



BRIGHTER & FASTER GRAPHICS WITH NVIDIA® NSIGHT™ VSE 4.7™

An Yan Software Engineer



Developer Tools: NVIDIA GameWorks™

AGENDA

Introducing NVIDIA® Nsight™ Visual Studio Edition 4.7™

Profiling Infiltrator from Epic Games



Developing on modern GPUs can be a cloudy business...



Shader Bugs

GPU Bottlenecks

API State Management

Performance

API Usage

Driver

CPU<->GPU Interaction

Memory

Resources

GPU Frametime

Shader Modification

Contributing Fragments



NVIDIA DEVELOPER TOOLS

BUILD. DEBUG. PROFILE.

Microsoft®
DirectX®

OpenGL

nVIDIA
CUDA

OpenGL|ES



GNU
C/C++

IDE INTEGRATION

Visual Studio®

eclipse

STANDALONE TOOLS



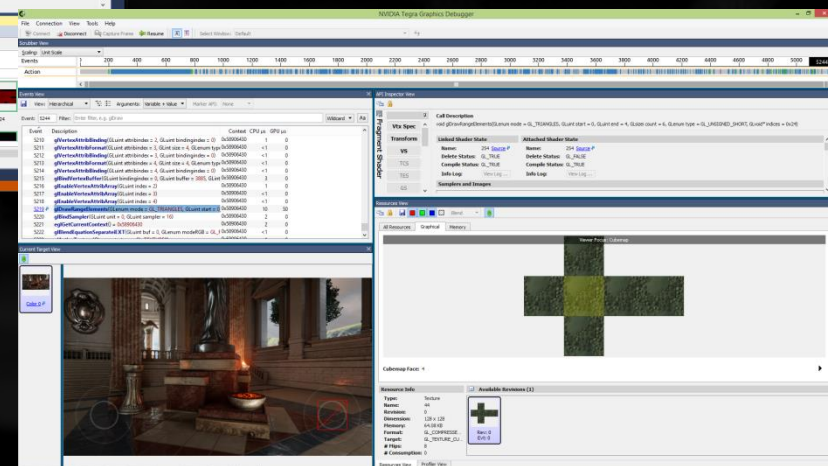
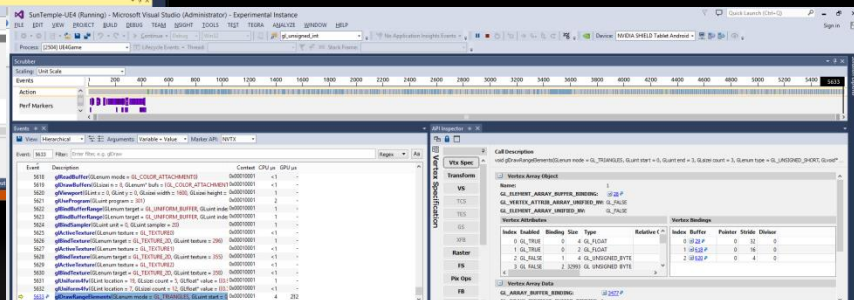
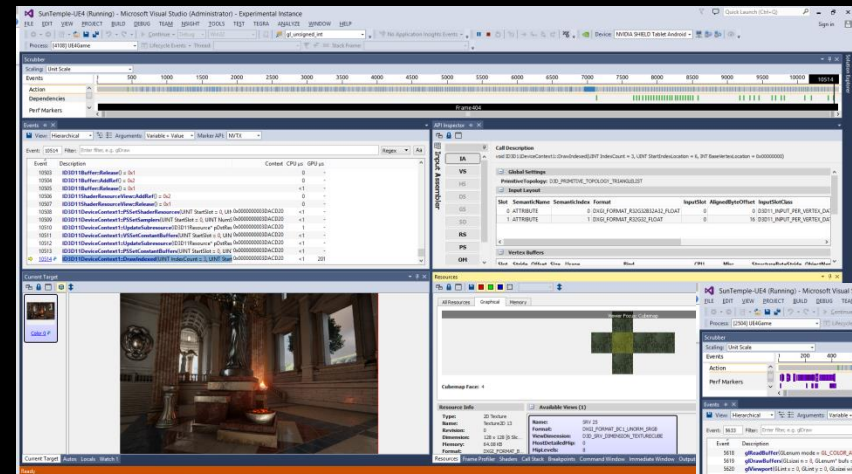
HARDWARE SUPPORT CPU AND GPU DEBUGGING & PROFILING



