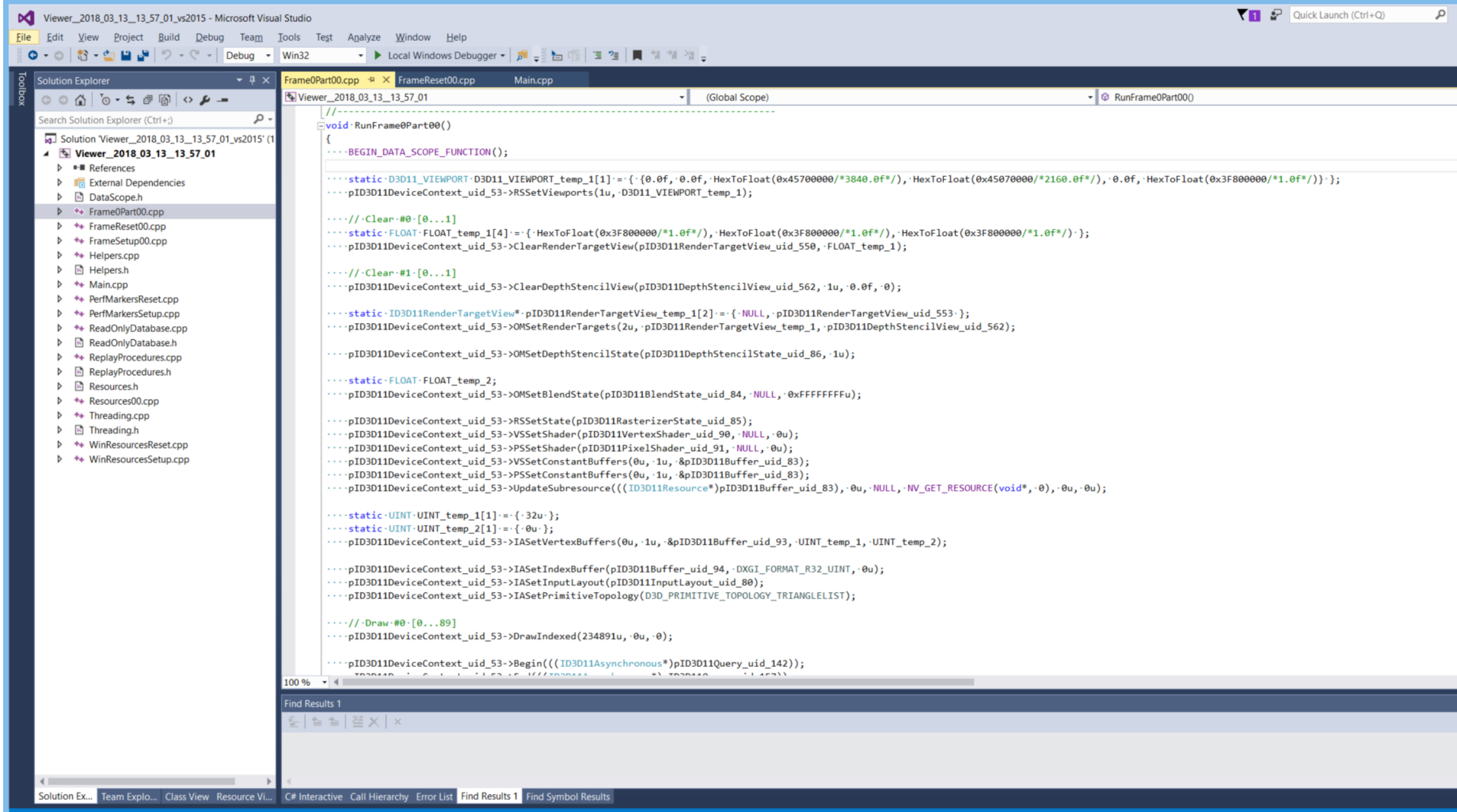
Build Log:

```
  Deleting file "Release\Viewer__.2626BEA9.tlog\unsuccessfulbuild".
  Touching "Release\Viewer__.2626BEA9.tlog\Viewer__2018_03_21__15_05_20.lastbuildstate".
Done Building Project "C:\Users\lbavoil.NVIDIA.COM\Documents\NVIDIA Nsight Graphics\HBAO+
\CppCaptures\Viewer__2018_03_21__15_05_20\Viewer__2018_03_21__15_05_20_vs2017.vcxproj"
(default targets).
Done Building Project "C:\Users\lbavoil.NVIDIA.COM\Documents\NVIDIA Nsight Graphics\HBAO+
\CppCaptures\Viewer__2018_03_21__15_05_20\Viewer__2018_03_21__15_05_20_vs2017.sln"
(Viewer__2018_03_21__15_05_20 target(s)).

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:06.66
```


Viewer__2018_03_13__13_56_32
File
Home
Share
View
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Copy to
Delete
Rename
Organize
New folder
New item
Easy access
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
This PC › Documents › NVIDIA Nsight Graphics › HBAO+ DX11 › CppCaptures › Viewer__2018_03_13__13_56_32
Search Viewer__2018_
Quick access
OneDrive
This PC
SSD 500 (E:)
Network
Name
Date modified
Type
Size
Release 3/13/2018 2:35 PM File folder
data.bin 3/13/2018 1:56 PM BIN File 423,541 KB
data.bin.rec 3/13/2018 1:56 PM VLC media file (.rec) 3 KB
DataScope.h 3/13/2018 1:56 PM C/C++ Header 6 KB
Frame0Part00.cpp 3/13/2018 1:56 PM C++ Source 94 KB
FrameReset00.cpp 3/13/2018 1:56 PM C++ Source 8 KB
FrameSetup00.cpp 3/13/2018 1:56 PM C++ Source 27 KB
Helpers.cpp 3/13/2018 1:56 PM C++ Source 15 KB
Helpers.h 3/13/2018 1:56 PM C/C++ Header 5 KB
Main.cpp 3/13/2018 1:56 PM C++ Source 5 KB
PerfMarkersReset.cpp 3/13/2018 1:56 PM C++ Source 1 KB
PerfMarkersSetup.cpp 3/13/2018 1:56 PM C++ Source 1 KB
ReadOnlyDatabase.cpp 3/13/2018 1:56 PM C++ Source 19 KB
ReadOnlyDatabase.h 3/13/2018 1:56 PM C/C++ Header 11 KB
ReplayProcedures.cpp 3/13/2018 1:56 PM C++ Source 3 KB
ReplayProcedures.h 3/13/2018 1:56 PM C/C++ Header 1 KB
Resources.h 3/13/2018 1:56 PM C/C++ Header 9 KB
Resources00.cpp 3/13/2018 1:56 PM C++ Source 89 KB
screenshot_2018_3_13_13_56_34.bmp 3/13/2018 1:56 PM IrfanView BMP File 6,076 KB
Threading.cpp 3/13/2018 1:56 PM C++ Source 8 KB
Threading.h 3/13/2018 1:56 PM C/C++ Header 5 KB
Viewer__2018_03_13__13_56_32.nvcapcpp 3/13/2018 1:56 PM NVCAPCPP File 675 KB
Viewer__2018_03_13__13_56_32_vs2013.sln 3/13/2018 1:56 PM Microsoft Visual St... 2 KB
Viewer__2018_03_13__13_56_32_vs2013.vc... 3/13/2018 1:56 PM VC++ Project 10 KB
Viewer__2018_03_13__13_56_32_vs2015.sln 3/13/2018 1:56 PM Microsoft Visual St... 2 KB
Viewer__2018_03_13__13_56_32_vs2015.vc... 3/13/2018 1:56 PM VC++ Project 10 KB
Viewer__2018_03_13__13_56_32_vs2017.sln 3/13/2018 1:56 PM Microsoft Visual St... 2 KB
Viewer__2018_03_13__13_56_32_vs2017.vc... 3/13/2018 1:56 PM VC++ Project 10 KB
WinResourcesReset.cpp 3/13/2018 1:56 PM C++ Source 1 KB
WinResourcesSetup.cpp 3/13/2018 1:56 PM C++ Source 1 KB


Viewer__2018_03_13__13_57_01_vs2015 - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug Win32 Local Windows Debugger
Quick Launch (Ctrl+Q)
Toolbox
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Viewer__2018_03_13__13_57_01_vs2015' (1
Viewer__2018_03_13__13_57_01
References
External Dependencies
DataScope.h
Frame0Part00.cpp
FrameReset00.cpp
FrameSetup00.cpp
Helpers.cpp
Helpers.h
Main.cpp
PerfMarkersReset.cpp
PerfMarkersSetup.cpp
ReadOnlyDatabase.cpp
ReadOnlyDatabase.h
ReplayProcedures.cpp
ReplayProcedures.h
Resources.h
Resources00.cpp
Threading.cpp
Threading.h
WinResourcesReset.cpp
WinResourcesSetup.cpp
Frame0Part00.cpp
FrameReset00.cpp
Main.cpp
Viewer__2018_03_13__13_57_01
(Global Scope)
RunFrame0Part00()
//--
void RunFrame0Part00()
{
BEGIN_DATA_SCOPE_FUNCTION();
static D3D11_VIEWPORT D3D11_VIEWPORT_temp_1[1] = {{0.0f, 0.0f, HexToFloat(0x45700000/*3840.0f*/), HexToFloat(0x45070000/*2160.0f*/), 0.0f, HexToFloat(0x3F800000/*1.0f*/)}};
pID3D11DeviceContext_uid_53->RSSetViewports(1u, D3D11_VIEWPORT_temp_1);
//Clear #0 [0...1]
static FLOAT FLOAT_temp_1[4] = {HexToFloat(0x3F800000/*1.0f*/), HexToFloat(0x3F800000/*1.0f*/), HexToFloat(0x3F800000/*1.0f*/), 0};
pID3D11DeviceContext_uid_53->ClearRenderTargetView(pID3D11RenderTargetView_uid_550, FLOAT_temp_1);
//Clear #1 [0...1]
pID3D11DeviceContext_uid_53->ClearDepthStencilView(pID3D11DepthStencilView_uid_562, 1u, 0.0f, 0);
static ID3D11RenderTargetView* pID3D11RenderTargetView_temp_1[2] = {NULL, pID3D11RenderTargetView_uid_553};
pID3D11DeviceContext_uid_53->OMSetRenderTargets(2u, pID3D11RenderTargetView_temp_1, pID3D11DepthStencilView_uid_562);
pID3D11DeviceContext_uid_53->OMSetDepthStencilState(pID3D11DepthStencilState_uid_86, 1u);
static FLOAT FLOAT_temp_2;
pID3D11DeviceContext_uid_53->OMSetBlendState(pID3D11BlendState_uid_84, NULL, 0xFFFFFFFFu);
pID3D11DeviceContext_uid_53->RSSetState(pID3D11RasterizerState_uid_85);
pID3D11DeviceContext_uid_53->VSSetShader(pID3D11VertexShader_uid_90, NULL, 0u);
pID3D11DeviceContext_uid_53->PSSetShader(pID3D11PixelShader_uid_91, NULL, 0u);
pID3D11DeviceContext_uid_53->VSSetConstantBuffers(0u, 1u, &pID3D11Buffer_uid_83);
pID3D11DeviceContext_uid_53->PSSetConstantBuffers(0u, 1u, &pID3D11Buffer_uid_83);
pID3D11DeviceContext_uid_53->UpdateSubresource(((ID3D11Resource*)pID3D11Buffer_uid_83), 0u, NULL, NV_GET_RESOURCE(void*, 0), 0u, 0u);
static UINT UINT_temp_1[1] = {32u};
static UINT UINT_temp_2[1] = {0u};
pID3D11DeviceContext_uid_53->IASetVertexBuffers(0u, 1u, &pID3D11Buffer_uid_93, UINT_temp_1, UINT_temp_2);
pID3D11DeviceContext_uid_53->IASetIndexBuffer(pID3D11Buffer_uid_94, DXGI_FORMAT_R32_UINT, 0u);
pID3D11DeviceContext_uid_53->IASetInputLayout(pID3D11InputLayout_uid_80);
pID3D11DeviceContext_uid_53->IASetPrimitiveTopology(D3D_PRIMITIVE_TOPOLOGY_TRIANGLELIST);
//Draw #0 [0...89]
pID3D11DeviceContext_uid_53->DrawIndexed(234891u, 0u, 0);
pID3D11DeviceContext_uid_53->Begin(((ID3D11Asynchronous*)pID3D11Query_uid_142));
100 %
Find Results 1
Solution Ex... Team Explo... Class View Resource Vi...
C# Interactive Call Hierarchy Error List Find Results 1 Find Symbol Results
Ready
Ln 18 Col 5 Ch 5 INS

Test - NVIDIA Nsight Graphics

File Connection C++ Capture Tools Window Help

Connect Disconnect Terminate Generate C++ Capture

Project Explorer

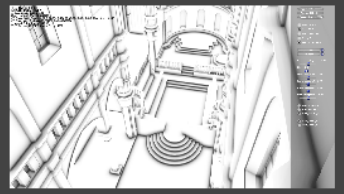
Search project...

Test

Viewer__2018_03_13__...

Viewer__2018_03_13__13_56_32.nvcapcpp

Output Image



Application Details

Application:	Viewer.exe
Arguments:	
Host Name:	localhost
Capture Date:	3/13/2018 1:56 PM
GPU(s):	TITAN V
Driver Version:	391.1

Capture Statistics

API:	D3D11.3
# Events:	813
# Actions:	97

Build

Tool Chain: vs2017

Comments

Add comments...

Run

Status: Built

Options: ☑ Reset state between frames

Run: Execute Connect...

Status: Built

Options: ☑ Reset state between frames

Run: Execute Connect...

Test - NVIDIA Nsight Graphics

File Connection C++ Capture Tools Window Help

Connect Disconnect Terminate Generate C++ Capture

Project Explorer

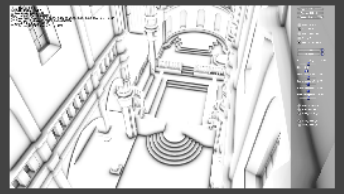
Search project...

Test

Viewer__2018_03_13_...

Viewer__2018_03_13__13_56_32.nvcapcpp

Output Image



Application Details

Application:	Viewer.exe
Arguments:	
Host Name:	localhost
Capture Date:	3/13/2018 1:56 PM
GPU(s):	TITAN V
Driver Version:	391.1

Capture Statistics

API:	D3D11.3
# Events:	813
# Actions:	97

Build

Tool Chain: vs2017

Comments

Add comments...

Run

Status: Built

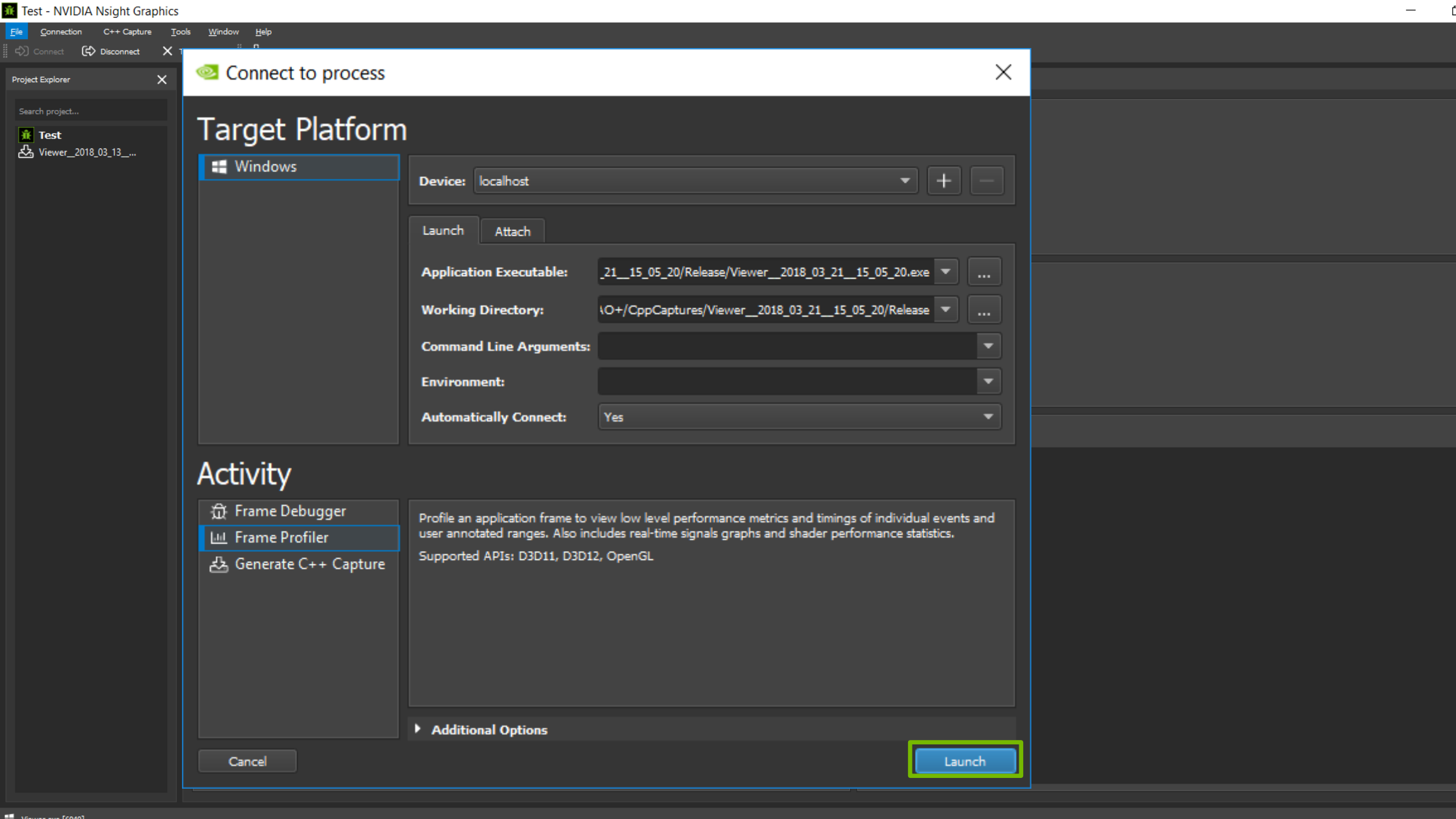
Options: ☑ Reset state between frames

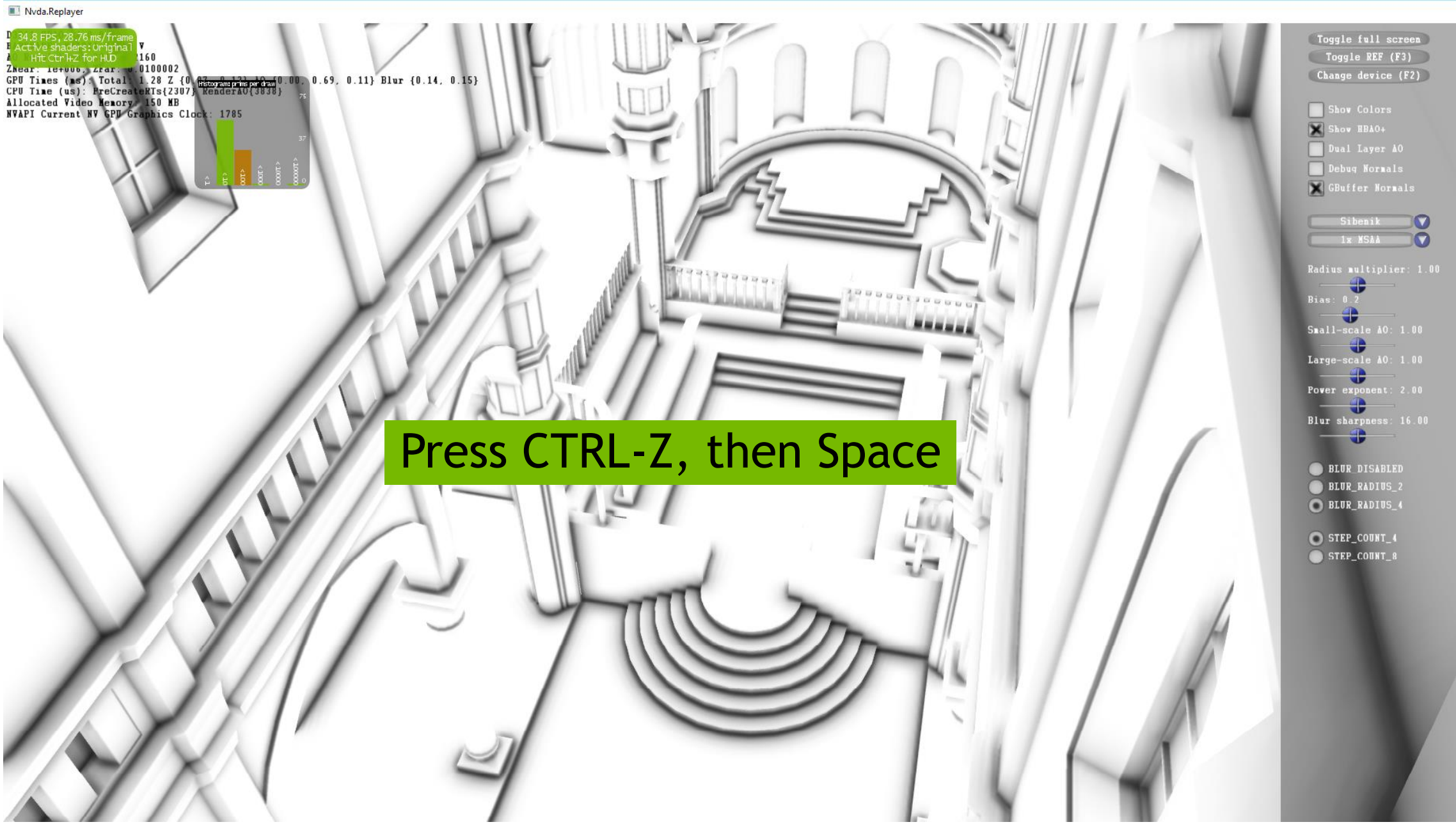
Run: Execute | Connect...

Status: Built

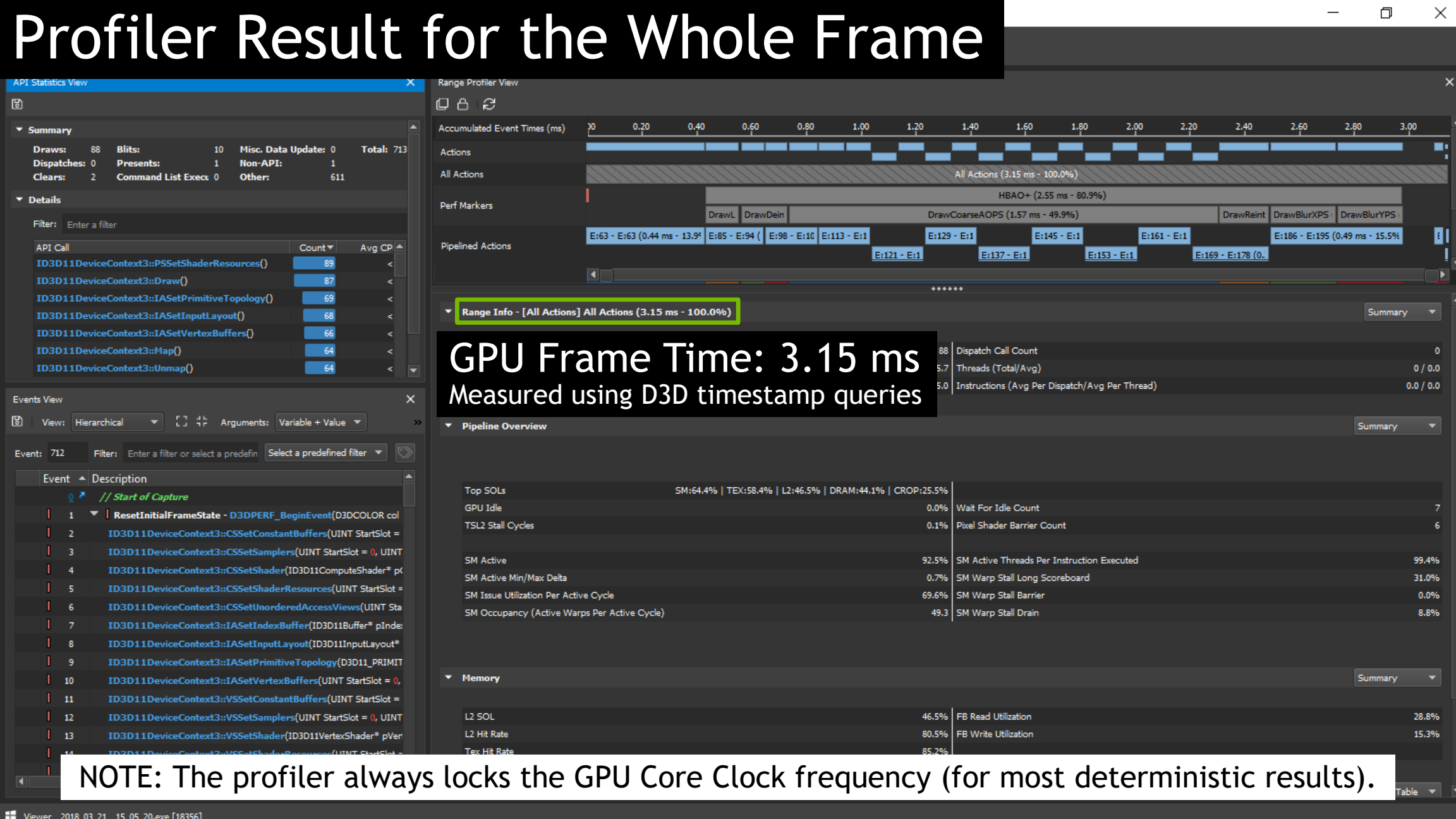
Options: ☑ Reset state between frames

Run: Execute | Connect...


Test - NVIDIA Nsight Graphics
File
Connection
C++ Capture
Tools
Window
Help
Connect
Disconnect
Project Explorer
Search project...
Test
Viewer__2018_03_13_...
Connect to process
Target Platform
Windows
Device:
localhost
Launch
Attach
Application Executable:
_21__15_05_20/Release/Viewer__2018_03_21__15_05_20.exe
Working Directory:
\O+/CppCaptures/Viewer__2018_03_21__15_05_20/Release
Command Line Arguments:
Environment:
Automatically Connect:
Yes
Activity
Frame Debugger
Frame Profiler
Generate C++ Capture
Profile an application frame to view low level performance metrics and timings of individual events and user annotated ranges. Also includes real-time signals graphs and shader performance statistics.
Supported APIs: D3D11, D3D12, OpenGL
Additional Options
Cancel
Launch


Nvda.Replayer
34.8 FPS, 28.76 ms/frame
Active shaders: Original
Hit Ctrl+Z for HUD
GPU Times (ms): Total: 1.28 Z {0.07, 0.13} AO {0.00, 0.69, 0.11} Blur {0.14, 0.15}
CPU Time (us): PreCreateRTs{2307} RenderAO{3838}
Allocated Video Memory: 150 MB
NVAPI Current NV GPU Graphics Clock: 1785
Histogram: prims per draw
Press CTRL-Z, then Space
Toggle full screen
Toggle REF (F3)
Change device (F2)
Show Colors
Show HBAO+
Dual Layer AO
Debug Normals
GBuffer Normals
Sibenik
1x MSAA
Radius multiplier: 1.00
Bias: 0.2
Small-scale AO: 1.00
Large-scale AO: 1.00
Power exponent: 2.00
Blur sharpness: 16.00
BLUR_DISABLED
BLUR_RADIUS_2
BLUR_RADIUS_4
STEP_COUNT_4
STEP_COUNT_8

Profiler Result for the Whole Frame

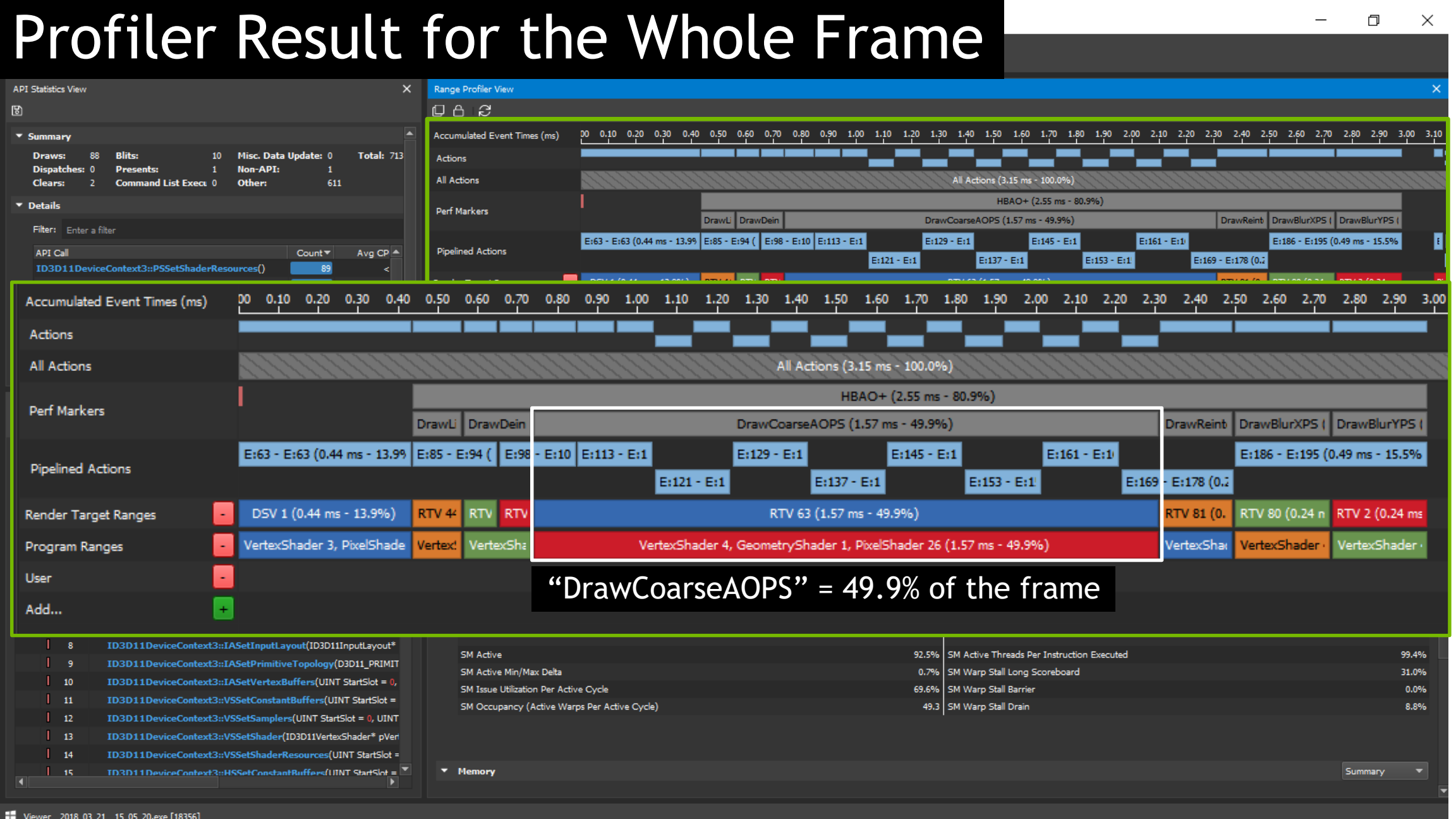


GPU Frame Time: 3.15 ms

Measured using D3D timestamp queries

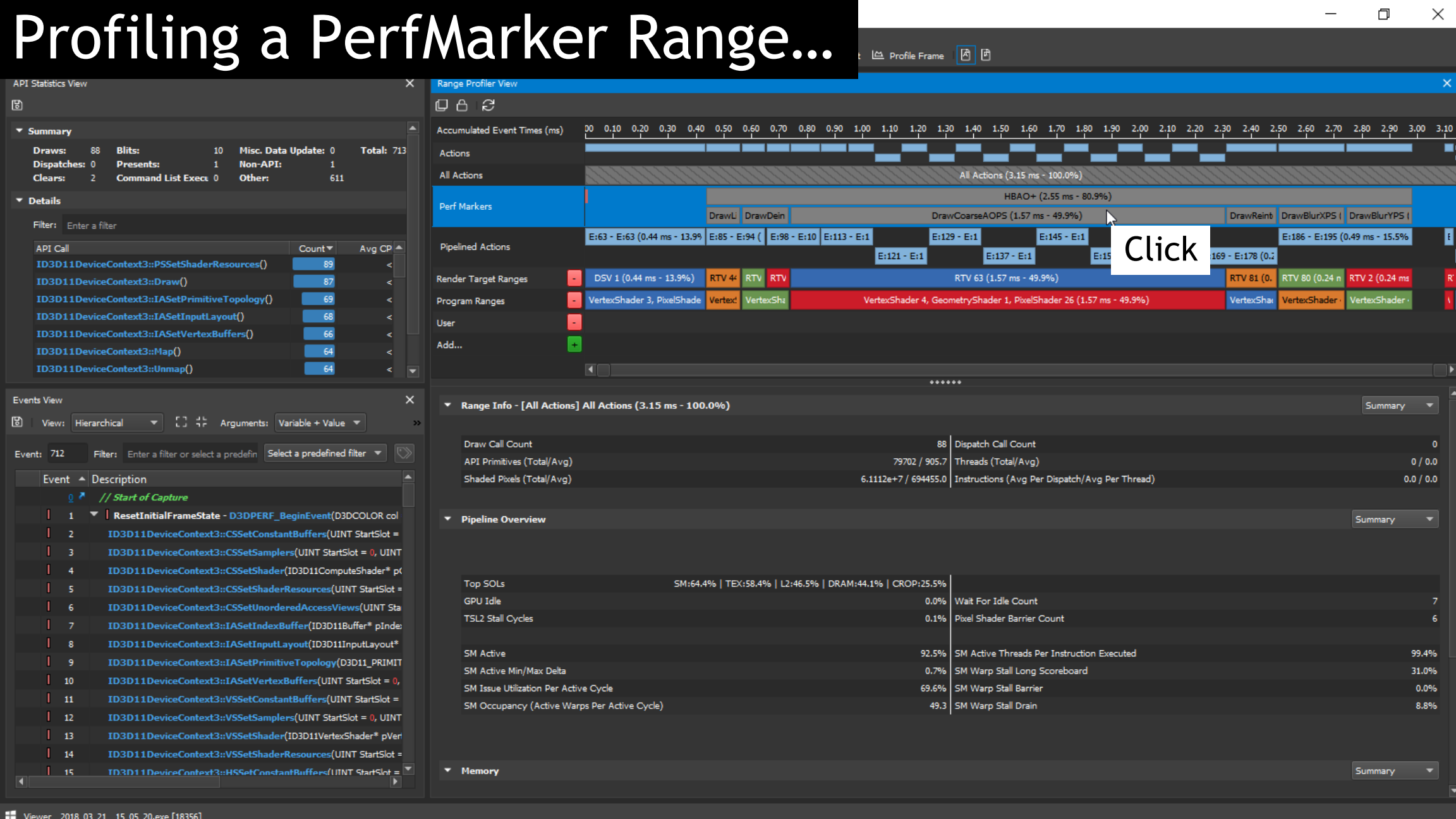
NOTE: The profiler always locks the GPU Core Clock frequency (for most deterministic results).