 **nvidia**
GAMEWORKS™



PICK A PLATFORM/API



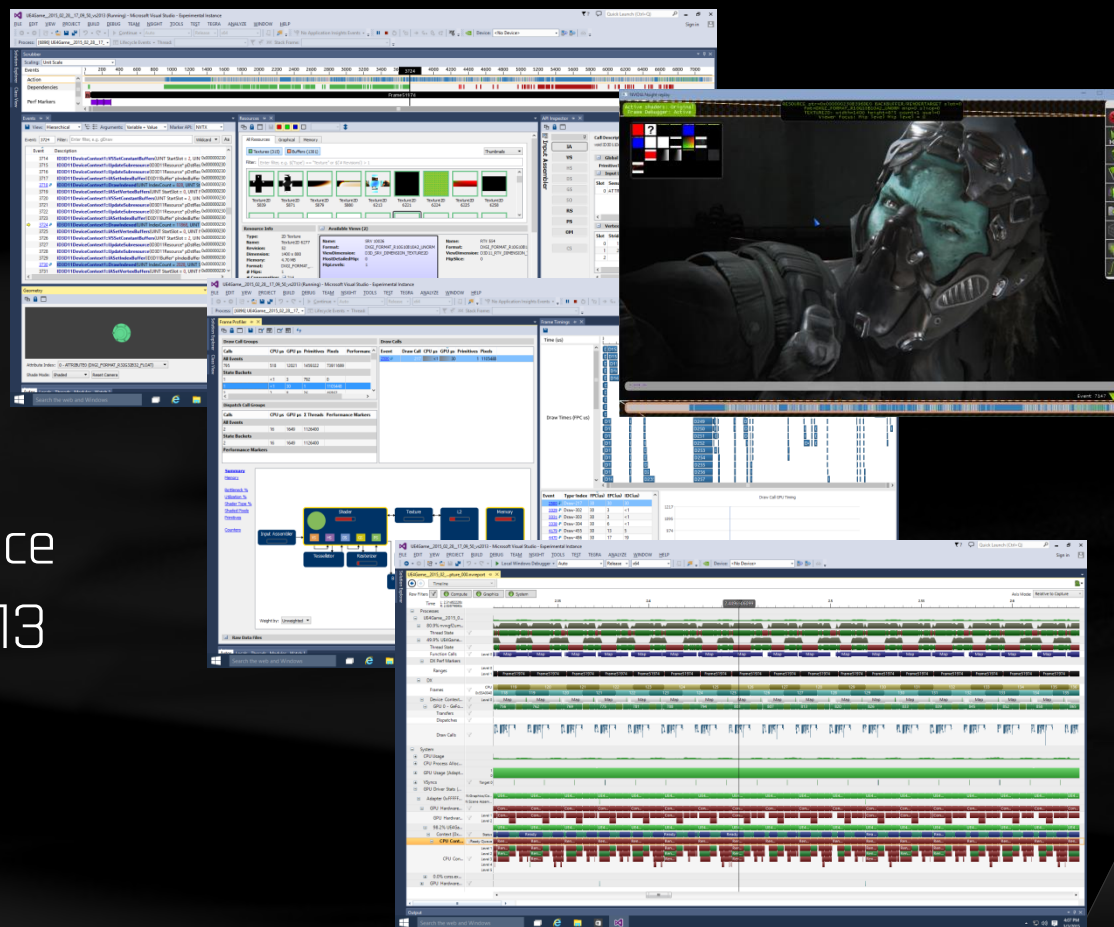
Microsoft®
DirectX®



NVIDIA® NSIGHT™ VISUAL STUDIO EDITION

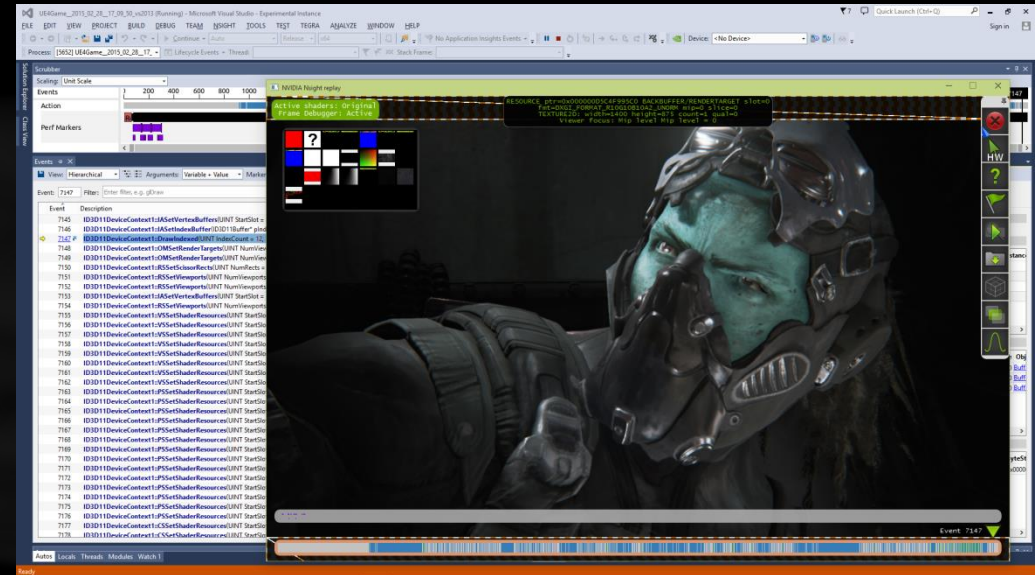
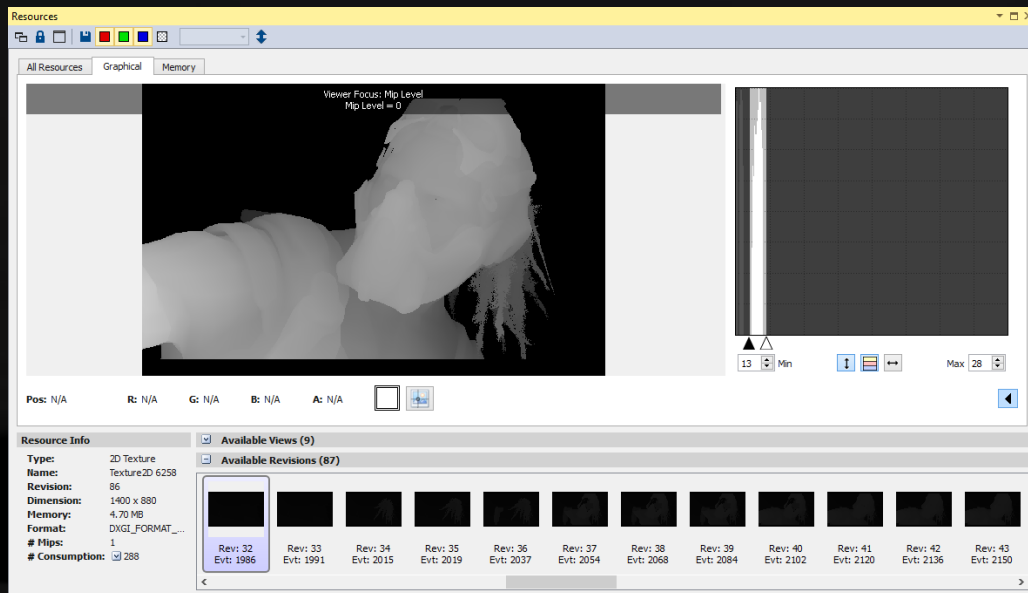
Accelerating Visual Computing Development

- Supports DirectX 9/11 and OpenGL
- Debug and profile graphics workloads
- Debug HLSL and GLSL shaders
- Debug and profile CUDA kernels
- Platform level profiling with system trace
- All in Visual Studio 2010, 2012 and 2013



NEW IN VERSION 4.7

- Maxwell architecture support (minus shader debugging)
- OpenGL 4.4 (minus ARB_sparse_texture)
- OpenGL capture with source code generation
- Texture histogram and range remapping
- CUDA 7.5 support

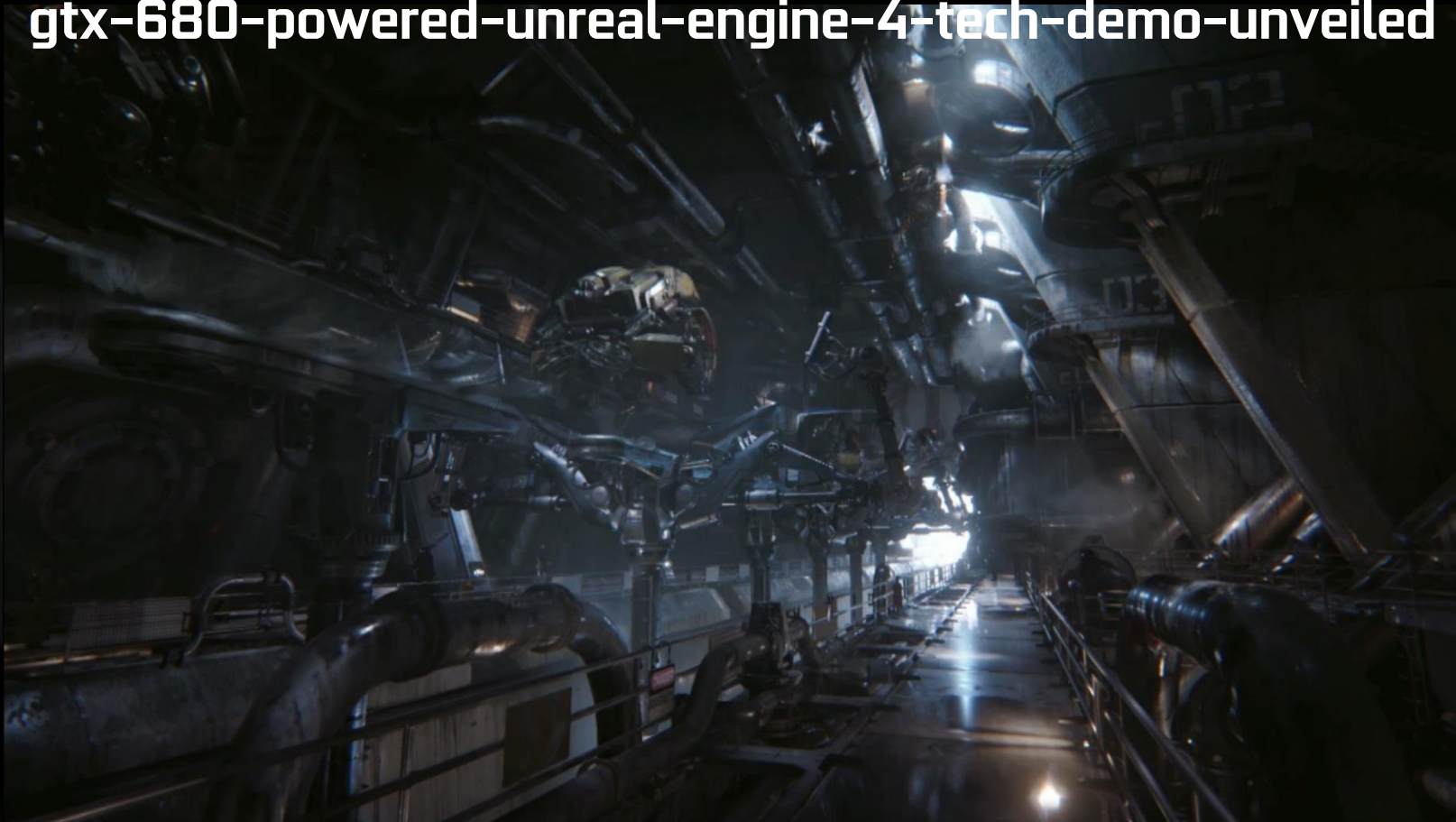


PROFILING INFILTRATOR



PROFILING INFILTRATOR

View Infiltrator at <http://www.geforce.cn/whats-new/articles/infiltrator-gtx-680-powered-unreal-engine-4-tech-demo-unveiled>



PROFILING INFILTRATOR – CPU BOUND

The screenshot shows the NVIDIA Infiltrator application settings and launch interface within the Visual Studio IDE. The interface is divided into several sections:

- Application Settings:** This section is highlighted with a green callout. It includes fields for Connection Name (localhost), Application (C:\Users\kief\Desktop\GDC2015\Infiltrator_Capture_GPUBound\x64\Release\UE4Game_2015_02_28_17_09_50.exe), Arguments, Working Directory (C:\Users\kief\Desktop\GDC2015\Infiltrator_Capture_GPUBound\), and Remote Options.
- Triggers and Actions:** This section shows Start: On Process Launch and Stop: Manually.
- Activity Type:** This section is highlighted with a green callout. It includes options for Trace Application, Trace Process Tree (selected), Profile CUDA Application, and Profile CUDA Process Tree.
- Trace Settings:** This section is highlighted with a green callout. It includes domains (DirectX) and various trace categories (System, Tools Extension, CUDA, OpenCL, DirectX) with checkboxes for API Trace, Workload Trace, and Session Control.
- Launch!:** This section is highlighted with a green callout. It includes buttons for Start, Stop, and Cancel, and a checkbox for Open Report on Stop.

The bottom of the interface shows the Output window with the text "Find Results 2".

Application Settings

Trace System & DirectX

Launch!



PROFILING INFILTRATOR – CPU BOUND

Runtime/Driver Consumes
CPU Time...Next Generation
APIs Should Help!

Activity1.nvact* InfiltratorDemo150...apture_000.nvreport

Summary Report

Target Process: UE4Game.exe [2464] Clear

Session Overview

Summary of session information related to the captured data.