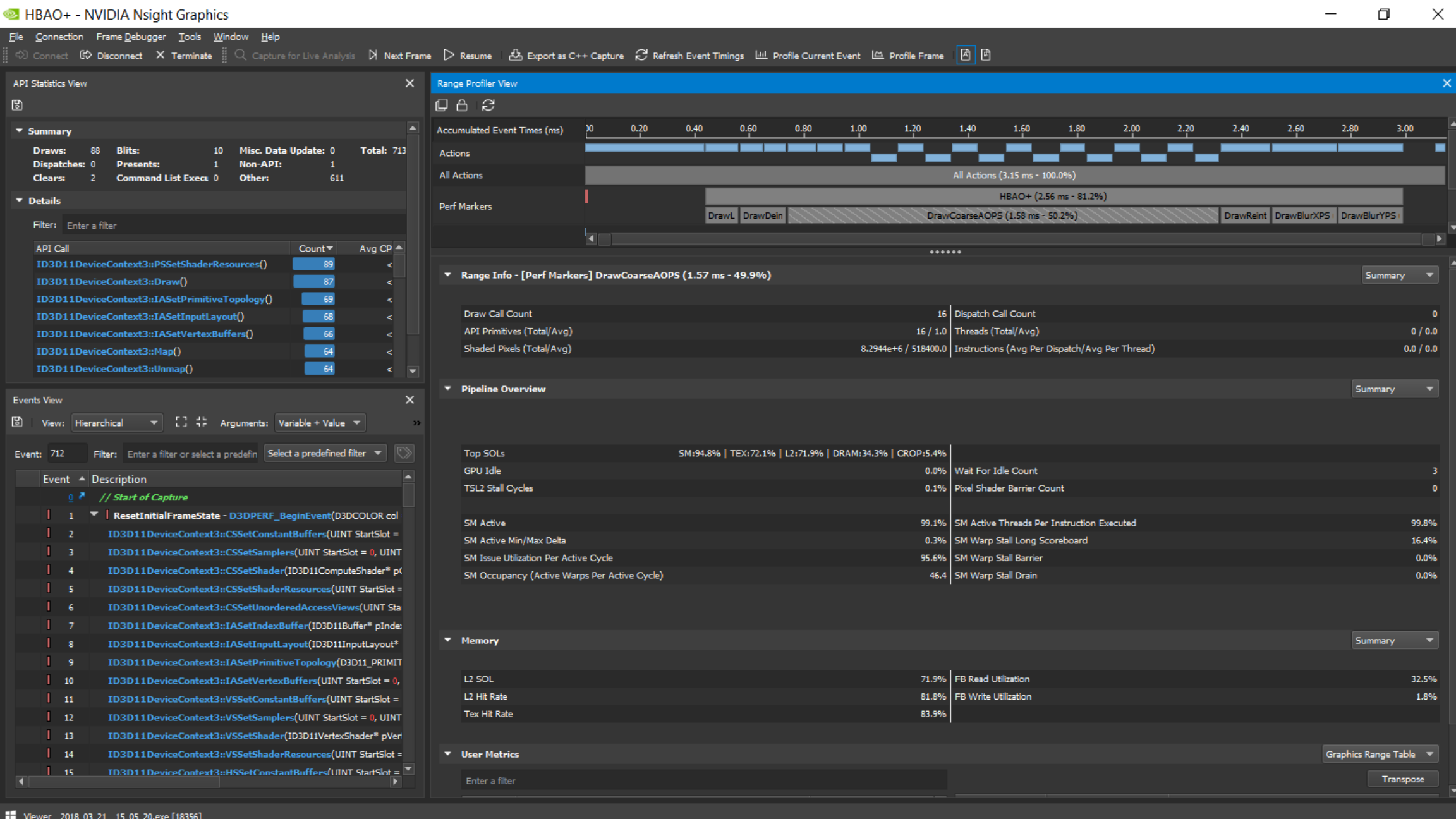
Profiler Result for the Whole Frame



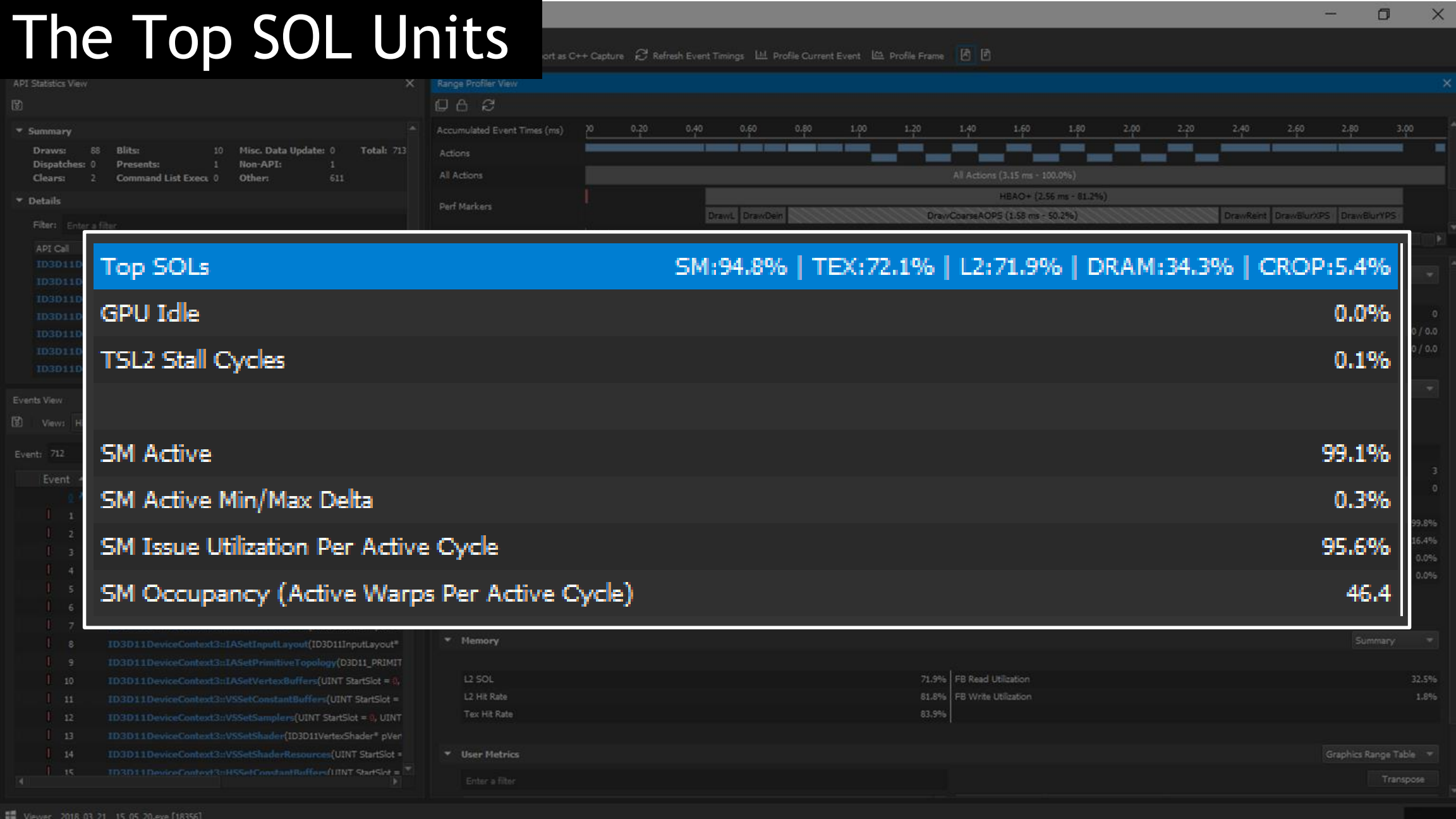
"DrawCoarseAOPS" = 49.9% of the frame

Profiling a PerfMarker Range…




HBAO+ - NVIDIA Nsight Graphics
File Connection Frame Debugger Tools Window Help
Connect Disconnect Terminate Capture for Live Analysis Next Frame Resume Export as C++ Capture Refresh Event Timings Profile Current Event Profile Frame
API Statistics View
Summary
Draws: 88 Blits: 10 Misc. Data Update: 0 Total: 713
Dispatches: 0 Presents: 1 Non-API: 1
Clears: 2 Command List Execu 0 Other: 611
Details
Filter: Enter a filter
API Call Count Avg CP
ID3D11DeviceContext3::PSSetShaderResources() 89
ID3D11DeviceContext3::Draw() 87
ID3D11DeviceContext3::IASetPrimitiveTopology() 69
ID3D11DeviceContext3::IASetInputLayout() 68
ID3D11DeviceContext3::IASetVertexBuffers() 66
ID3D11DeviceContext3::Map() 64
ID3D11DeviceContext3::Unmap() 64
Events View
View: Hierarchical Arguments: Variable + Value
Event: 712 Filter: Enter a filter or select a predefin Select a predefined filter
Event Description
0 // Start of Capture
1 ResetInitialFrameState - D3DPERF_BeginEvent(D3DCOLOR col
2 ID3D11DeviceContext3::CSSetConstantBuffers(UINT StartSlot =
3 ID3D11DeviceContext3::CSSetSamplers(UINT StartSlot = 0, UINT
4 ID3D11DeviceContext3::CSSetShader(ID3D11ComputeShader* p(
5 ID3D11DeviceContext3::CSSetShaderResources(UINT StartSlot =
6 ID3D11DeviceContext3::CSSetUnorderedAccessViews(UINT Sta
7 ID3D11DeviceContext3::IASetIndexBuffer(ID3D11Buffer* pInde
8 ID3D11DeviceContext3::IASetInputLayout(ID3D11InputLayout*
9 ID3D11DeviceContext3::IASetPrimitiveTopology(D3D11_PRIMIT
10 ID3D11DeviceContext3::IASetVertexBuffers(UINT StartSlot = 0,
11 ID3D11DeviceContext3::VSSetConstantBuffers(UINT StartSlot =
12 ID3D11DeviceContext3::VSSetSamplers(UINT StartSlot = 0, UINT
13 ID3D11DeviceContext3::VSSetShader(ID3D11VertexShader* pVer
14 ID3D11DeviceContext3::VSSetShaderResources(UINT StartSlot =
15 ID3D11DeviceContext3::HSSetConstantBuffers(UINT StartSlot =
Range Profiler View
Accumulated Event Times (ms)
0 0.20 0.40 0.60 0.80 1.00 1.20 1.40 1.60 1.80 2.00 2.20 2.40 2.60 2.80 3.00
Actions
All Actions
All Actions (3.15 ms - 100.0%)
Perf Markers
HBAO+ (2.56 ms - 81.2%)
DrawL DrawDein DrawCoarseAOPS (1.58 ms - 50.2%) DrawReint DrawBlurXPS DrawBlurYPS
Range Info - [Perf Markers] DrawCoarseAOPS (1.57 ms - 49.9%)
Summary
Draw Call Count 16
API Primitives (Total/Avg) 16 / 1.0
Shaded Pixels (Total/Avg) 8.2944e+6 / 518400.0
Dispatch Call Count 0
Threads (Total/Avg) 0 / 0.0
Instructions (Avg Per Dispatch/Avg Per Thread) 0.0 / 0.0
Pipeline Overview
Summary
Top SOLs SM:94.8% | TEX:72.1% | L2:71.9% | DRAM:34.3% | CROP:5.4%
GPU Idle 0.0%
TSL2 Stall Cycles 0.1%
SM Active 99.1%
SM Active Min/Max Delta 0.3%
SM Issue Utilization Per Active Cycle 95.6%
SM Occupancy (Active Warps Per Active Cycle) 46.4
Wait For Idle Count 3
Pixel Shader Barrier Count 0
SM Active Threads Per Instruction Executed 99.8%
SM Warp Stall Long Scoreboard 16.4%
SM Warp Stall Barrier 0.0%
SM Warp Stall Drain 0.0%
Memory
Summary
L2 SOL 71.9%
L2 Hit Rate 81.8%
Tex Hit Rate 83.9%
FB Read Utilization 32.5%
FB Write Utilization 1.8%
User Metrics
Graphics Range Table
Enter a filter
Transpose
Viewer 2018_03_21_15_05_20.exe [18356]

The Top SOL Units



The Peak-Perf% Analysis Method

For each “Top SOL%” unit:

1. **If SOL% > 80% ➔ (A) try removing work from this unit**
   - If SM: By opportunistically skipping instructions using branches (or early depth test)
   - If SM: By moving math instructions to lookup tables
   - If TEX: By moving structured-buffer loads to constant-buffer loads, etc.
2. **If SOL% < 60% ➔ (B) try increasing the SOL% of this unit**
   - By removing “idle cycles” (GPU unit is not doing any work for a % of the time)
   - By removing “stall cycles” (GPU unit has internal inefficiencies)
   - By avoiding “slow paths” if possible (e.g. 32-bit index buffers, and FP32x4 textures)
3. If SOL% in [60,80], do both (A) and (B)

Range Profiling & Async Compute

- For DX12, Nsight Frame Captures flatten all async COMPUTE queues to the main DIRECT queue

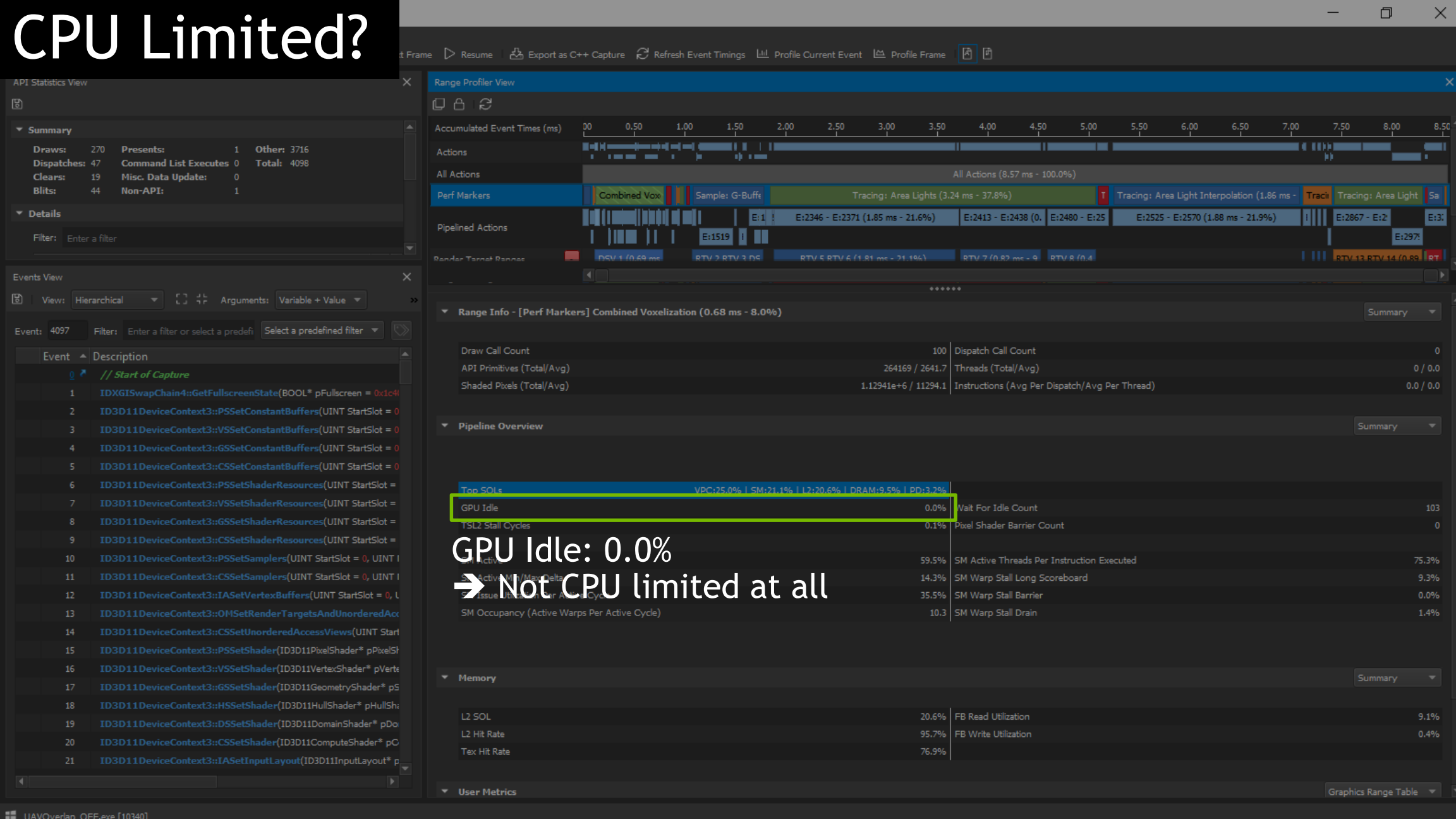
- For understanding overlaps of async compute work with graphics work, Nsight **GPU Trace** can be used

Example DX11 Workload:

Voxelization using UAV Atomics

GPU: GTX 1080

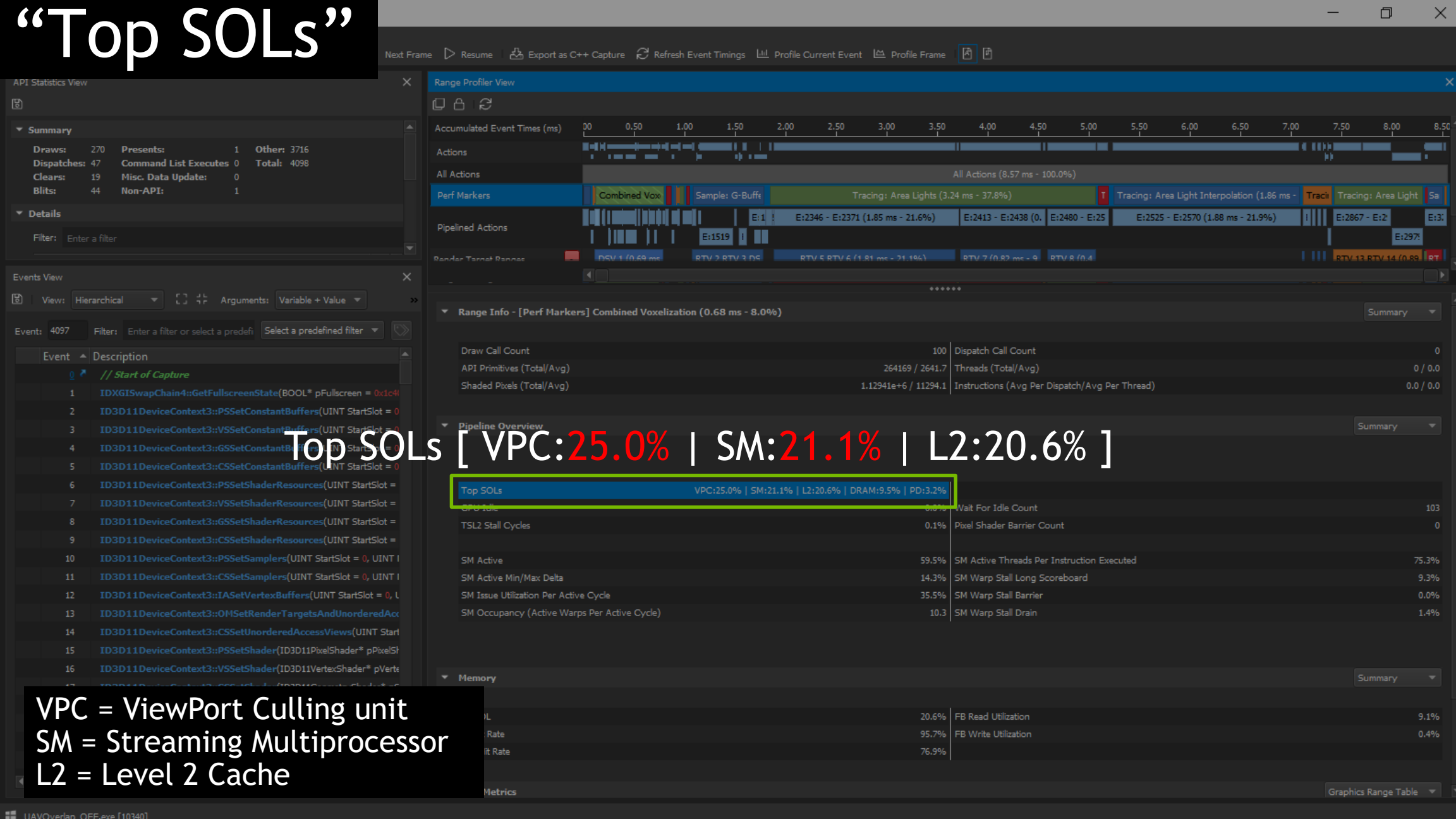
CPU Limited?



GPU Idle: 0.0%

➔ Not CPU limited at all

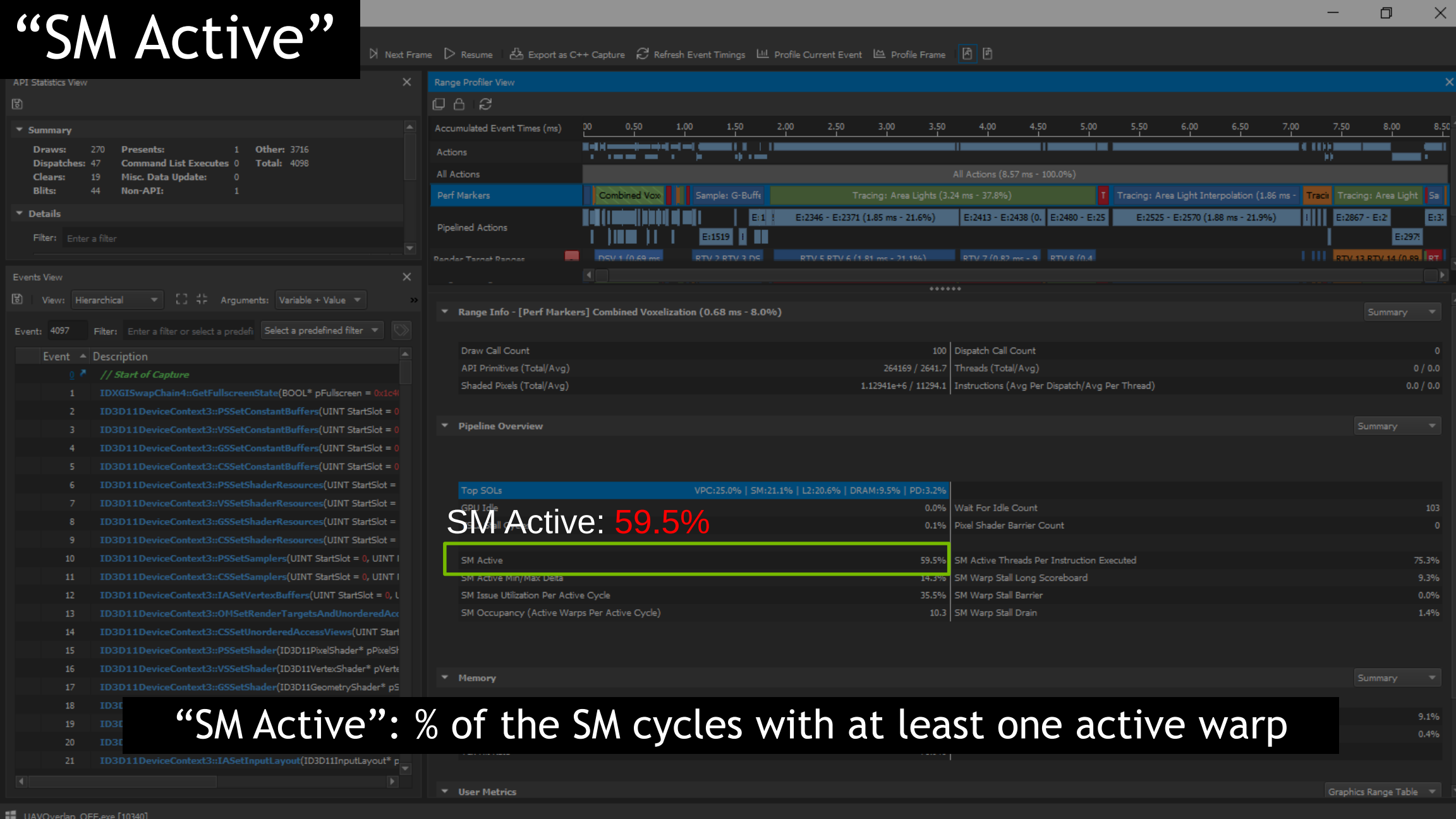
“Top SOLs”