- [Session Summary](#)
- [Timeline](#)
- [Activity](#)

InfiltratorDemo.exe (Trace Process Tree)

Captured 9.24 seconds of data on 2/28/2015 6:46:04 PM

Arguments:
Working Dir: c:\users\jkiel\desktop\infiltrator\windowsnoeditor
Connection: DTPNCWin764-02

Process Overview

Summary of process information relating to the captured data.

Process Name: UE4Game.exe
Process ID: 2464
Command Line: "c:\users\jkiel\desktop\infiltrator\windowsnoeditor\Engine\Binaries\Win64\UE4Game.exe" ..\..\..\InfiltratorDemo\InfiltratorDemo.uproject
Process Tree: UE4Game.exe [2464]
Threads: 35

DirectX Overview

Summary of captured DirectX activity.

- [API Calls | Summary](#)
- [Draw Calls](#)
- [Frames](#)
- [Dispatches](#)
- [Performance Markers](#)
- [Transfers](#)

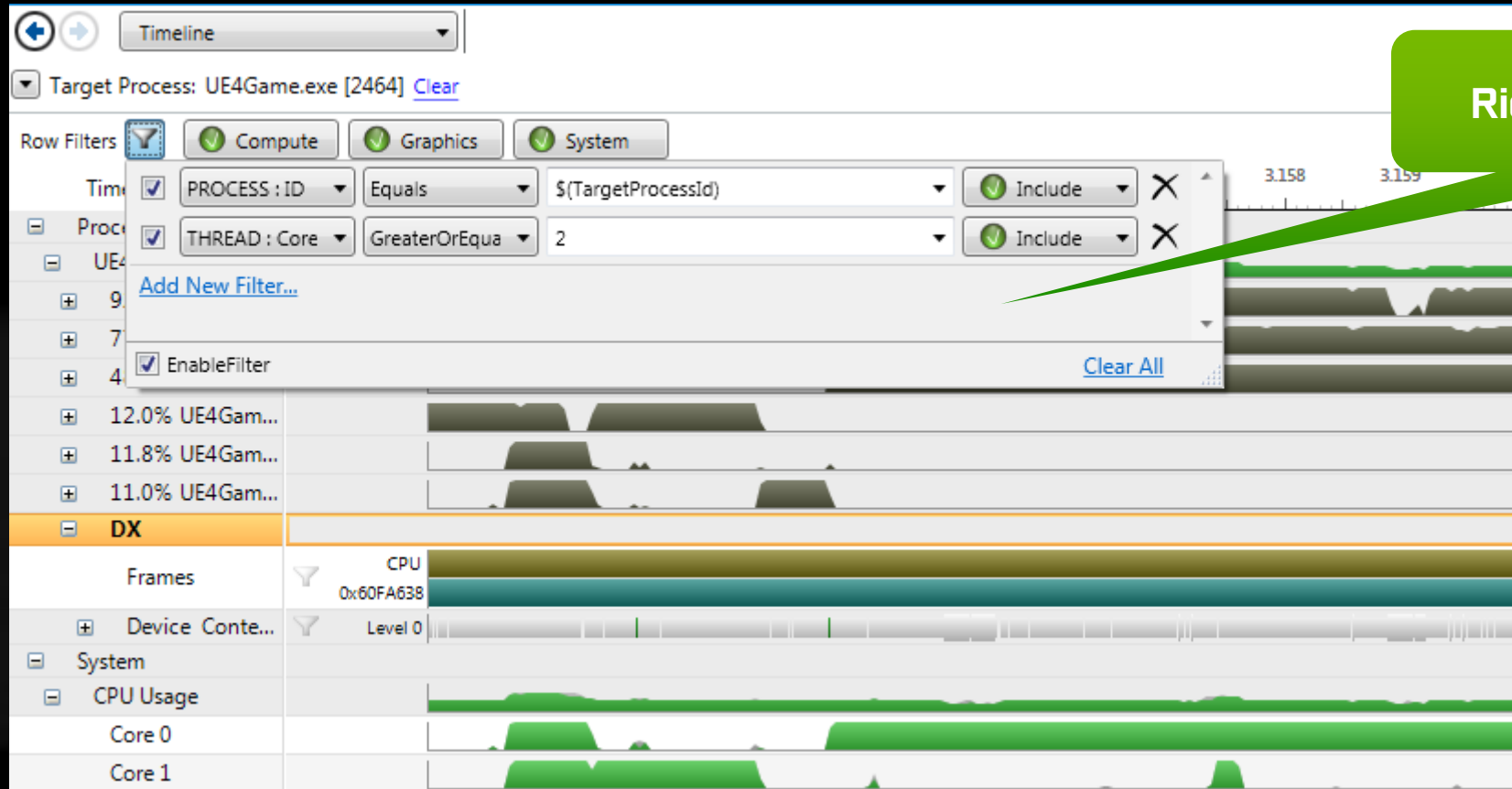
CPU All Cores			
API Time (%)	Min	Avg	Max
51.2	31	33.3	42
Batch	2,106	2,492.8	3,727

GPU 0 GeForce GTX 980			
Utilization (%)	Min	Avg	Max
23.5	14	52	67
ms/F	3.3	26.2	71.2
Cmd%	879	4,048	5,420

Summary Screen Results



PROFILING INFILTRATOR – CPU BOUND

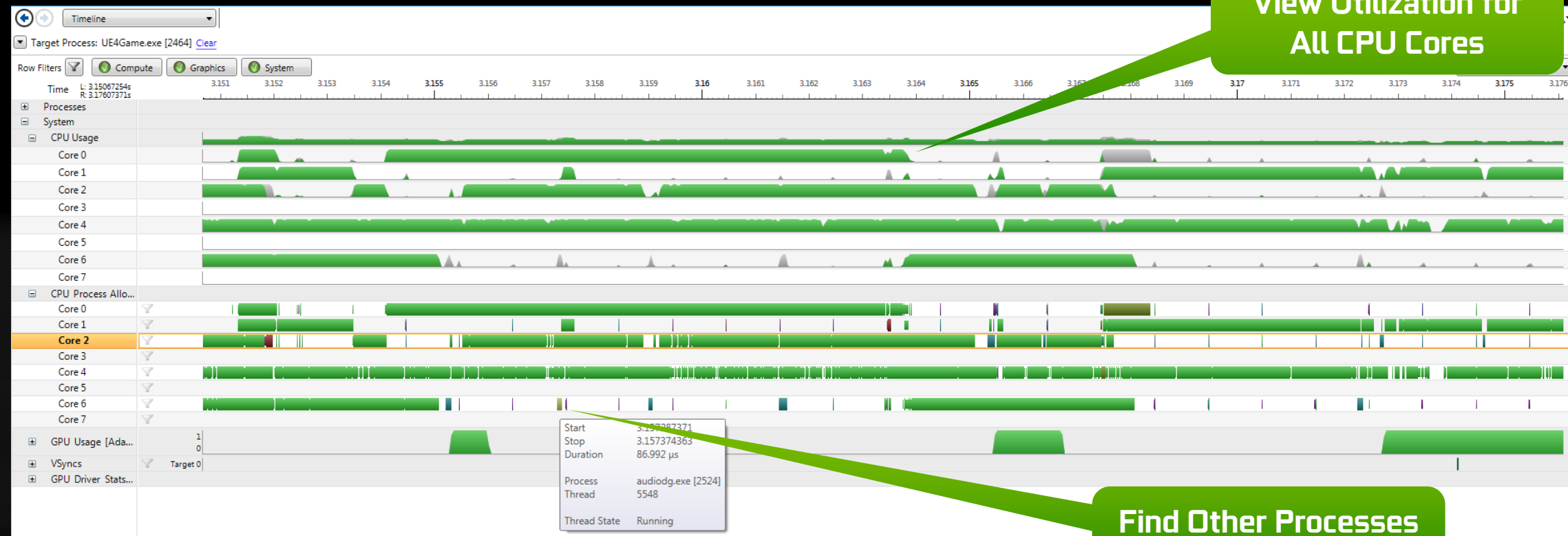


Rich Filtering Capabilities



PROFILING INFILTRATOR – CPU BOUND

View Utilization for
All CPU Cores



Find Other Processes
That Steal CPU Time



The screenshot displays the NVIDIA Nsight System performance analysis tool. The top bar shows the target process as 'UE4Game.exe [2464]'. The 'Row Filters' section includes 'Compute', 'Graphics', and 'System'. The 'Time' axis is set to 'Relative to Capture'. The 'Processes' section shows the CPU usage of various processes, with 'UE4Game.exe' at 89.01%. The 'Frames' section shows the GPU usage of various frames, with frame 748 highlighted. The 'GPU 0 - GeForce RTX 3080' section shows the GPU usage of various frames, with frame 748 highlighted. The 'Transfers' and 'Dispatches' sections show the transfer and dispatch events. The 'Draw Calls' section shows the draw call events. A green arrow points to a specific event in the 'GPU 0 - GeForce RTX 3080' section, which is labeled '748'.

**Gaps In GPU Workloads
= CPU Bound**

PROFILING INFILTRATOR – GPU BOUND

UE4Game_2015_02_...pture_000.nvreport

Summary Report



Session Overview

Summary of session information related to the captured data.

- [Session Summary](#)
- [Timeline](#)
- [Activity](#)

UE4Game_2015_02_28_17_09_50.exe (Trace Application)

Captured 9.00 seconds of data on 2/28/2015 7:22:52 PM

Arguments:
Working Dir: C:\Users\jkiel\Documents\NVIDIA Nsight\Captures\UE4Game_2015_02_28_17_09_50\
Connection: localhost



DirectX Overview

Summary of captured DirectX activity.

- [API Calls | Summary](#)
- [Frames](#)
- [Performance Markers](#)
- [Draw Calls](#)
- [Dispatches](#)
- [Transfers](#)

CPU All Cores

API Time (%)	Min	Avg	Max
86.0	8	8.0	8
Batch	2,560	2,560.0	2,560

GPU 0 GeForce GTX 980

Utilization (%)	Min	Avg	Max
96.8	18	18	23
ms/F	86.1	99.1	99.6
Cmd%	791	794	796
#Items			

⚠ There are 481 CPU frames, but only 480 GPU frames.

Summary Results: Lots
Of GPU Work!