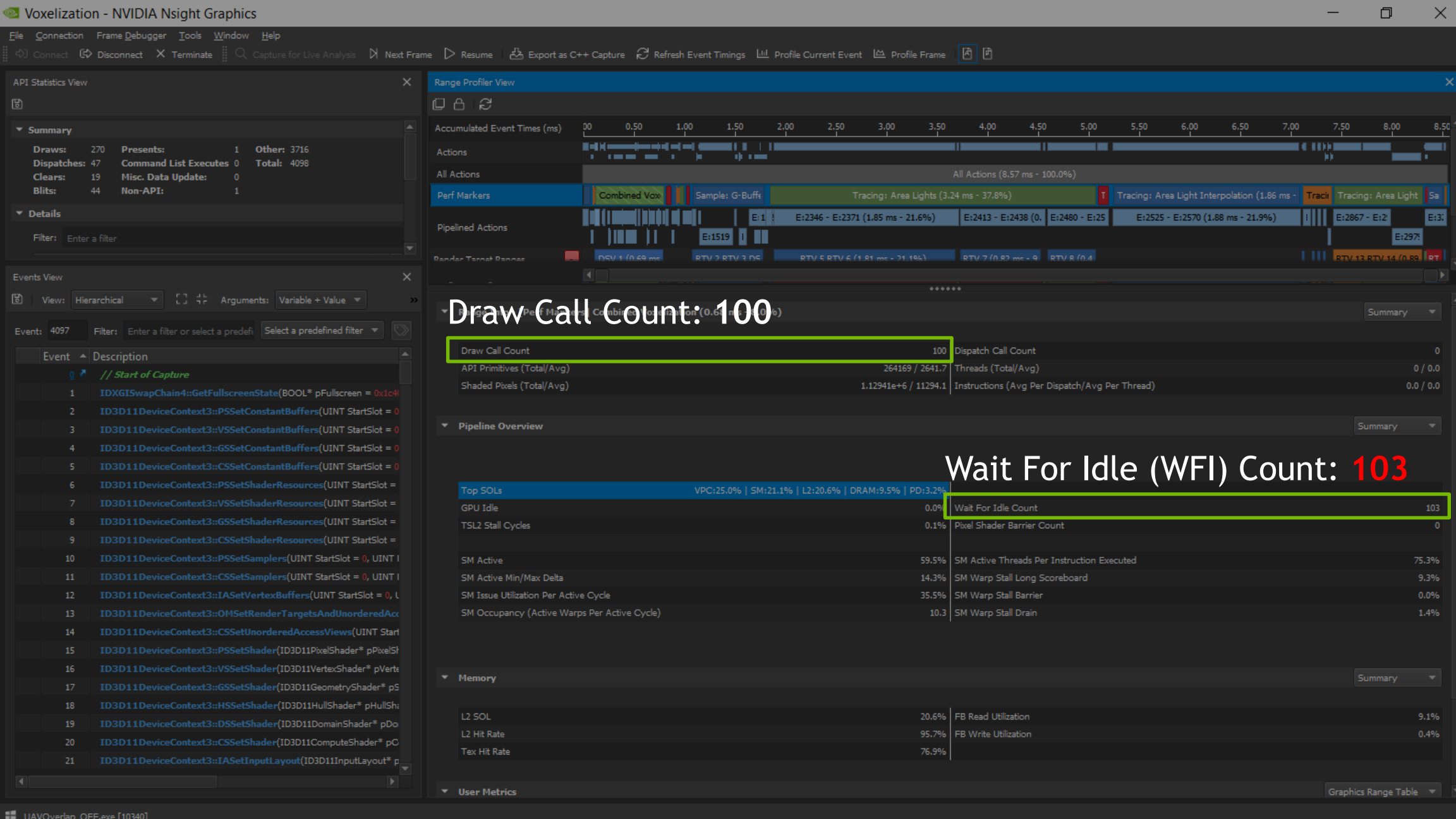
Top SOLs [VPC:25.0% | SM:21.1% | L2:20.6%]

VPC = ViewPort Culling unit
SM = Streaming Multiprocessor
L2 = Level 2 Cache

“SM Active”

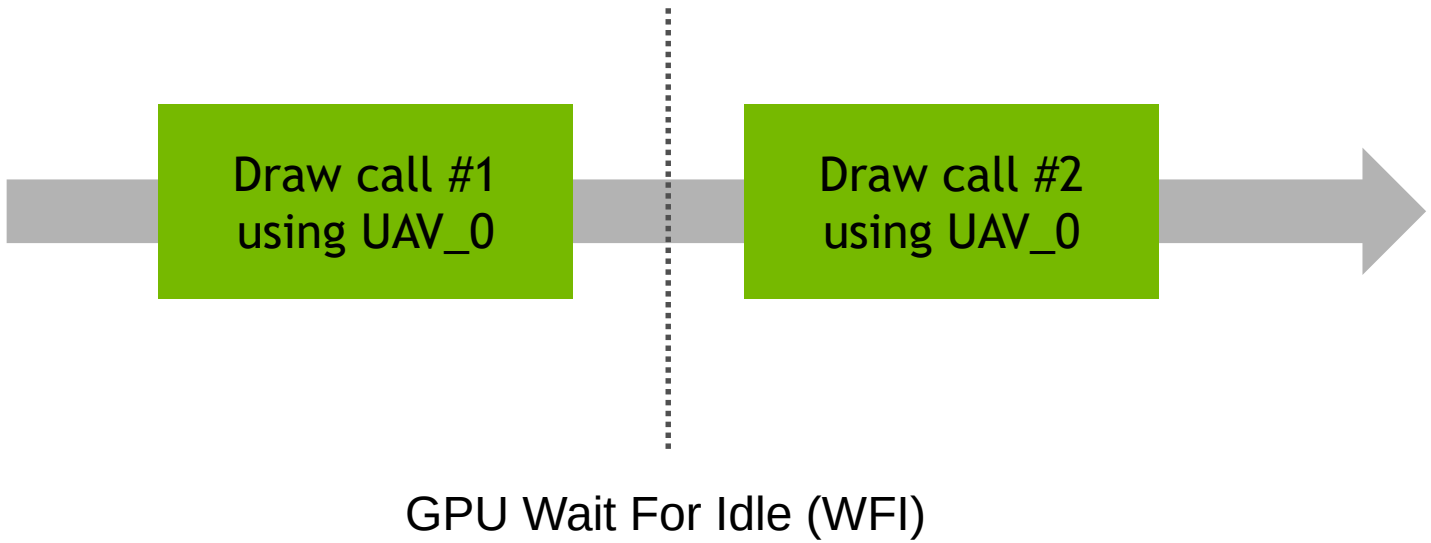


“SM Active”: % of the SM cycles with at least one active warp


Voxelization - NVIDIA Nsight Graphics
File Connection Frame Debugger Tools Window Help
Connect Disconnect Terminate Capture for Live Analysis Next Frame Resume Export as C++ Capture Refresh Event Timings Profile Current Event Profile Frame
API Statistics View
Summary
Draws: 270 Presents: 1 Other: 3716
Dispatches: 47 Command List Executes 0 Total: 4098
Clears: 19 Misc. Data Update: 0
Blits: 44 Non-API: 1
Details
Filter: Enter a filter
Range Profiler View
Accumulated Event Times (ms)
Actions
All Actions
All Actions (8.57 ms - 100.0%)
Perf Markers
Combined Vox
Sample: G-Buffe
Tracing: Area Lights (3.24 ms - 37.8%)
Tracing: Area Light Interpolation (1.86 ms -
Tracii
Tracing: Area Light
Pipelined Actions
E:2346 - E:2371 (1.85 ms - 21.6%)
E:2413 - E:2438 (0.
E:2480 - E:25
E:2525 - E:2570 (1.88 ms - 21.9%)
E:2867 - E:2
E:1519
Events View
View: Hierarchical
Arguments: Variable + Value
Event: 4097
Filter: Enter a filter or select a predefi
Select a predefined filter
Event
Description
0 // Start of Capture
1 IDXGISwapChain4::GetFullscreenState(BOOL* pFullscreen = 0x1c4
2 ID3D11DeviceContext3::PSSetConstantBuffers(UINT StartSlot = 0
3 ID3D11DeviceContext3::VSSetConstantBuffers(UINT StartSlot = 0
4 ID3D11DeviceContext3::GSSetConstantBuffers(UINT StartSlot = 0
5 ID3D11DeviceContext3::CSSetConstantBuffers(UINT StartSlot = 0
6 ID3D11DeviceContext3::PSSetShaderResources(UINT StartSlot =
7 ID3D11DeviceContext3::VSSetShaderResources(UINT StartSlot =
8 ID3D11DeviceContext3::GSSetShaderResources(UINT StartSlot =
9 ID3D11DeviceContext3::CSSetShaderResources(UINT StartSlot =
10 ID3D11DeviceContext3::PSSetSamplers(UINT StartSlot = 0, UINT
11 ID3D11DeviceContext3::CSSetSamplers(UINT StartSlot = 0, UINT
12 ID3D11DeviceContext3::IASetVertexBuffers(UINT StartSlot = 0,
13 ID3D11DeviceContext3::OMSetRenderTargetsAndUnorderedAcc
14 ID3D11DeviceContext3::CSSetUnorderedAccessViews(UINT Start
15 ID3D11DeviceContext3::PSSetShader(ID3D11PixelShader* pPixelSh
16 ID3D11DeviceContext3::VSSetShader(ID3D11VertexShader* pVerte
17 ID3D11DeviceContext3::GSSetShader(ID3D11GeometryShader* pS
18 ID3D11DeviceContext3::HSSetShader(ID3D11HullShader* pHullSha
19 ID3D11DeviceContext3::DSSetShader(ID3D11DomainShader* pDo
20 ID3D11DeviceContext3::CSSetShader(ID3D11ComputeShader* pC
21 ID3D11DeviceContext3::IASetInputLayout(ID3D11InputLayout* p
Draw Call Count: 100
Summary
Draw Call Count 100
API Primitives (Total/Avg) 264169 / 2641.7
Shaded Pixels (Total/Avg) 1.12941e+6 / 11294.1
Dispatch Call Count 0
Threads (Total/Avg) 0 / 0.0
Instructions (Avg Per Dispatch/Avg Per Thread) 0.0 / 0.0
Pipeline Overview
Summary
Wait For Idle (WFI) Count: 103
Top SOLs VPC:25.0% | SM:21.1% | L2:20.6% | DRAM:9.5% | PD:3.2%
GPU Idle 0.0%
TSL2 Stall Cycles 0.1%
SM Active 59.5%
SM Active Min/Max Delta 14.3%
SM Issue Utilization Per Active Cycle 35.5%
SM Occupancy (Active Warps Per Active Cycle) 10.3
Wait For Idle Count 103
Pixel Shader Barrier Count 0
SM Active Threads Per Instruction Executed 75.3%
SM Warp Stall Long Scoreboard 9.3%
SM Warp Stall Barrier 0.0%
SM Warp Stall Drain 1.4%
Memory
Summary
L2 SOL 20.6%
L2 Hit Rate 95.7%
Tex Hit Rate 76.9%
FB Read Utilization 9.1%
FB Write Utilization 0.4%
User Metrics
Graphics Range Table
UAVOverlap_OFF.exe [10340]

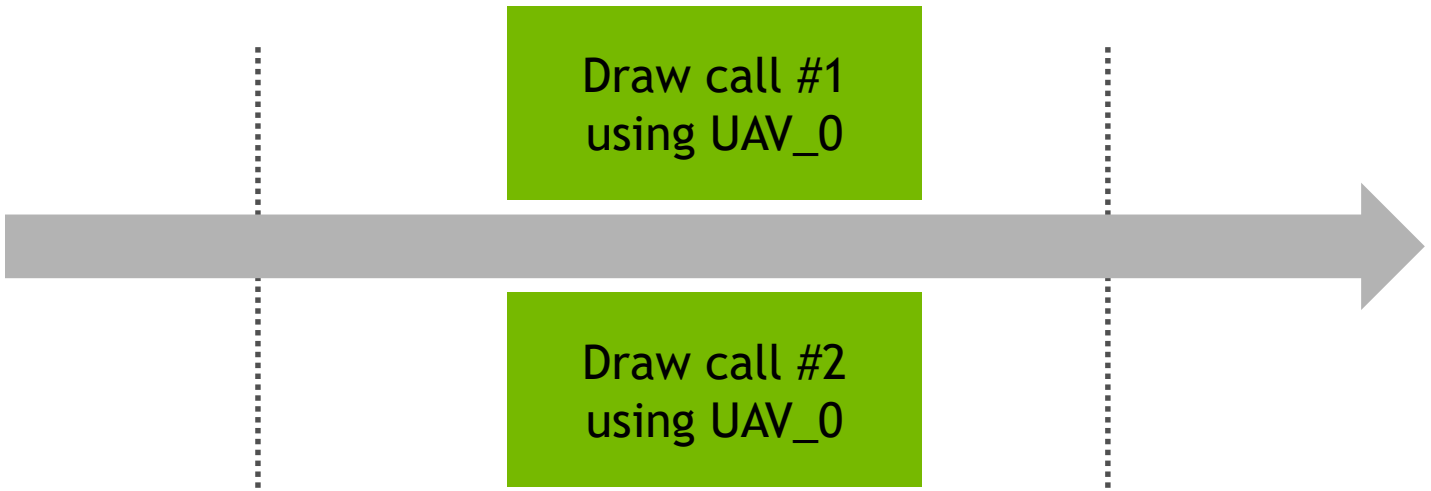
DX11 Driver Behavior

By default: Serialize Draw calls with bound UAV in common



DX11 Driver Behavior

Optimized: Concurrent Draw Calls



UAV-Overlap Optimization

Add NvAPI_D3D11_{Begin,End}UAVOverlap

	BEFORE	AFTER	RATIO
WFI Count	103	3	
Top SOLs	**VPC:25.0%** SM:21.1% L2:20.6%	**VPC:52.3%** SM:44.3% L2:42.6%	**VPC: 2.1x** SM: 2.1x L2: 2.1x
SM Active%	59.1%	95.1%	1.6x
GPU Elapsed Time	0.69 ms	0.38 ms	1.8x Gain

The Peak-Perf% Analysis Method

BEFORE: Top SOLs: [VPC:25.0% | SM:21.1% | L2:20.6%]
AFTER: Top SOLs: [VPC:52.3% | SM:44.3% | L2:42.6%]

For each “Top SOL%” unit:

1. If SOL% > 80% ➔ (A) try removing work from this unit
2. If SOL% < 60% ➔ (B) try increasing the SOL% of this unit
   - **By removing “idle cycles” (GPU unit is not doing any work for a % of the time)**
   - By removing “stall cycles” (GPU unit has internal inefficiencies)
   - By avoiding “slow paths” if possible (e.g. avoiding 32-bit index buffers, and avoiding FP32x4 texture formats).
3. If SOL% in [60,80], do both (A) and (B)

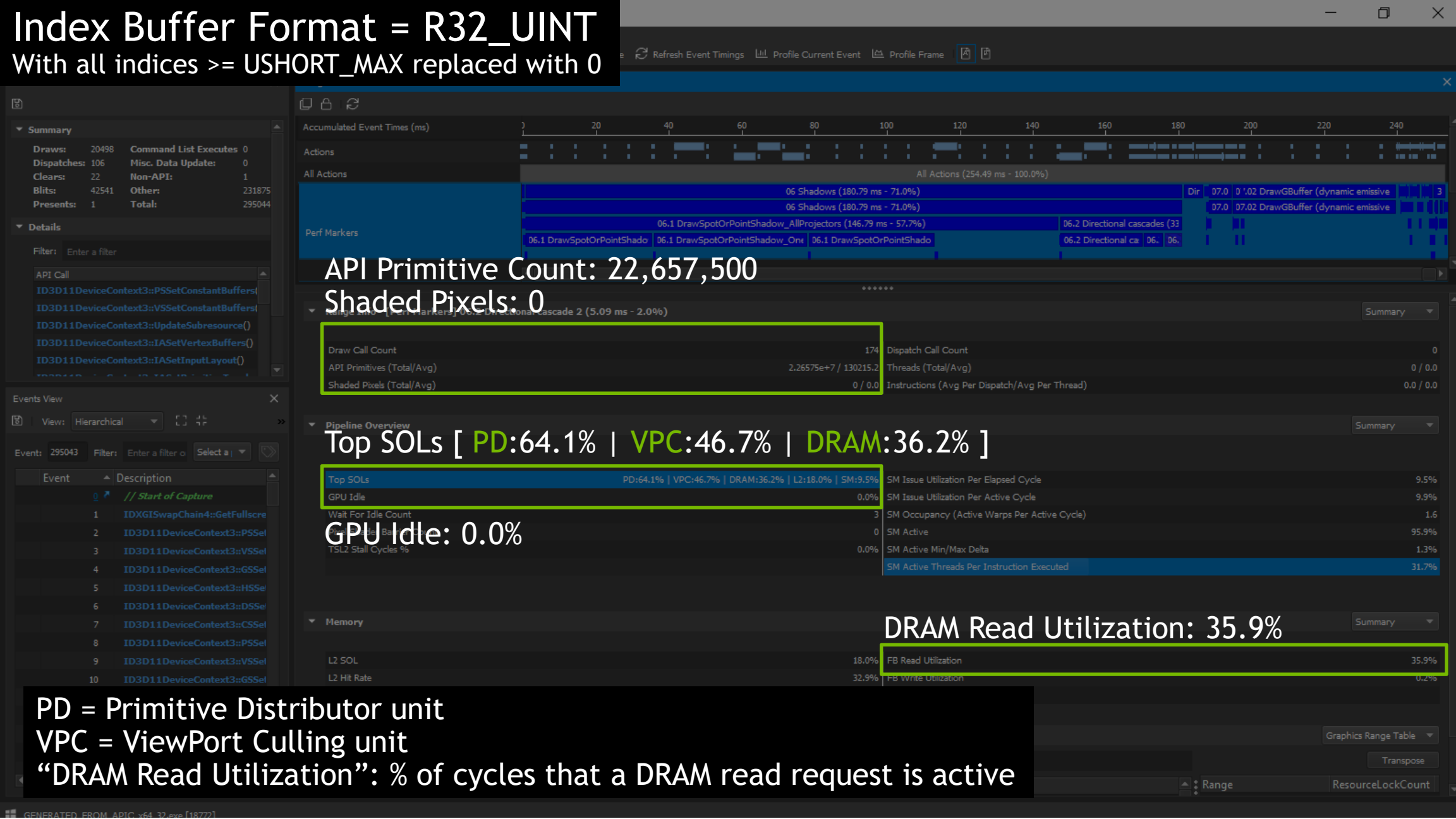
Example Workload:

Drawing Tiny Triangles

GPU: GTX 1080

Index Buffer Format = R32_UINT

With all indices >= USHORT_MAX replaced with 0



PD = Primitive Distributor unit

VPC = ViewPort Culling unit

“DRAM Read Utilization”: % of cycles that a DRAM read request is active

Index-Buffer Format Optimization

32->16 bits per index

	BEFORE	AFTER	RATIO
Top SOLs	**PD:64.1%** VPC:46.7% DRAM:36.2%	**PD:80.5%** VPC:58.7% DRAM:28.5%	**PD:1.3x** VPC:1.3x DRAM: 0.8x
DRAM Read Utilization	36%	28%	0.78x
GPU Elapsed Time	5.09 ms	2.37 ms	**2.1x Gain**

The Peak-Perf% Analysis Method

BEFORE: Top SOLs: [PD:64.1% | VPC:46.7% | DRAM:36.2%]
AFTER: Top SOLs: [PD:80.5% | VPC:58.7% | DRAM:28.5%]

For each “Top SOL%” unit:

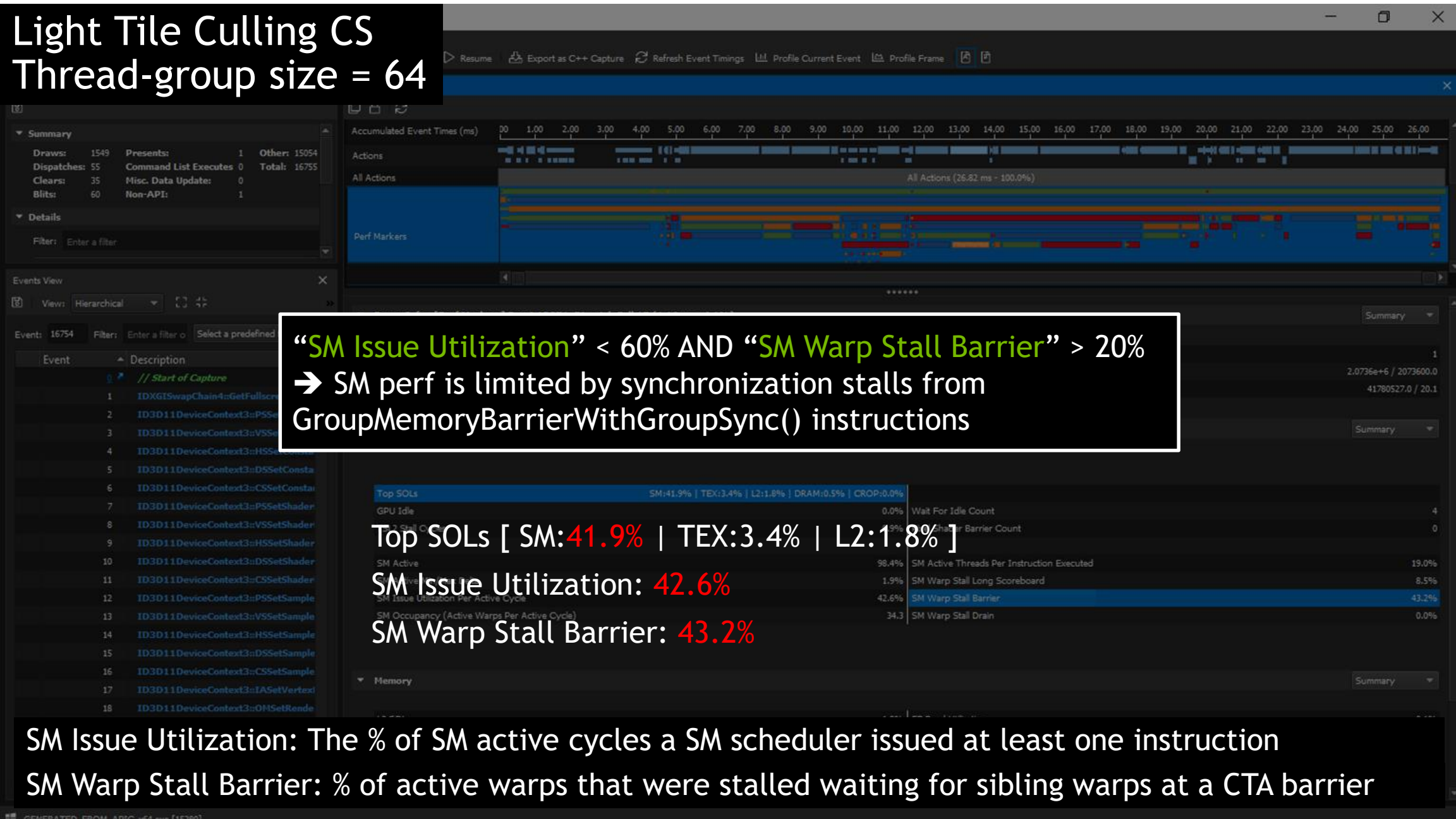
1. **If SOL% > 80% ➔ (A) try removing work from this unit**
2. **If SOL% < 60% ➔ (B) try increasing the SOL% of this unit**
   - By removing “idle cycles” (GPU unit is not doing any work for a % of the time)
   - By removing “stall cycles” (GPU unit has internal inefficiencies)
   - By avoiding “slow paths” if possible (e.g. 32-bit index buffers, and FP32x4 textures)
3. **If SOL% in [60,80], do both (A) and (B)**

Example Workload:

Light-Tile Culling Compute Shader

GPU: GTX 1080

Light Tile Culling CS
Thread-group size = 64



"SM Issue Utilization" < 60% AND "SM Warp Stall Barrier" > 20%
➔ SM perf is limited by synchronization stalls from GroupMemoryBarrierWithGroupSync() instructions

Top SOLs [SM:41.9% | TEX:3.4% | L2:1.8%]

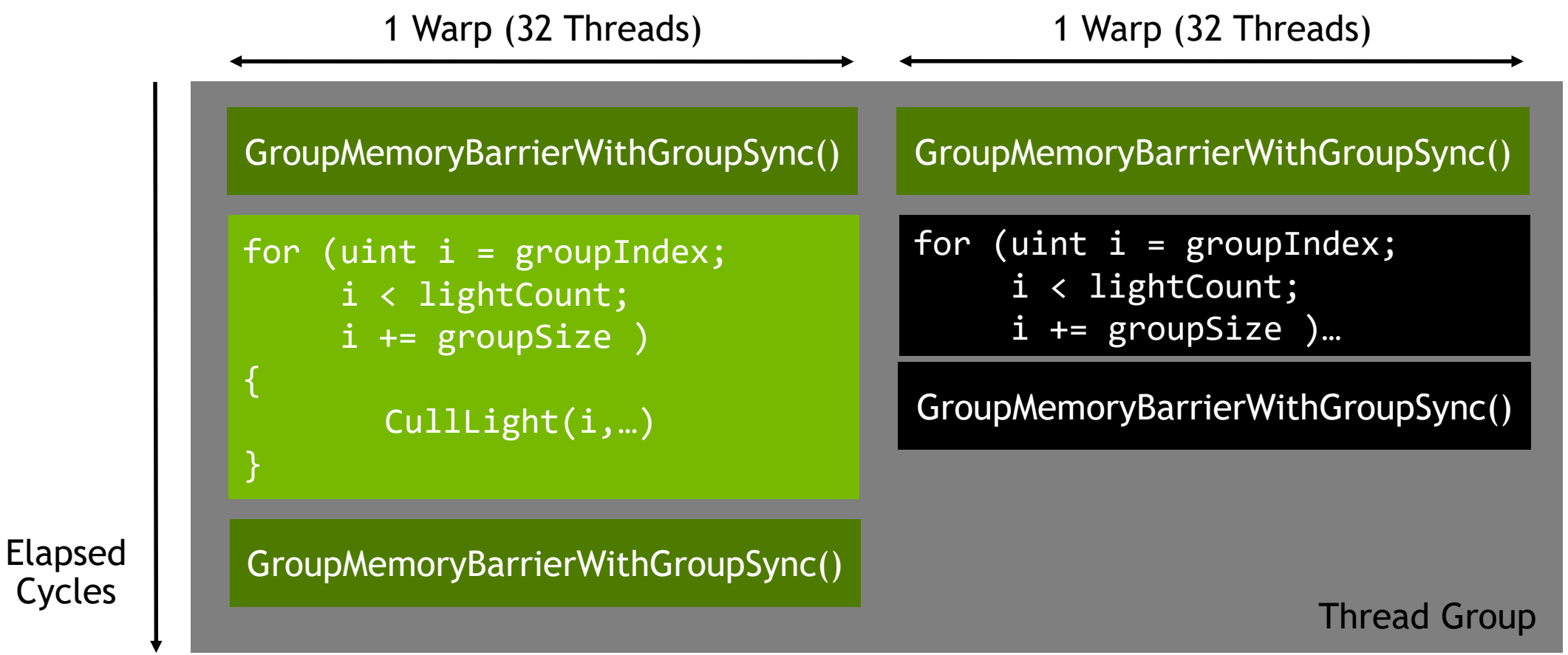
SM Issue Utilization: 42.6%

SM Warp Stall Barrier: 43.2%

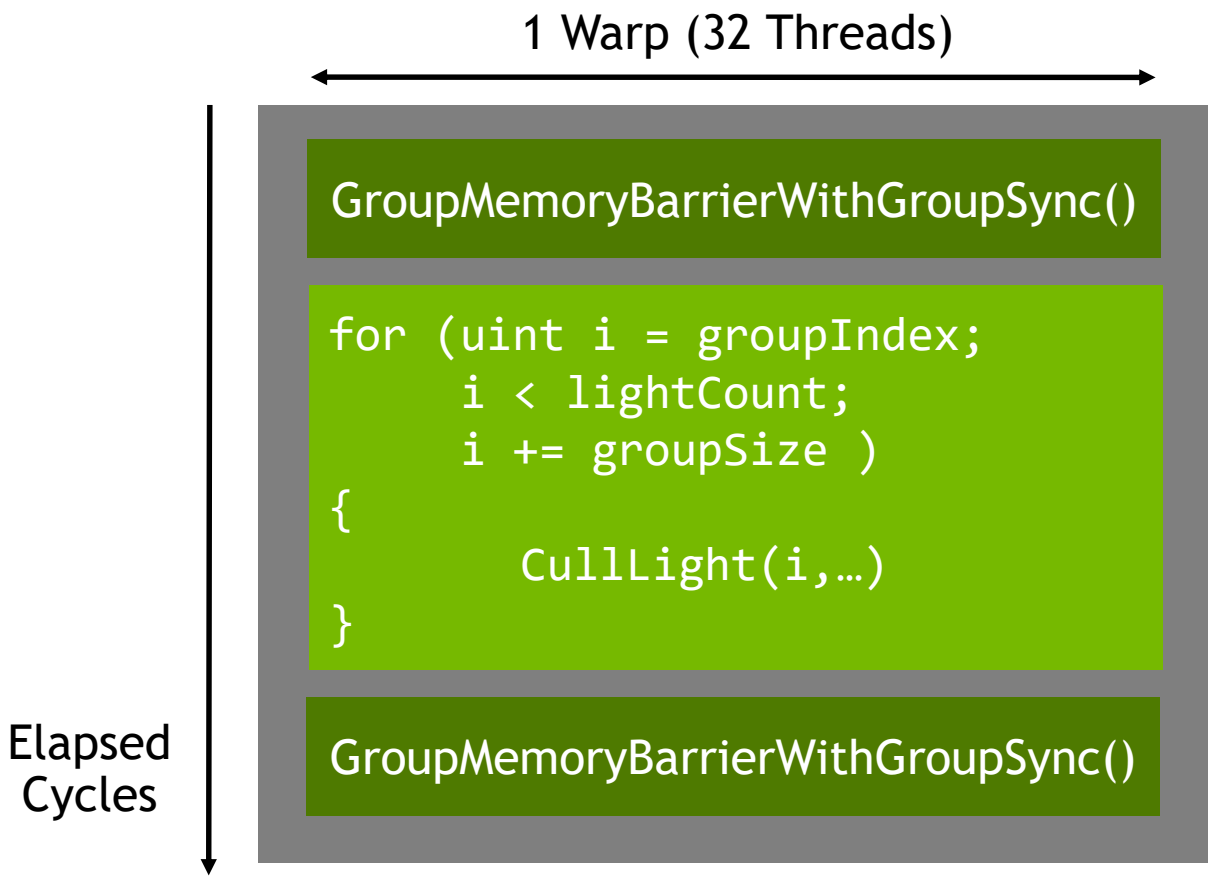
SM Issue Utilization: The % of SM active cycles a SM scheduler issued at least one instruction

SM Warp Stall Barrier: % of active warps that were stalled waiting for sibling warps at a CTA barrier

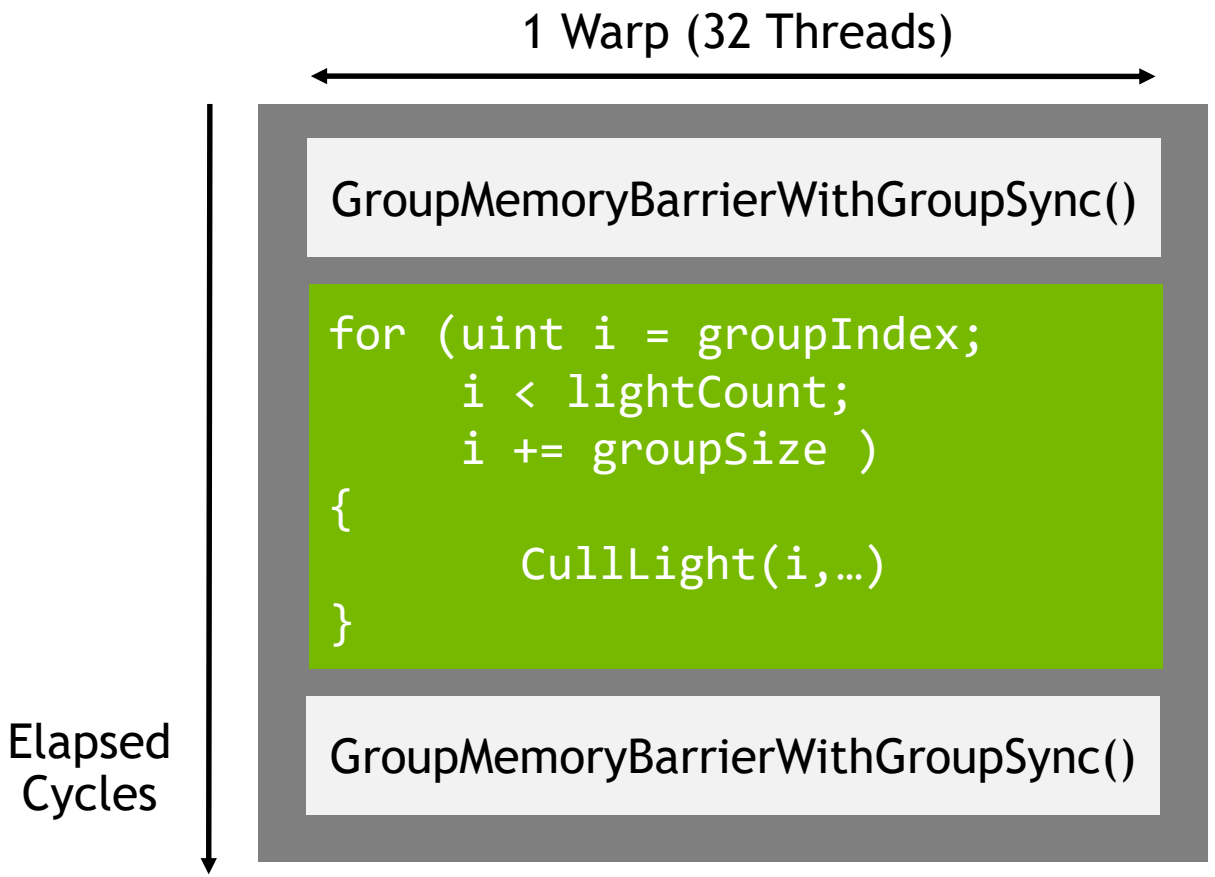
BEFORE: 2-Warp Thread Groups



AFTER: 1-Warp Thread Groups



AFTER: 1-Warp Thread Groups



For **single-warp thread groups**, barrier instructions are **free** on NVIDIA GPUs.

Thread-Group Size Reduction:

64 threads -> 32 threads

	BEFORE	AFTER	RATIO
Top SOL	SM:41.9%	SM:73.7%	SM:1.76x
SM Issue Utilization	42.6%	76.6%	1.80x
SM Warp Stall on Barriers	43.2%	0.0%	0.0x
SM Occupancy (Active Warps)	34.3	31.2	0.91x
GPU Elapsed Time	1.10 ms	0.33 ms	**3.3x Gain**

The Peak-Perf% Analysis Method

BEFORE: Top SOLs: [SM:41.9% | TEX:3.4% | L2:1.8%]
AFTER: Top SOLs: [SM:73.7% | TEX:4.9% | L2:4.2%]

For each “Top SOL%” unit (from high to low SOL%):

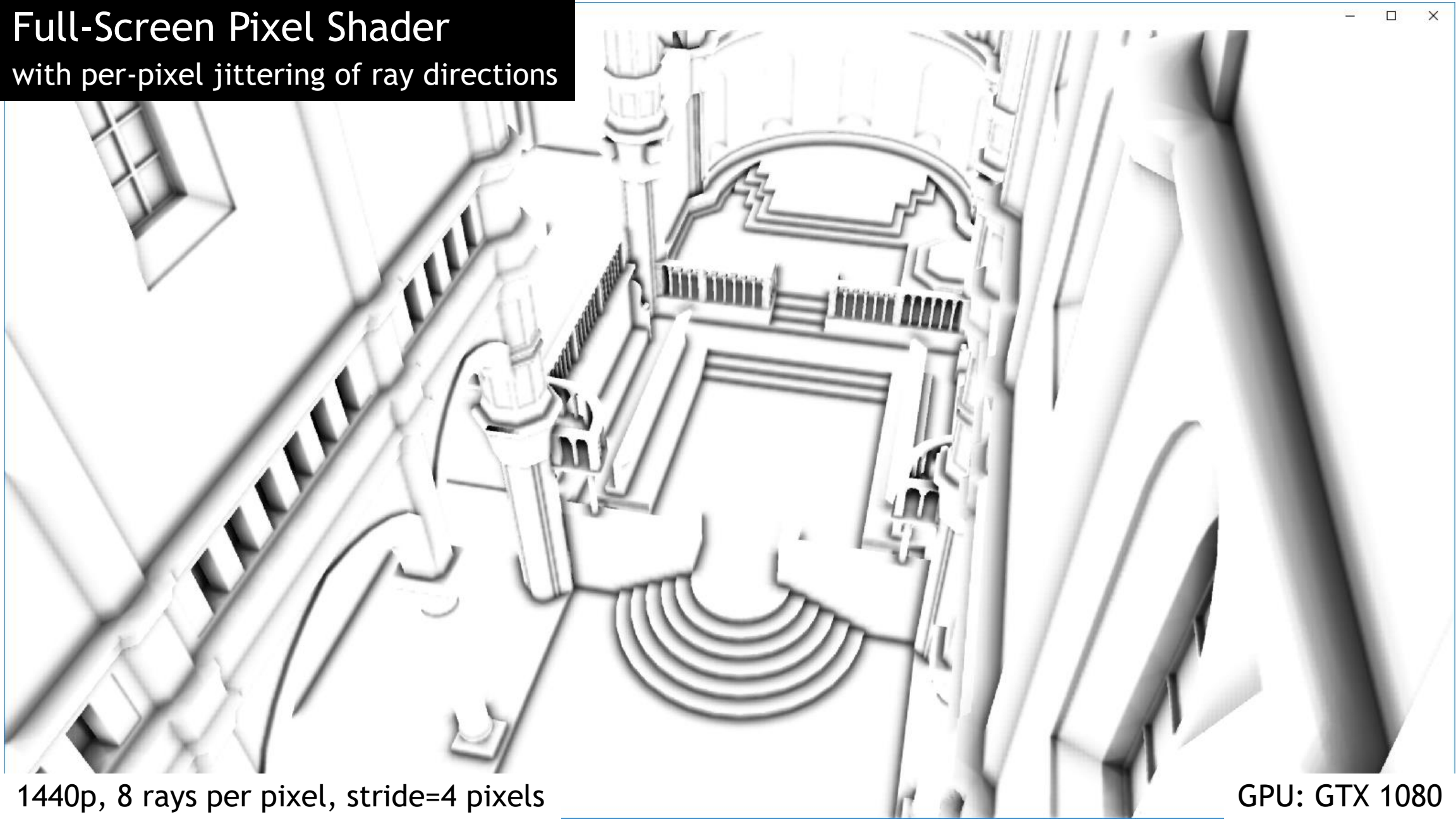
1. If SOL% > 80% ➔ (A) try removing work from this unit
2. **If SOL% < 60% ➔ (B) try increasing the SOL% of this unit**
   - By removing “idle cycles” (GPU unit is not doing any work for a % of the time)
   - By removing “stall cycles”: SM Warp Stalls on Shared-Memory Barriers
   - By avoiding “slow paths” if possible (e.g. 32-bit index buffers, and FP32x4 textures)
3. If SOL% in [60,80], do both (A) and (B)

Example Workload:

Ray-Marched SSAO

Full-Screen Pixel Shader

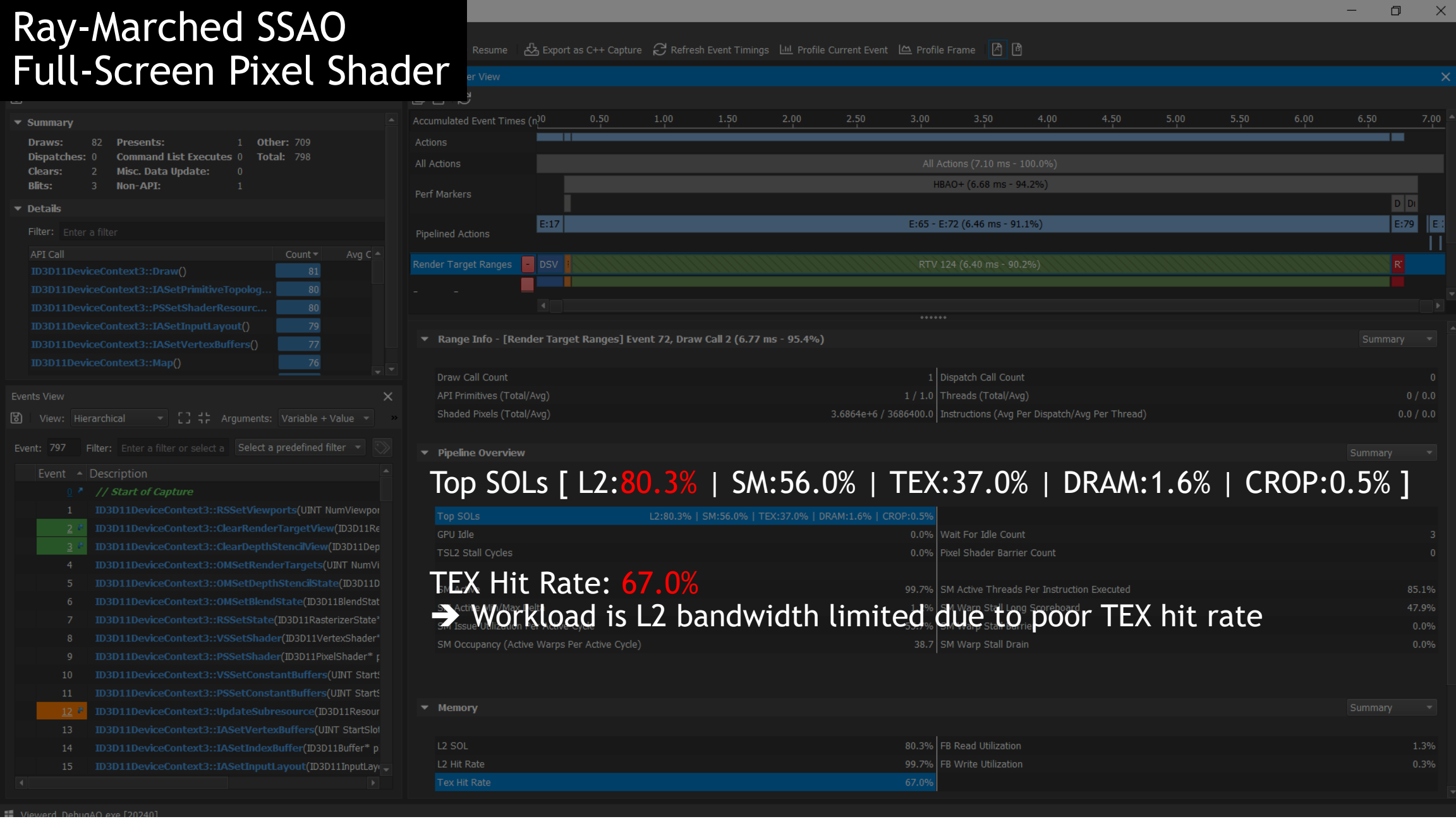
with per-pixel jittering of ray directions