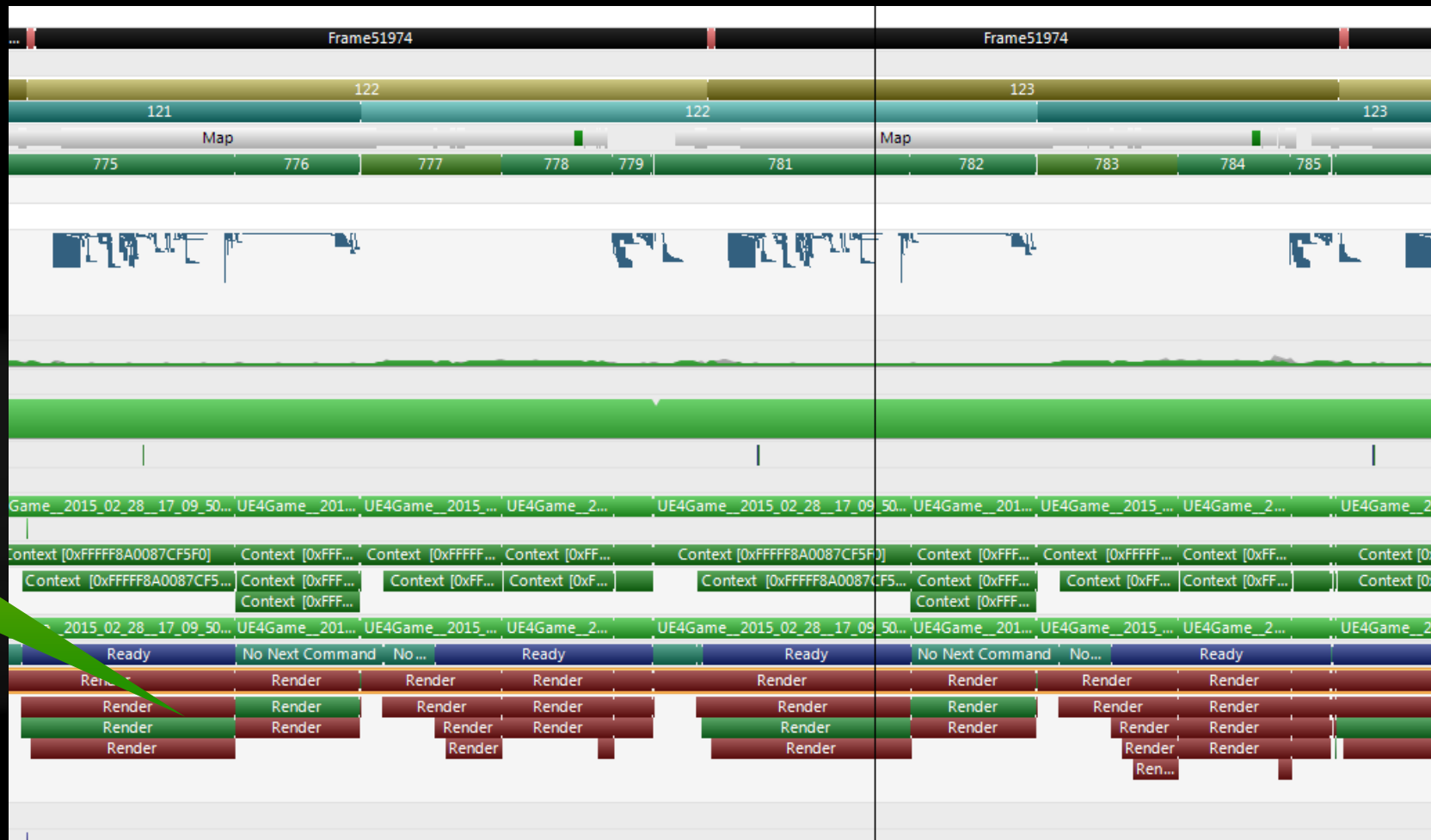


PROFILING INFILTRATOR – GPU BOUND



PROFILING INFILTRATOR – GPU BOUND

Correlate With
GPUView ETW Data



PROFILING INFILTRATOR – GPU BOUND

Blocking Map
Call On CPU

Jump To Source

Full Stack Trace

The screenshot displays the NVIDIA Infiltrator profiling tool interface. The top panel shows a timeline of GPU and CPU activity. The middle panel displays a 'Function Calls' view, highlighting a 'Map' call. The bottom panel shows a 'Full Stack Trace' for the selected function call, listing the call stack from the application down to the operating system.

Function Calls View:

Function	Thread	Start Time	End Time
Map	0x55A0040	237110488	243111908

Full Stack Trace:

Address	Module Name	Function	File Name
0x00000000	UE4Game_2015_02_28_17_09_50.exe	void __cdecl RunFrame0Part0(void)	c:\users\jkie\documents\UE4Game_2015_02_28_17_09_50.exe
0x00000000	UE4Game_2015_02_28_17_09_50.exe	WinMain	c:\users\jkie\documents\UE4Game_2015_02_28_17_09_50.exe
0x00000000	UE4Game_2015_02_28_17_09_50.exe	_tmainCRTStartup	f:\dd\vctools\crt\crtw32\bin\crt0.obj
0x00000000	kernel32.dll		
0x00000000	ntdll.dll		

Source Code View:

```
3440 NV_CHECK_RESULT(result);
3441 memcpy(D3D11_MAPPED_SUBRESOURCE_temp_259.pData, NV_GET_RESOURCE(void*, 258), 96); // Applying update to mapped data
3442 pID3D11DeviceContext_uifid_16->Unmap(pID3D11Resource_uifid_254680, 0);
3443
3444 // Map 259 of 464
3445 D3D11_MAPPED_SUBRESOURCE D3D11_MAPPED_SUBRESOURCE_temp_260;
3446 result = pID3D11DeviceContext_uifid_16->Map(pID3D11Resource_uifid_254682, 0, D3D11_MAP_READ, 0, &D3D11_MAPPED_SUBRESOURCE_temp_260);
3447 NV_CHECK_RESULT(result);
3448 memcpy(D3D11_MAPPED_SUBRESOURCE_temp_260.pData, NV_GET_RESOURCE(void*, 259), 96); // Applying update to mapped data
3449 pID3D11DeviceContext_uifid_16->Unmap(pID3D11Resource_uifid_254682, 0);
3450
3451 // Map 260 of 464
3452 D3D11_MAPPED_SUBRESOURCE D3D11_MAPPED_SUBRESOURCE_temp_261;
3453 result = pID3D11DeviceContext_uifid_16->Map(pID3D11Resource_uifid_254684, 0, D3D11_MAP_READ, 0, &D3D11_MAPPED_SUBRESOURCE_temp_261);
3454 NV_CHECK_RESULT(result);
3455 memcpy(D3D11_MAPPED_SUBRESOURCE_temp_261.pData, NV_GET_RESOURCE(void*, 260), 48); // Applying update to mapped data
3456 pID3D11DeviceContext_uifid_16->Unmap(pID3D11Resource_uifid_254684, 0);
3457
3458 // Map 261 of 464
3459 D3D11_MAPPED_SUBRESOURCE D3D11_MAPPED_SUBRESOURCE_temp_262;
3460 result = pID3D11DeviceContext_uifid_16->Map(pID3D11Resource_uifid_254686, 0, D3D11_MAP_READ, 0, &D3D11_MAPPED_SUBRESOURCE_temp_262);
3461 NV_CHECK_RESULT(result);
3462 memcpy(D3D11_MAPPED_SUBRESOURCE_temp_262.pData, NV_GET_RESOURCE(void*, 261), 48); // Applying update to mapped data
3463 pID3D11DeviceContext_uifid_16->Unmap(pID3D11Resource_uifid_254686, 0);
3464 pID3D11DeviceContext_uifid_16->OMSetRenderTargets(1, &pID3D11RenderTargetView_uifid_1219, NULL);
3465
3466 static D3D11_VIEWPORT D3D11_VIEWPORT_temp_1 = {0.0f, 0.0f, HexToF16((float)0x44AF0800/*1400.0f*/), HexToF16((float)0x44AC8000/*1400.0f*/), 0.0f, 0.0f};
3467 pID3D11DeviceContext_uifid_16->RSSetViewports(1, &D3D11_VIEWPORT_temp_1);
3468
3469 // Map 262 of 464
3470 D3D11_MAPPED_SUBRESOURCE D3D11_MAPPED_SUBRESOURCE_temp_263;
3471 result = pID3D11DeviceContext_uifid_16->Map(pID3D11Resource_uifid_254688, 0, D3D11_MAP_READ, 0, &D3D11_MAPPED_SUBRESOURCE_temp_263);
3472 NV_CHECK_RESULT(result);
3473 pID3D11DeviceContext_uifid_16->Unmap(pID3D11Resource_uifid_254688, 0);
3474
3475 // Map 263 of 464
3476 D3D11_MAPPED_SUBRESOURCE D3D11_MAPPED_SUBRESOURCE_temp_264;
3477 result = pID3D11DeviceContext_uifid_16->Map(pID3D11Resource_uifid_254690, 0, D3D11_MAP_READ, 0, &D3D11_MAPPED_SUBRESOURCE_temp_264);
3478 NV_CHECK_RESULT(result);
3479 memcpy(D3D11_MAPPED_SUBRESOURCE_temp_264.pData, NV_GET_RESOURCE(void*, 262), 65536); // Applying update to mapped data
3480
3481 // Map 264 of 464
3482 D3D11_MAPPED_SUBRESOURCE D3D11_MAPPED_SUBRESOURCE_temp_265;
3483 result = pID3D11DeviceContext_uifid_16->Map(pID3D11Resource_uifid_254692, 0, D3D11_MAP_READ, 0, &D3D11_MAPPED_SUBRESOURCE_temp_265);
3484 NV_CHECK_RESULT(result);
3485 memcpy(D3D11_MAPPED_SUBRESOURCE_temp_265.pData, NV_GET_RESOURCE(void*, 263), 256); // Applying update to mapped data
3486 pID3D11DeviceContext_uifid_16->Unmap(pID3D11Resource_uifid_254692, 0);
3487
3488 // Map 265 of 464
3489 D3D11_MAPPED_SUBRESOURCE D3D11_MAPPED_SUBRESOURCE_temp_266;
3490 result = pID3D11DeviceContext_uifid_16->Map(pID3D11Resource_uifid_254694, 0, D3D11_MAP_READ, 0, &D3D11_MAPPED_SUBRESOURCE_temp_266);
3491 NV_CHECK_RESULT(result);
3492 memcpy(D3D11_MAPPED_SUBRESOURCE_temp_266.pData, NV_GET_RESOURCE(void*, 264), 64); // Applying update to mapped data
3493 pID3D11DeviceContext_uifid_16->Unmap(pID3D11Resource_uifid_254694, 0);
3494
3495 // Map 266 of 464
3496 D3D11_MAPPED_SUBRESOURCE D3D11_MAPPED_SUBRESOURCE_temp_267;
3497 result = pID3D11DeviceContext_uifid_16->Map(pID3D11Resource_uifid_254696, 0, D3D11_MAP_READ, 0, &D3D11_MAPPED_SUBRESOURCE_temp_267);
3498 NV_CHECK_RESULT(result);
3499 memcpy(D3D11_MAPPED_SUBRESOURCE_temp_267.pData, NV_GET_RESOURCE(void*, 265), 256); // Applying update to mapped data
3500 pID3D11DeviceContext_uifid_16->Unmap(pID3D11Resource_uifid_254696, 0);
3501
3502
```