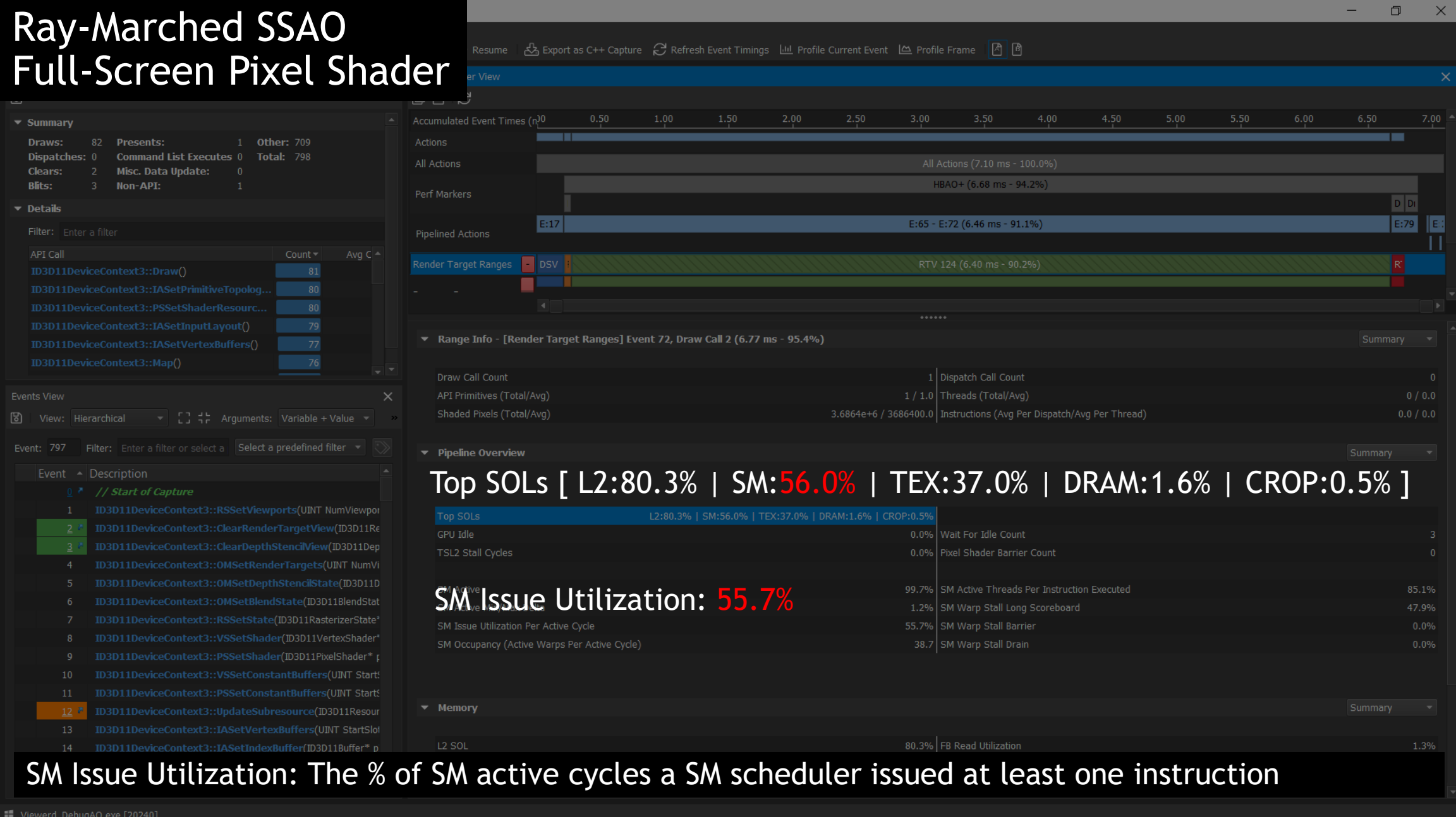
1440p, 8 rays per pixel, stride=4 pixels

GPU: GTX 1080

Ray-Marched SSAO Full-Screen Pixel Shader

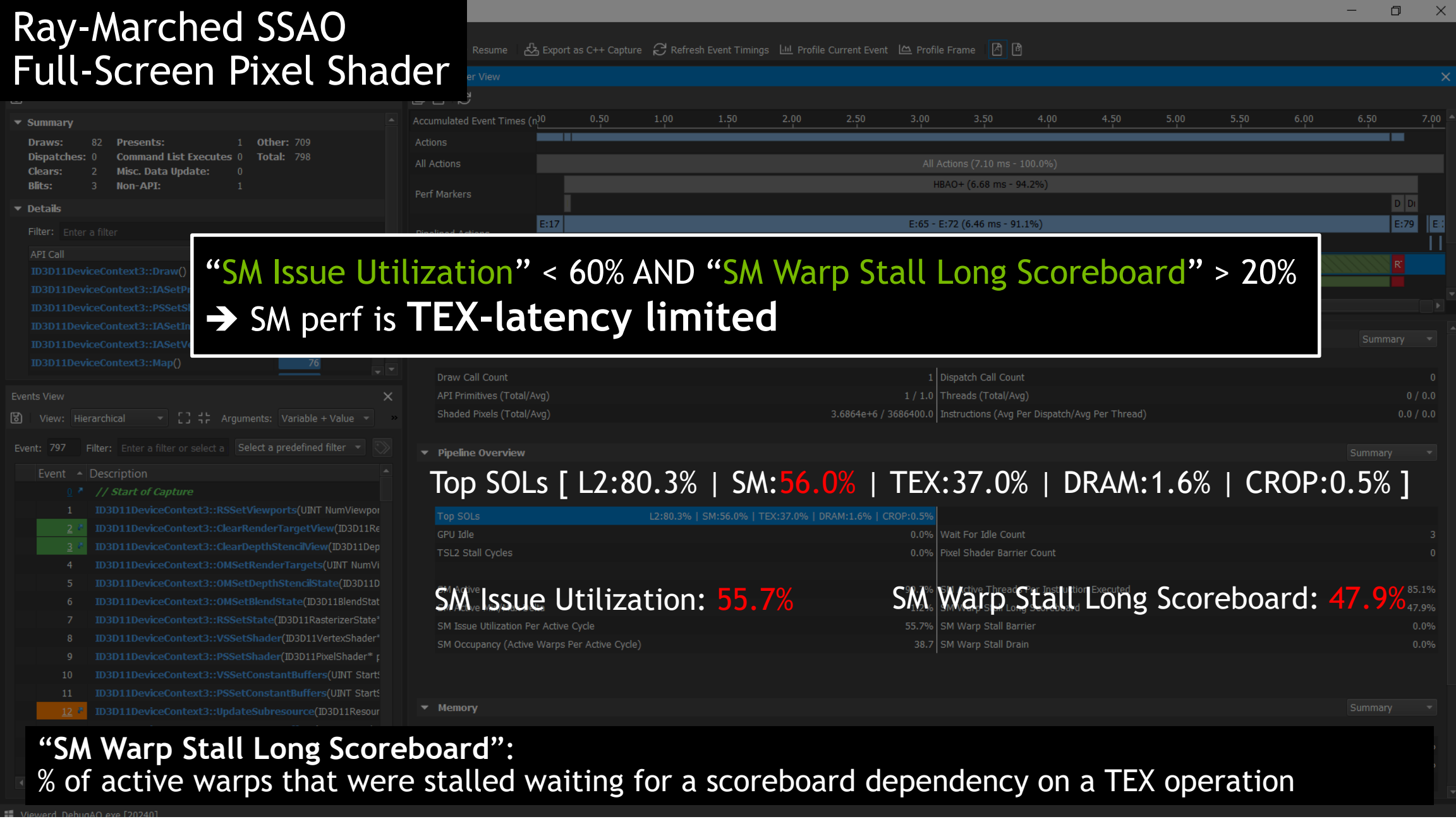


Ray-Marched SSAO Full-Screen Pixel Shader



SM Issue Utilization: The % of SM active cycles a SM scheduler issued at least one instruction

Ray-Marched SSAO Full-Screen Pixel Shader



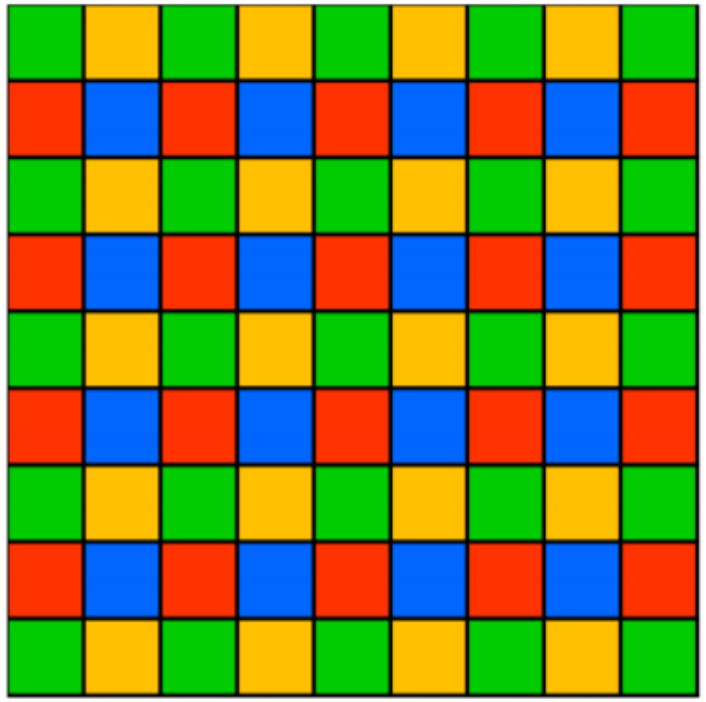
Our Solution:

"Interleaved Rendering"

Render each sampling pattern **separately**, using **downsampled** input textures

Assumption:

Interleaved Sampling Patterns

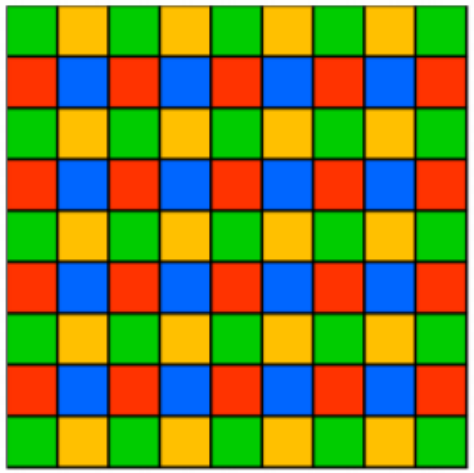


NxN sampling patterns interleaved on screen

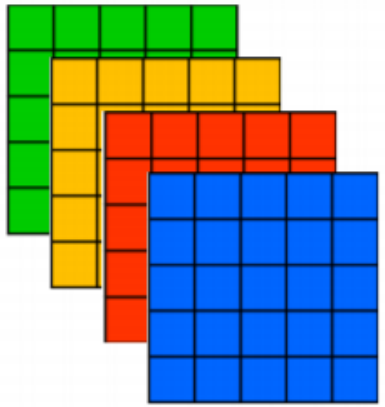
Typical sampling strategy for SSAO, SSDO, SSR, etc.

Per-pixel jitter seed fetched from a tiled “jitter texture”

STEP 1: Deinterleave Input



1 Draw call
with 4xMRTs



Full-Resolution Input Texture

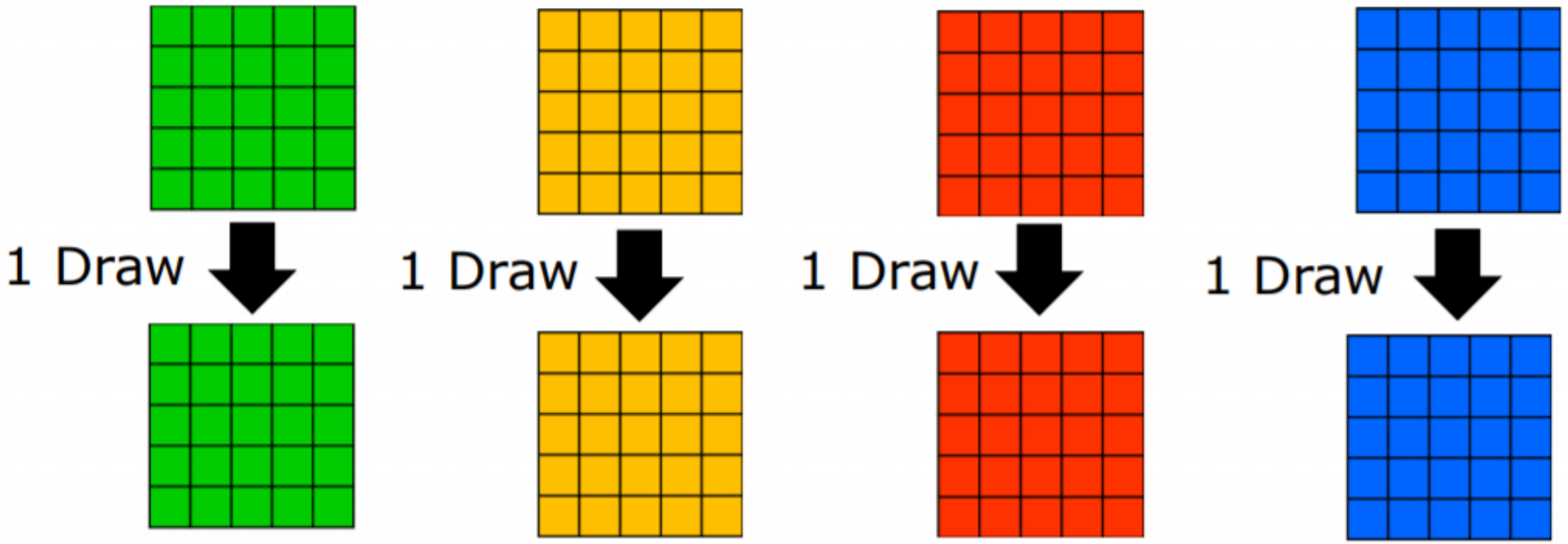
Width = W
Height = H

Half-Resolution 2D Texture Array

Width = iDivUp(W,2)
Height = iDivUp(H,2)

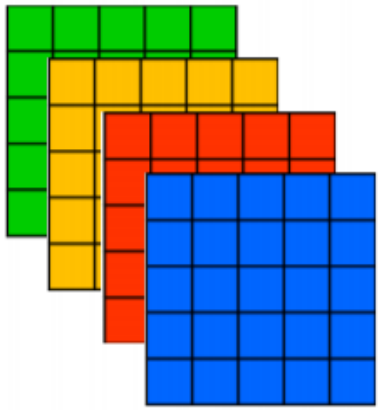
STEP 2: Jitter-Free Sampling

Input: Texture Array A (slices 0,1,2,3)



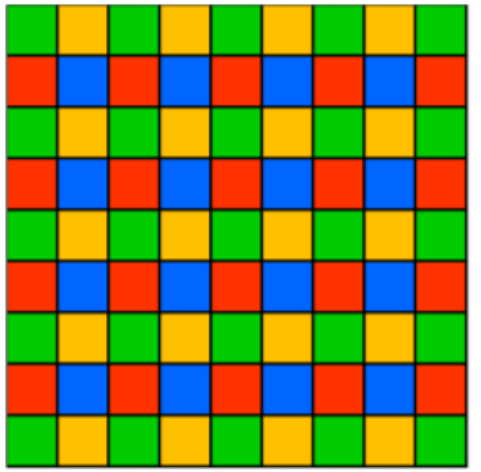
Output: Texture Array B (slices 0,1,2,3)

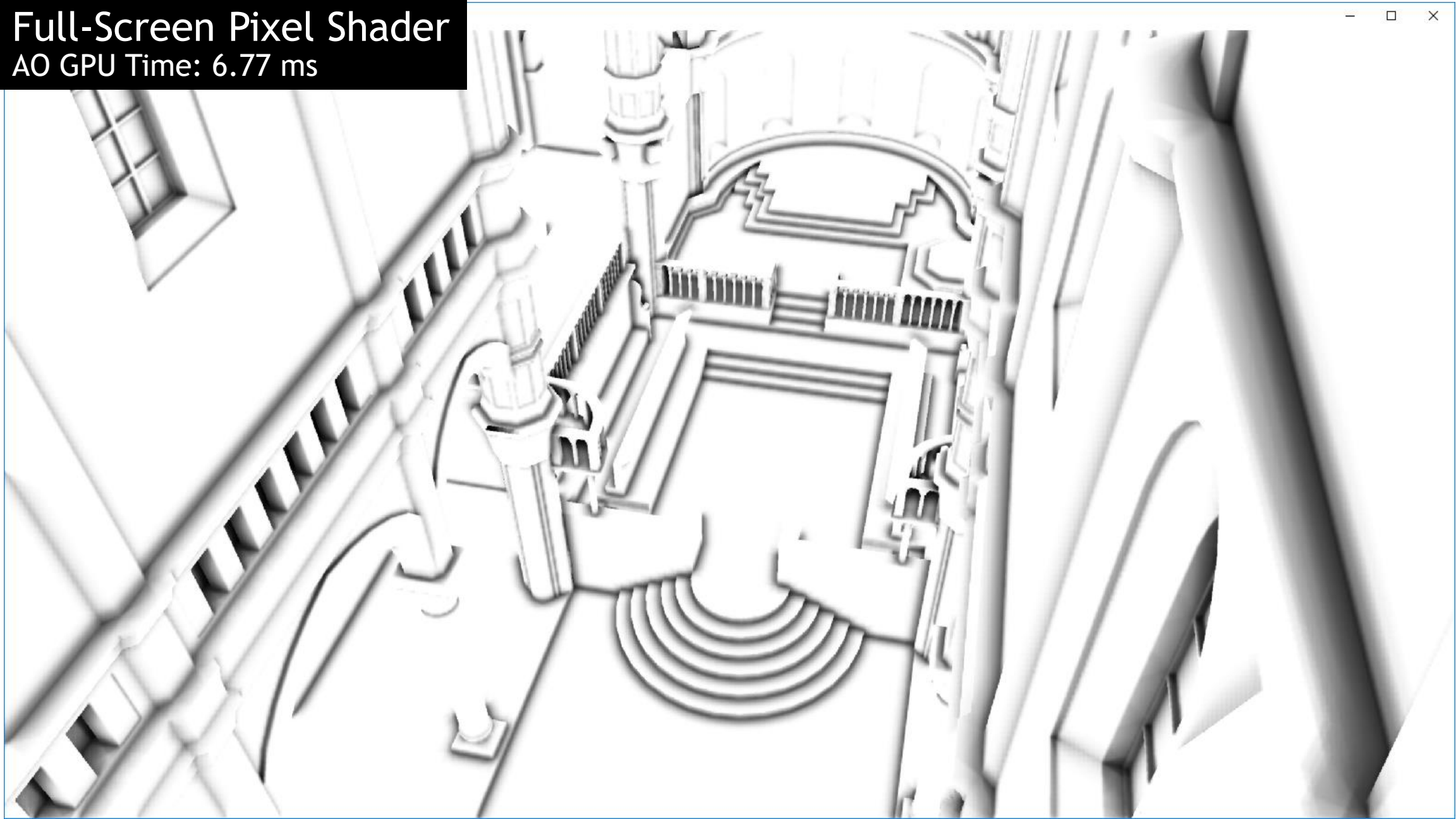
STEP 3: Interleave Results



1 Draw call

With 1 Tex2DArray fetch per pixel




Full-Screen Pixel Shader
AO GPU Time: 6.77 ms

Interleaved Rendering (3 Steps)

AO GPU Time: 0.10 + 5.04 + 0.08 = 5.22 ms [27% gain]

Interleaved Rendering Optimization

AO KERNEL	BEFORE	AFTER	RATIO
Top SOLs	**L2:80.3%** SM:56.0% TEX:37.0%	**L2:11.3%** SM:78.8% TEX:32.4%	**L2:0.14x** SM:1.4x TEX:0.9x
TEX Hit Rate	67%	93%	1.4x
SM Issue Utilization	56%	73%	1.3x
SM Warp Stall Long Scoreboard	48%	28%	0.6x

2x Partial Loop Unrolling

Before

```
do
{
    // Fetch Sample_1
    // Calculate RayXYZ_1
    // Advance Ray
} while (
    ...
    );
```

After

```
do
{
    // Fetch Sample_1
    // Fetch Sample_2
    // Calculate RayXYZ_1
    // Advance Ray

    // Calculate RayXYZ_2
    // Advance Ray
} while (
...
);
```

2x Partial Loop Unrolling

	BEFORE	AFTER	RATIO
Top SOLs	SM:78.8% TEX:32.4% L2:11.3%	SM:88.6% TEX:37.4% L2:9.9%	SM:1.1x TEX:1.2x L2:0.9x
SM Issue Utilization	73%	84%	1.15x
SM Warp Stall on Long Scoreboard	28%	12%	0.43x
SM Occupancy (Active Warps)	39.0	33.8	0.87x
GPU Elapsed Time	5.04 ms	4.53 ms	11% Gain

The Peak-Perf% Analysis Method

BEFORE: Top SOLs: [L2:80.3% | SM:56.0% | TEX:37.0%]
AFTER: Top SOLs: [L2:9.9% | SM:88.6% | TEX:37.4%]

For each “Top SOL%” unit:

1. If SOL% > 80% ➔ (A) try removing work from this unit
   - **Reduce the number of TEX->L2 requests by improving the TEX hit rate**
2. If SOL% < 60% ➔ (B) try increasing the SOL% of this unit
   - By removing “idle cycles” (GPU unit is not doing any work for a % of the time)
   - By removing “stall cycles” (GPU unit has internal inefficiencies)
   - By avoiding “slow paths” if possible (e.g. 32-bit index buffers, and FP32x4 textures)
3. If SOL% in [60,80], do both (A) and (B)

The Peak-Perf% Analysis Method

BEFORE: Top SOLs: [L2:80.3% | SM:56.0% | TEX:37.0%]
AFTER: Top SOLs: [L2:9.9% | SM:88.6% | TEX:37.4%]

For each “Top SOL%” unit:

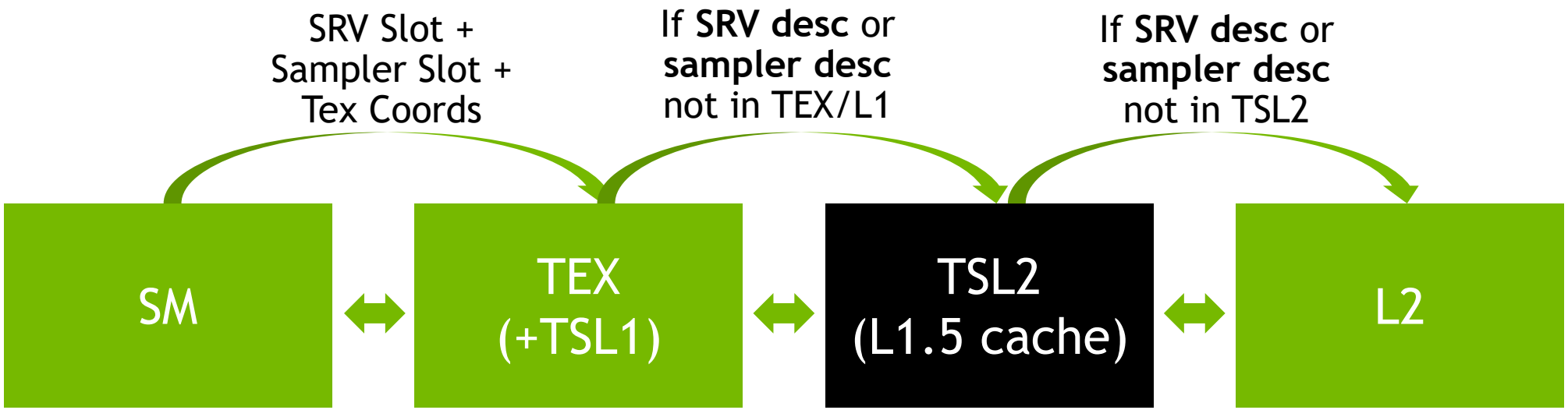
1. If SOL% > 80% ➔ (A) try removing work from this unit
2. If SOL% < 60% ➔ (B) try increasing the SOL% of this unit
   - By removing “idle cycles” (GPU unit is not doing any work for a % of the time)
   - **By removing “stall cycles”: SM Warp Stalls on TEX dependencies**
   - By avoiding “slow paths” if possible (e.g. 32-bit index buffers, and FP32x4 textures)
3. If SOL% in [60,80], do both (A) and (B)

DX12 Advanced Topic:

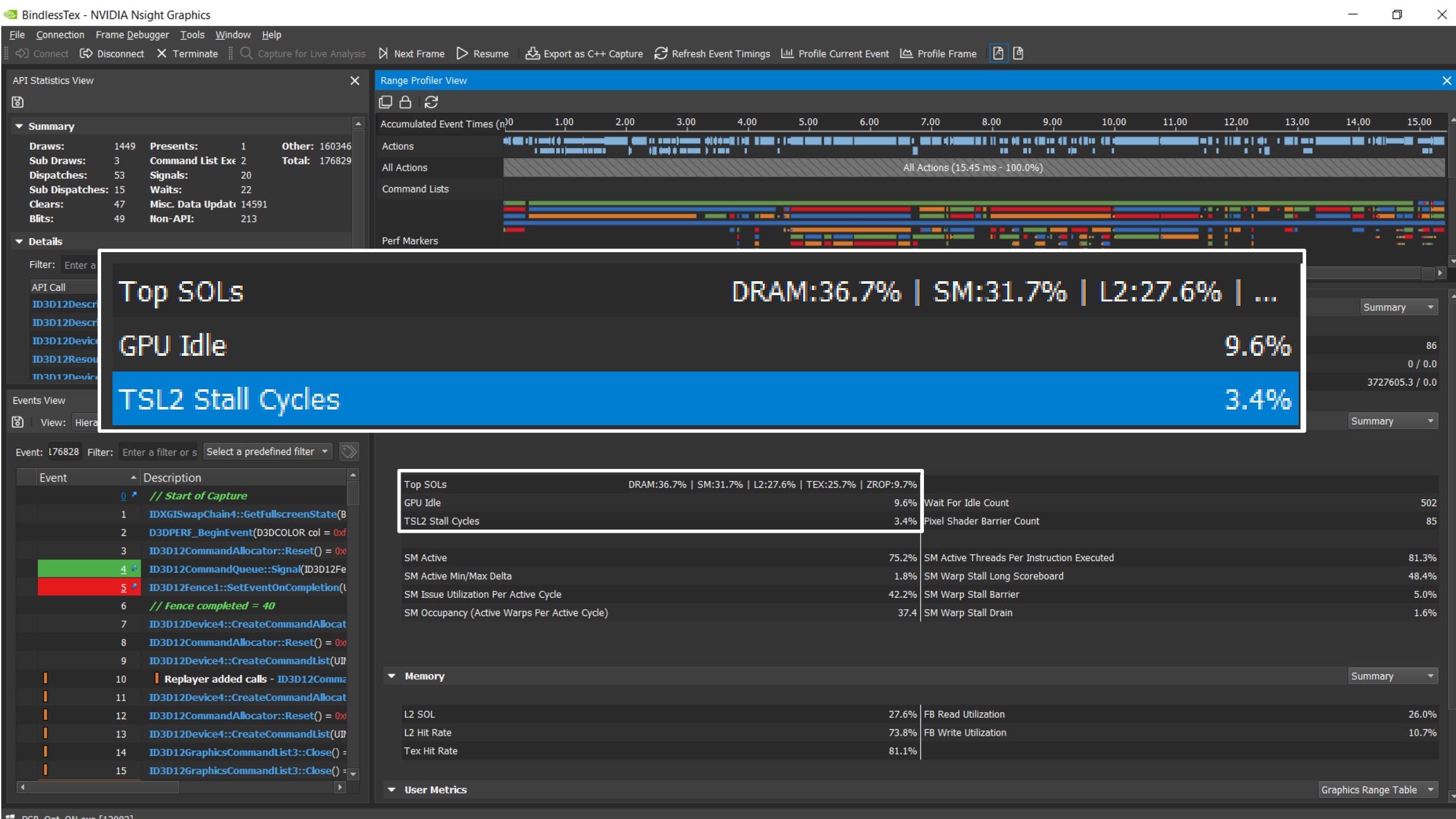
Binding SRV Descriptors

GPU: GTX 1080

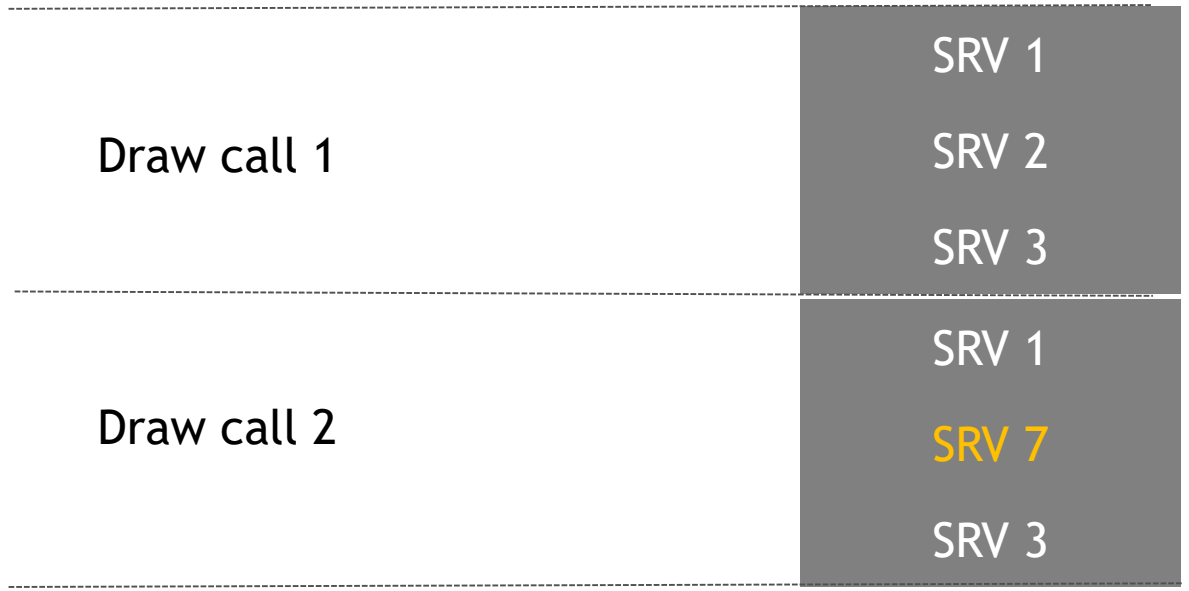
The TSL1 & TSL2 Caches



SRV descriptor contains texture metadata (type, dimensions, format, etc)

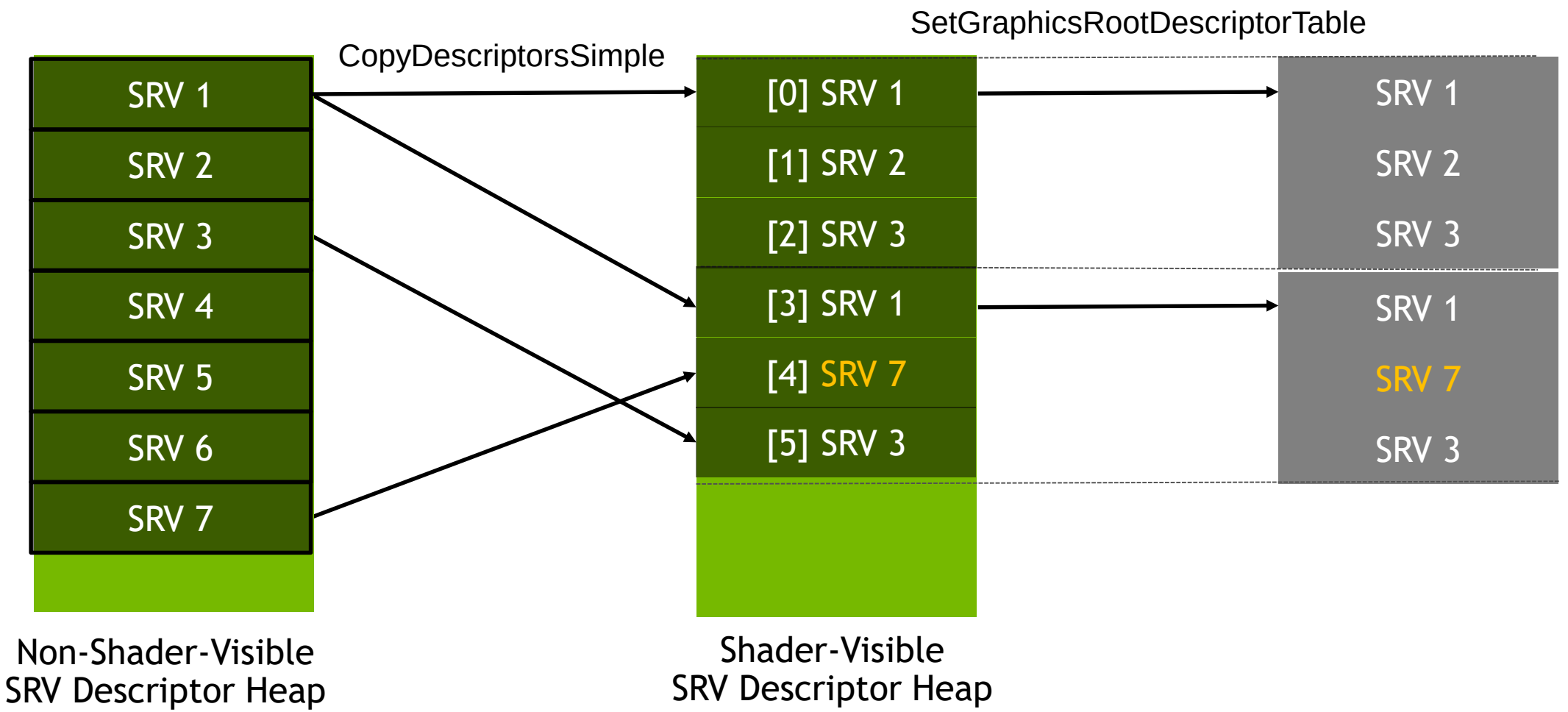

BindlessTex - NVIDIA Nsight Graphics
File Connection Frame Debugger Tools Window Help
Connect Disconnect Terminate Capture for Live Analysis Next Frame Resume Export as C++ Capture Refresh Event Timings Profile Current Event Profile Frame
API Statistics View
Summary
Draws: 1449 Presents: 1 Other: 160346
Sub Draws: 3 Command List Exe 2 Total: 176829
Dispatches: 53 Signals: 20
Sub Dispatches: 15 Waits: 22
Clears: 47 Misc. Data Updat 14591
Blits: 49 Non-API: 213
Details
Filter: Enter a
API Call
Range Profiler View
Accumulated Event Times (n)
0 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00 12.00 13.00 14.00 15.00
Actions
All Actions
All Actions (15.45 ms - 100.0%)
Command Lists
Perf Markers
Top SOLs DRAM:36.7% | SM:31.7% | L2:27.6% | ...
GPU Idle 9.6%
TSL2 Stall Cycles 3.4%
Summary
86
0 / 0.0
3727605.3 / 0.0
Events View
View: Hiera
Event: 176828 Filter: Enter a filter or s Select a predefined filter
Event Description
0 // Start of Capture
1 IDXGISwapChain4::GetFullscreenState(B
2 D3DPERF_BeginEvent(D3DCOLOR col = 0xf
3 ID3D12CommandAllocator::Reset() = 0x
4 ID3D12CommandQueue::Signal(ID3D12Fe
5 ID3D12Fence1::SetEventOnCompletion(
6 // Fence completed = 40
7 ID3D12Device4::CreateCommandAllocat
8 ID3D12CommandAllocator::Reset() = 0x
9 ID3D12Device4::CreateCommandList(UI
10 Replayer added calls - ID3D12Comma
11 ID3D12Device4::CreateCommandAllocat
12 ID3D12CommandAllocator::Reset() = 0x
13 ID3D12Device4::CreateCommandList(UI
14 ID3D12GraphicsCommandList3::Close() =
15 ID3D12GraphicsCommandList3::Close() =
Top SOLs DRAM:36.7% | SM:31.7% | L2:27.6% | TEX:25.7% | ZROP:9.7%
GPU Idle 9.6%
TSL2 Stall Cycles 3.4%
Wait For Idle Count 502
Pixel Shader Barrier Count 85
SM Active 75.2%
SM Active Min/Max Delta 1.8%
SM Issue Utilization Per Active Cycle 42.2%
SM Occupancy (Active Warps Per Active Cycle) 37.4
SM Active Threads Per Instruction Executed 81.3%
SM Warp Stall Long Scoreboard 48.4%
SM Warp Stall Barrier 5.0%
SM Warp Stall Drain 1.6%
Memory
Summary
L2 SOL 27.6%
L2 Hit Rate 73.8%
Tex Hit Rate 81.1%
FB Read Utilization 26.0%
FB Write Utilization 10.7%
User Metrics
Graphics Range Table

Typical DX12 SRV Binding Pattern

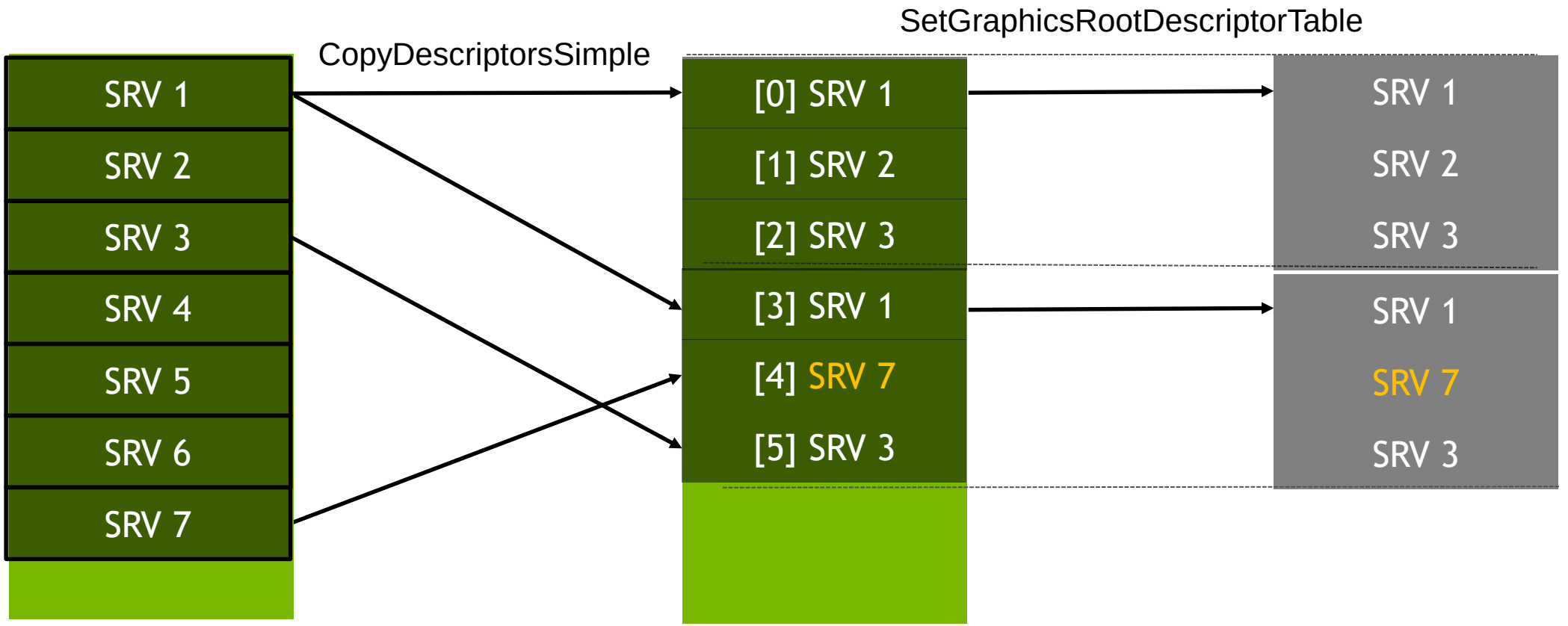


2 Draw Calls with same Root Signature

Typical DX12 SRV Binding Pattern



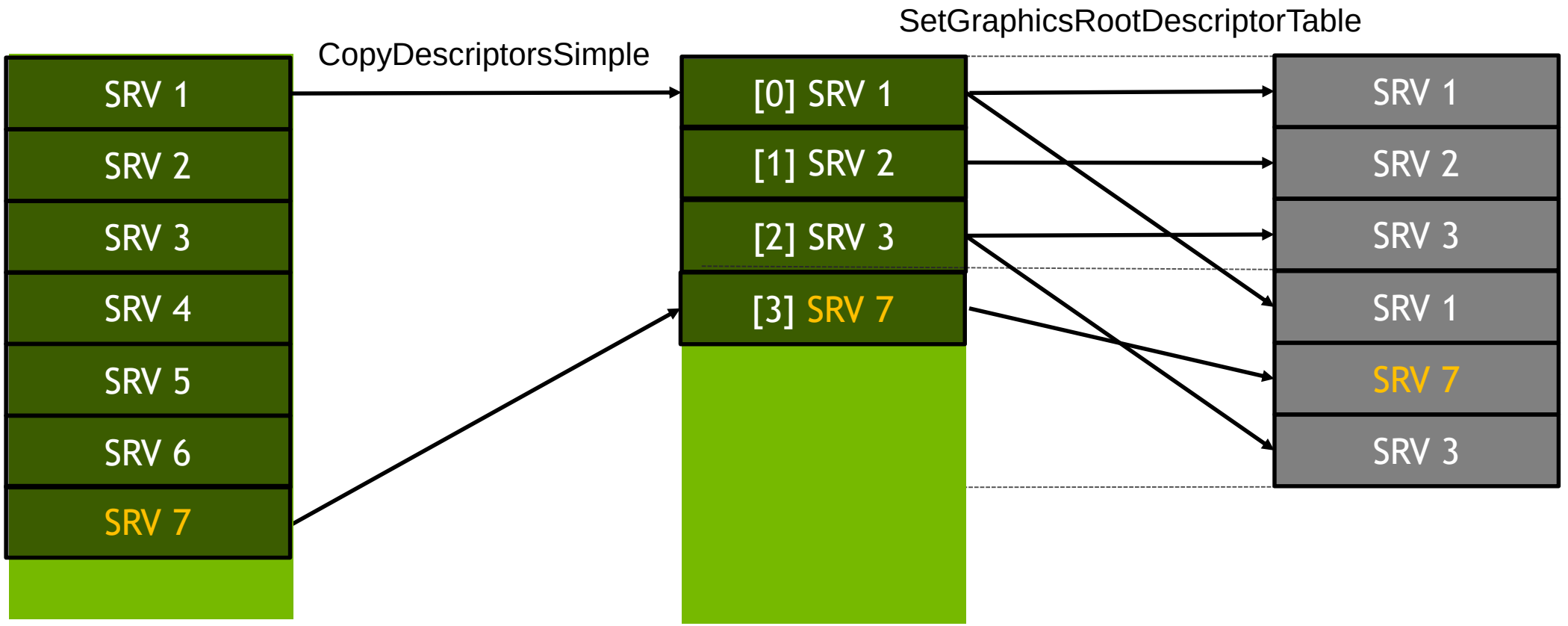
The Problem: Redundant Heap Entries



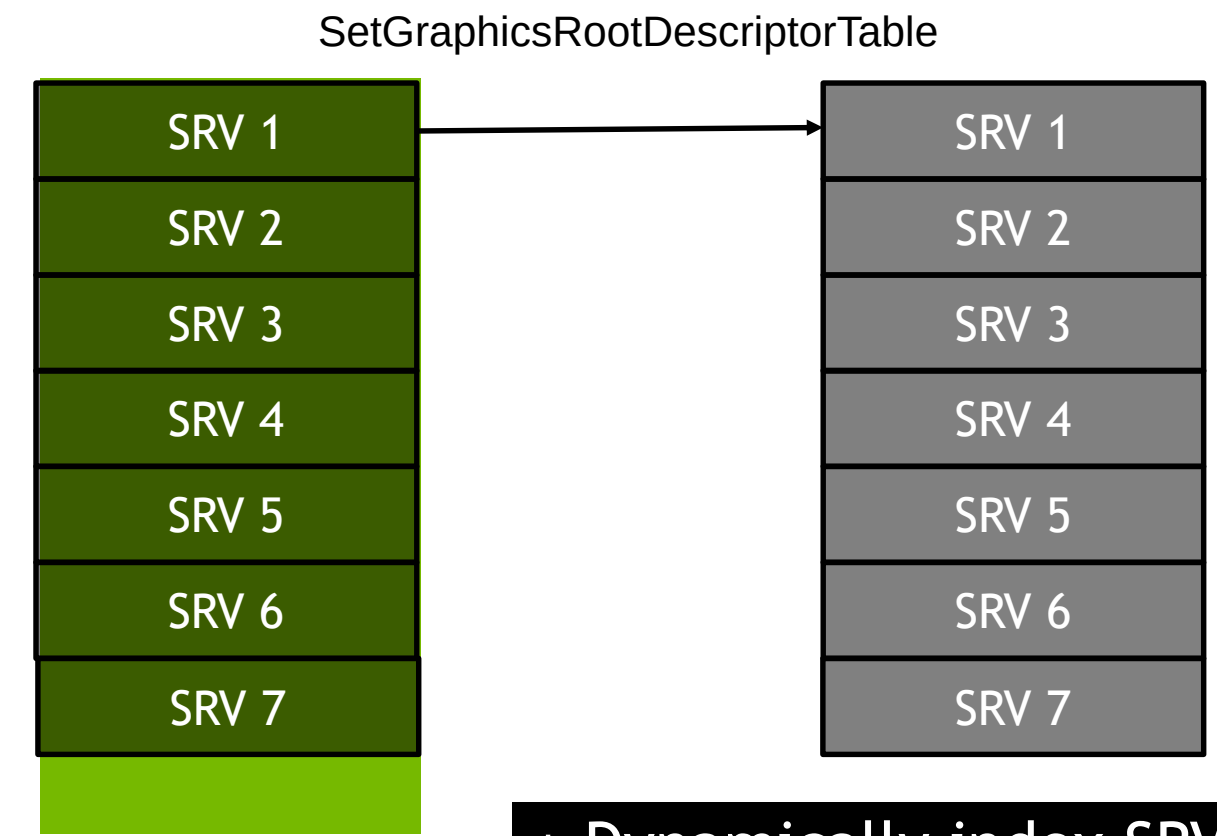
TSL1 & TSL2 caches use **heap indices** as tags

➔ Redundant entries in the shader-visible heap ➔ TSL1 & TSL2 cache thrashing ☹

Solution #1: Split SRV Ranges



Solution #2: Shader SRV Indexing



+ Dynamically index SRV descriptor in shaders using per-draw-call indices stored in a Root CBV

Split SRV Ranges vs Shader SRV Indexing

- Shader SRV Indexing
  - ☺ **Unique SRVs** in shader-visible descriptor heap
  - ☺ No CopyDescriptorsSimple calls used
  - ☹ Slight SM overhead (extra registers & instructions injected by driver)

- Split SRV Ranges
  - ☹ CopyDescriptorsSimple CPU overhead
  - ☹ SetGraphicsRootDescriptorTable CPU & GPU overhead
  - ☺ Can use the same shader byte code on DX12 & DX11

DX12 Advanced Topic:

Pixel Shader Barriers

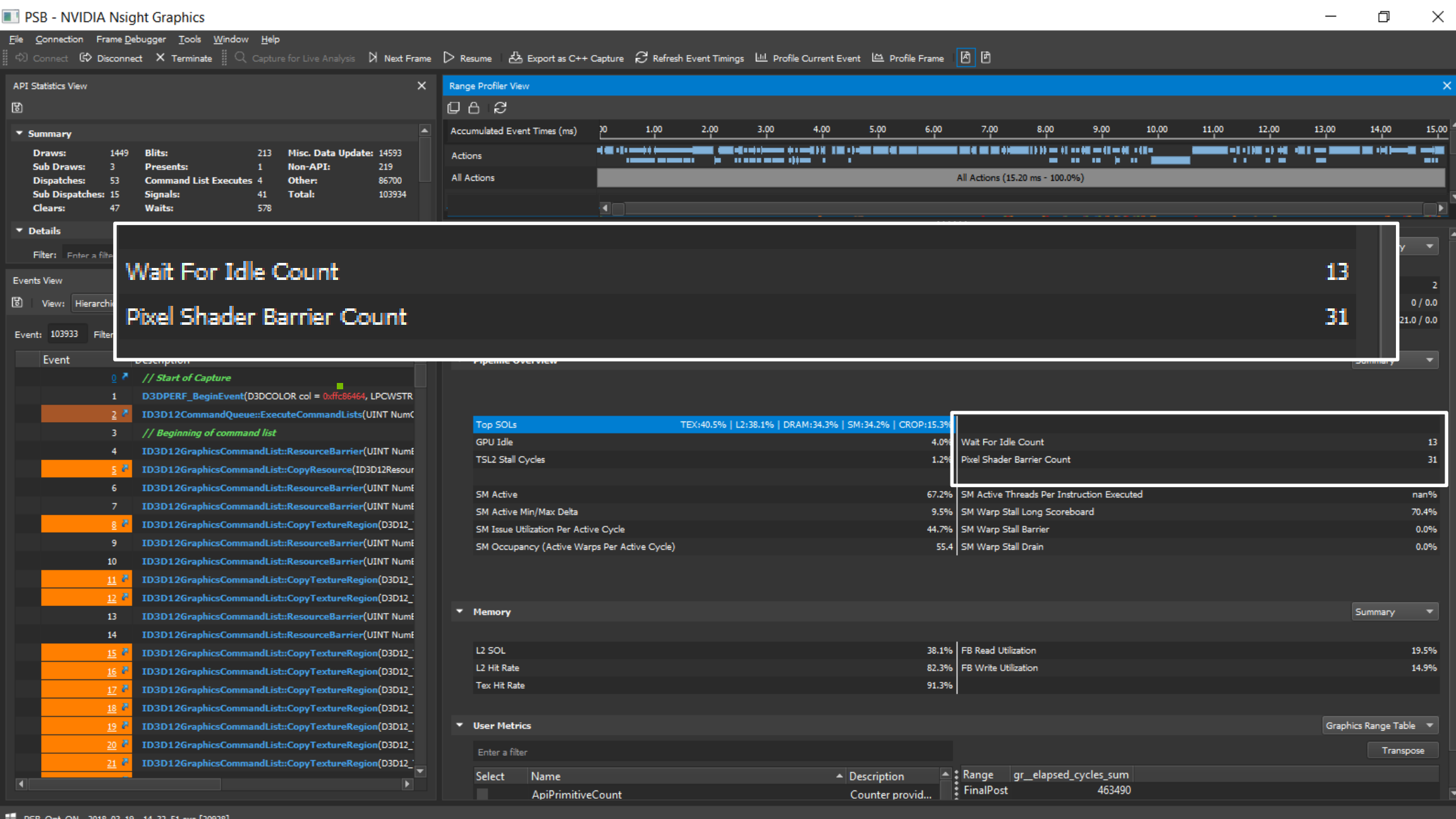
GPU: GTX 1080

Pixel Shader Barriers (PSBs)

- PSB == lightweight WFI (Wait For Idle) for PS-to-PS dependencies.
  - Hardware command available on Maxwell and beyond.
  - Used automatically by our driver on DX11.

- On DX12, used in ResourceBarrier Transition calls with:
  - StateBefore = D3D12_RESOURCE_STATE_**RENDER_TARGET**
  - StateAfter = D3D12_RESOURCE_STATE_**PIXEL_SHADER_RESOURCE**

- All other transitions map to full-pipeline WFIs.


PSB - NVIDIA Nsight Graphics
File Connection Frame Debugger Tools Window Help
Connect Disconnect Terminate Capture for Live Analysis Next Frame Resume Export as C++ Capture Refresh Event Timings Profile Current Event Profile Frame
API Statistics View
Summary
Draws: 1449 Blits: 213 Misc. Data Update: 14593
Sub Draws: 3 Presents: 1 Non-API: 219
Dispatches: 53 Command List Executes 4 Other: 86700
Sub Dispatches: 15 Signals: 41 Total: 103934
Clears: 47 Waits: 578
Details
Filter:
Range Profiler View
Accumulated Event Times (ms)
0.00 1.00 2.00 3.00 4.00 5.00 6.00 7.00 8.00 9.00 10.00 11.00 12.00 13.00 14.00 15.00
Actions
All Actions
All Actions (15.20 ms - 100.0%)
Wait For Idle Count 13
Pixel Shader Barrier Count 31
Events View
View:
Event: 103933 Filter
Event
0 // Start of Capture
1 D3DPERF_BeginEvent(D3DCOLOR col = 0xffc86464, LPCWSTR
2 ID3D12CommandQueue::ExecuteCommandLists(UINT Num
3 // Beginning of command list
4 ID3D12GraphicsCommandList::ResourceBarrier(UINT Num
5 ID3D12GraphicsCommandList::CopyResource(ID3D12Resour
6 ID3D12GraphicsCommandList::ResourceBarrier(UINT Num
7 ID3D12GraphicsCommandList::ResourceBarrier(UINT Num
8 ID3D12GraphicsCommandList::CopyTextureRegion(D3D12_
9 ID3D12GraphicsCommandList::ResourceBarrier(UINT Num
10 ID3D12GraphicsCommandList::ResourceBarrier(UINT Num
11 ID3D12GraphicsCommandList::CopyTextureRegion(D3D12_
12 ID3D12GraphicsCommandList::CopyTextureRegion(D3D12_
13 ID3D12GraphicsCommandList::ResourceBarrier(UINT Num
14 ID3D12GraphicsCommandList::ResourceBarrier(UINT Num
15 ID3D12GraphicsCommandList::CopyTextureRegion(D3D12_
16 ID3D12GraphicsCommandList::CopyTextureRegion(D3D12_
17 ID3D12GraphicsCommandList::CopyTextureRegion(D3D12_
18 ID3D12GraphicsCommandList::CopyTextureRegion(D3D12_
19 ID3D12GraphicsCommandList::CopyTextureRegion(D3D12_
20 ID3D12GraphicsCommandList::CopyTextureRegion(D3D12_
21 ID3D12GraphicsCommandList::CopyTextureRegion(D3D12_
Top SOLs TEX:40.5% | L2:38.1% | DRAM:34.3% | SM:34.2% | CROP:15.3%
GPU Idle 4.0%
TSL2 Stall Cycles 1.2%
SM Active 67.2%
SM Active Min/Max Delta 9.5%
SM Issue Utilization Per Active Cycle 44.7%
SM Occupancy (Active Warps Per Active Cycle) 55.4
Wait For Idle Count 13
Pixel Shader Barrier Count 31
SM Active Threads Per Instruction Executed nan%
SM Warp Stall Long Scoreboard 70.4%
SM Warp Stall Barrier 0.0%
SM Warp Stall Drain 0.0%
Memory
Summary
L2 SOL 38.1%
L2 Hit Rate 82.3%
Tex Hit Rate 91.3%
FB Read Utilization 19.5%
FB Write Utilization 14.9%
User Metrics
Graphics Range Table
Enter a filter
Transpose
Select Name Description
ApiPrimitiveCount Counter provid...
Range gr__elapsed_cycles_sum
FinalPost 463490

ResourceBarrier Flag Optimization

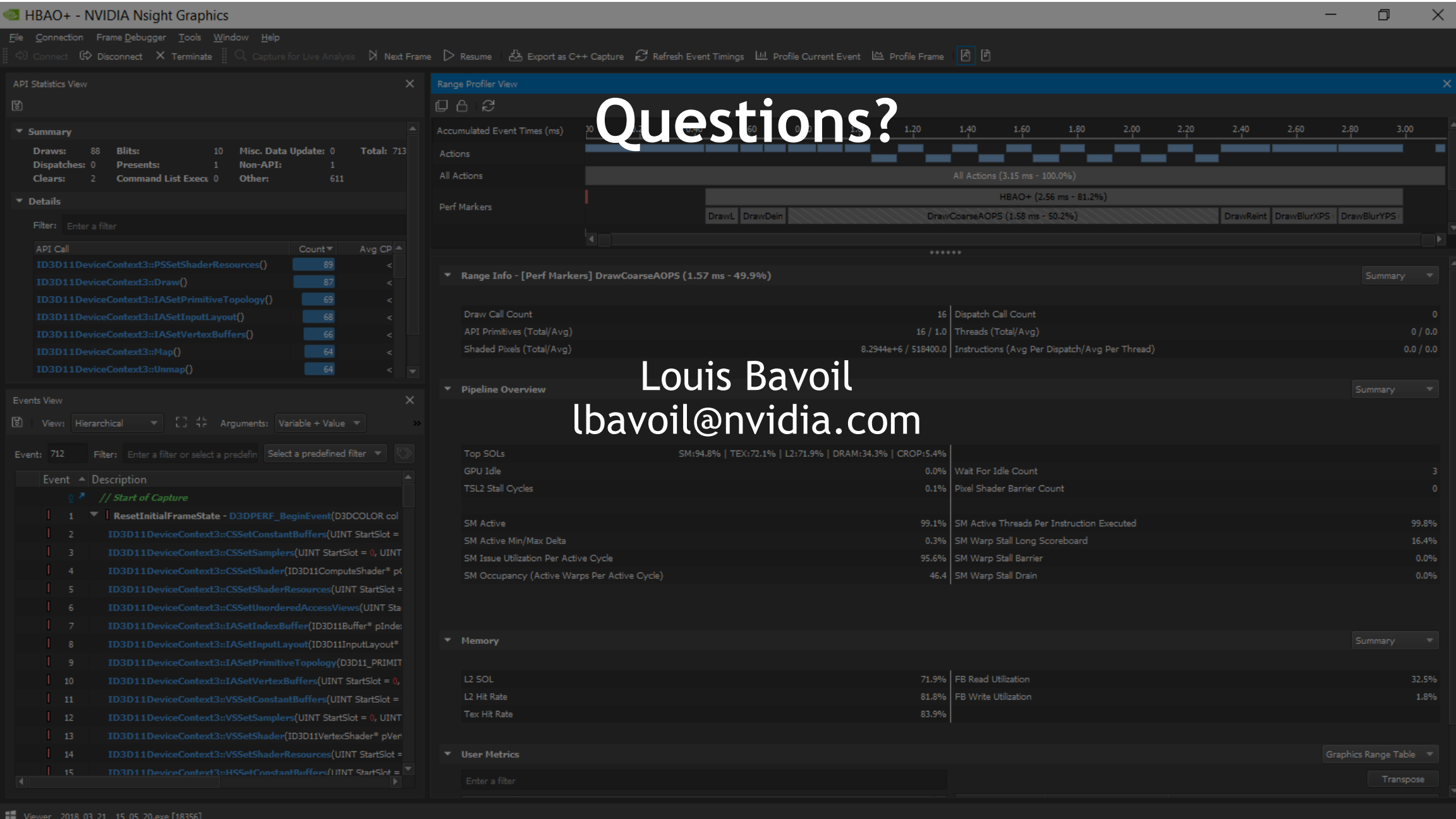
POST-PROCESSING CHAIN	BEFORE	AFTER	RATIO
Top SOLs	TEX:35.4% L2:33.3% SM:29.9%	TEX:40.5% L2:38.3% DRAM:36.1%	TEX:1.1x L2:1.2x DRAM:1.2x
Wait For Idle Count	44	13	
Pixel Shader Barrier Count	0	31	
GPU Elapsed Time	0.39 ms	0.29 ms	26% Gain

Conclusion

- **Nsight Graphics 1.0**
  - Makes it easier to export frames to C++ and build them as EXE
  - Exposes powerful hardware metrics in the Range Profiler

- Blog post for more details:
  - "The Peak-Performance Analysis Method for Optimizing Any GPU Workload"

- Demo of Nsight Graphics at NVIDIA Expo Booth



Questions?

Louis Bavoil
lbavoil@nvidia.com