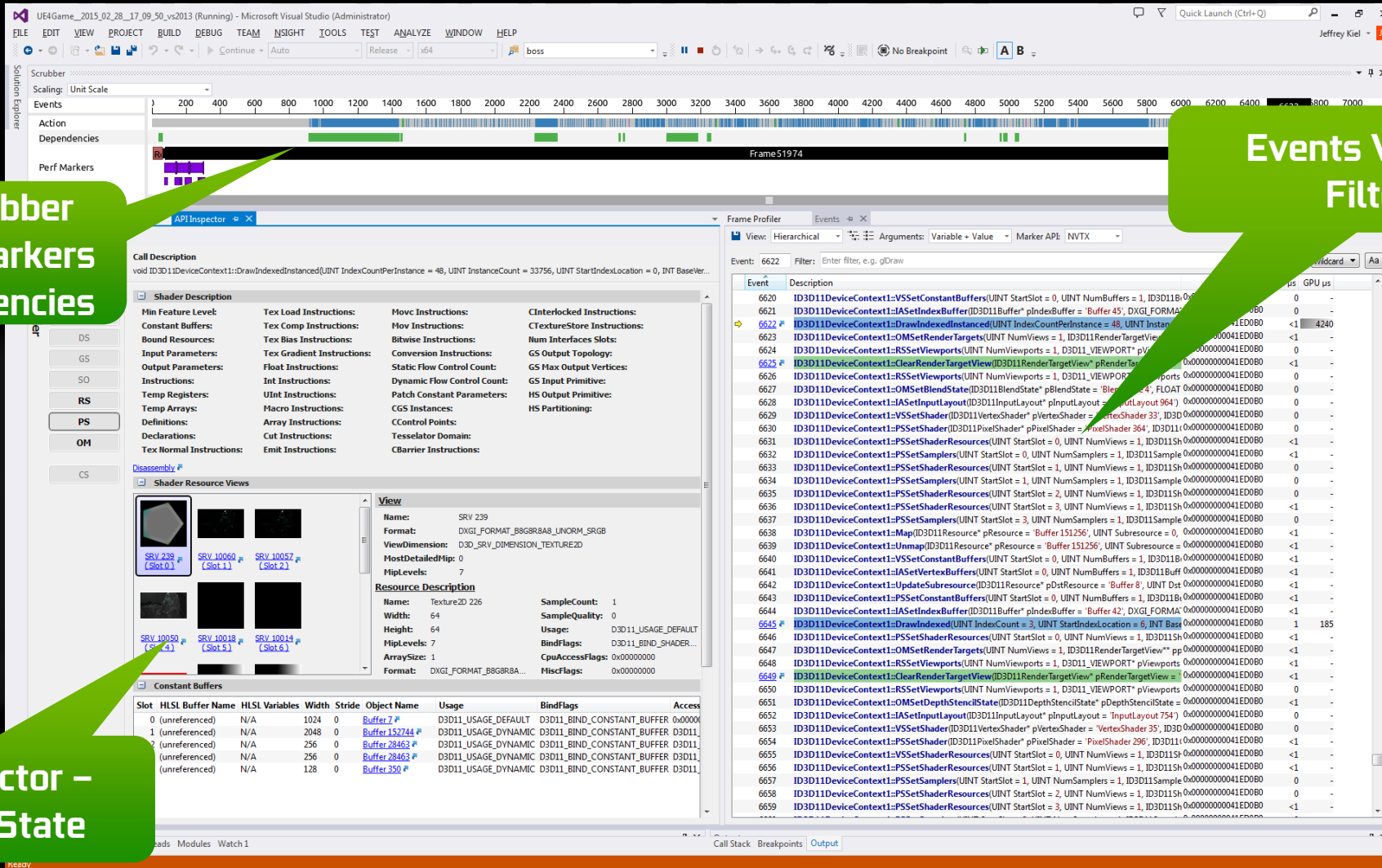


PROFILING INFILTRATOR - PROFILING

Scene Scrubber
With Perf Markers
and Dependencies

Events View With
Filtering

API Inspector –
View All State



PROFILING INFILTRATOR – PROFILING

Inspect All Resources

Visualize Pre-transform Geometry

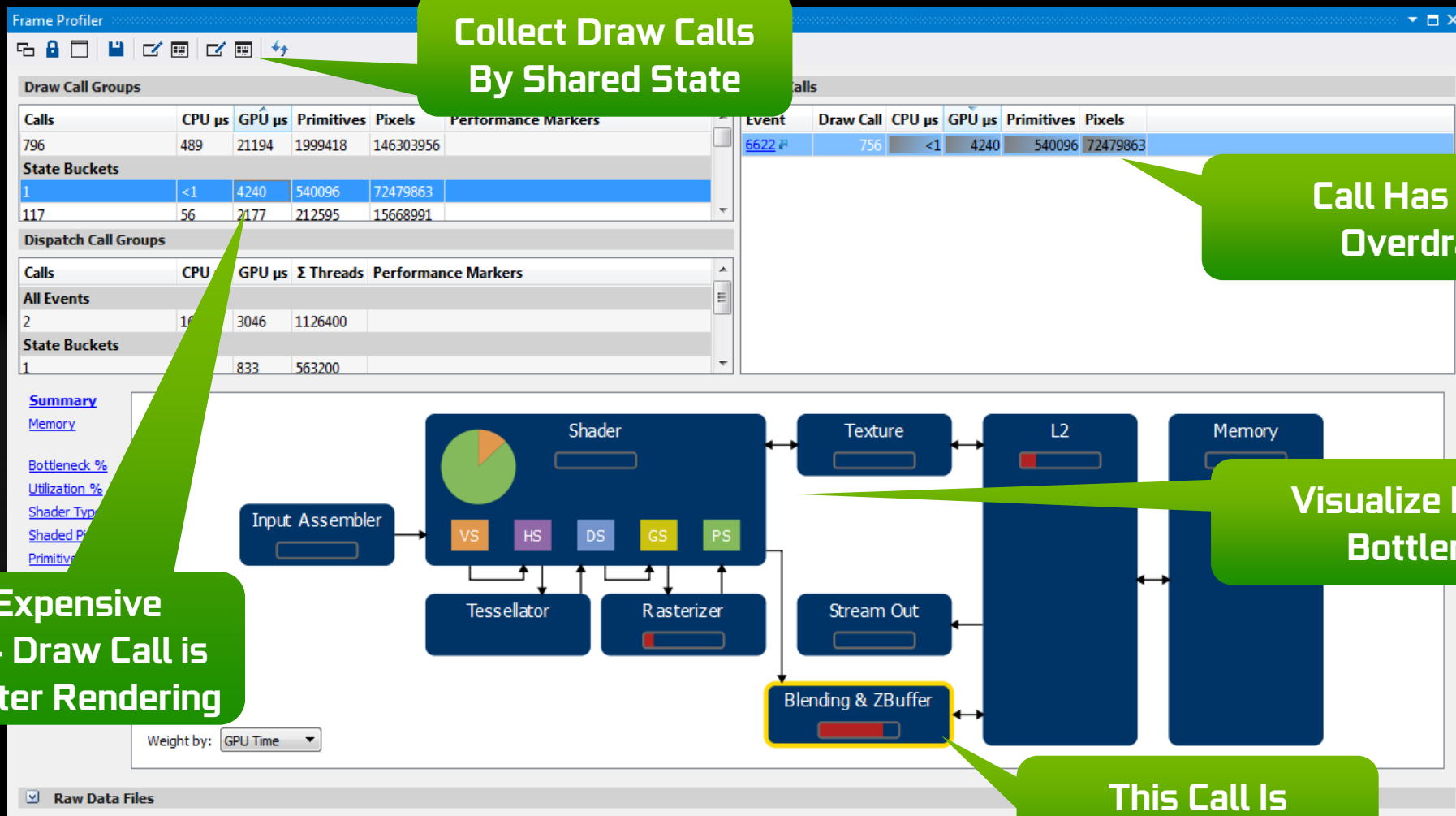
The screenshot displays the Unreal Engine 4 Profiling Infiltrator tool within the Microsoft Visual Studio environment. The interface is divided into several panels:

- Top Panel:** Shows the game's execution timeline with a scrubber and various event tracks (Action, Dependencies, Perf Markers).
- Resources Panel:** Displays a grid of 315 textures and 130 buffers. A filter is applied to show only textures. The selected texture is a 2D texture with dimensions 1400 x 880 and 4.70 MB.
- Resource Info Panel:** Provides detailed information about the selected resource, including its name, format, dimensions, and mipmap levels.
- Geometry Panel:** Visualizes the pre-transform geometry of the selected resource, showing a 3D model of a mechanical component.

The bottom status bar indicates the tool is ready and shows the current frame (Frame 51974).



PROFILING INFILTRATOR – PROFILING



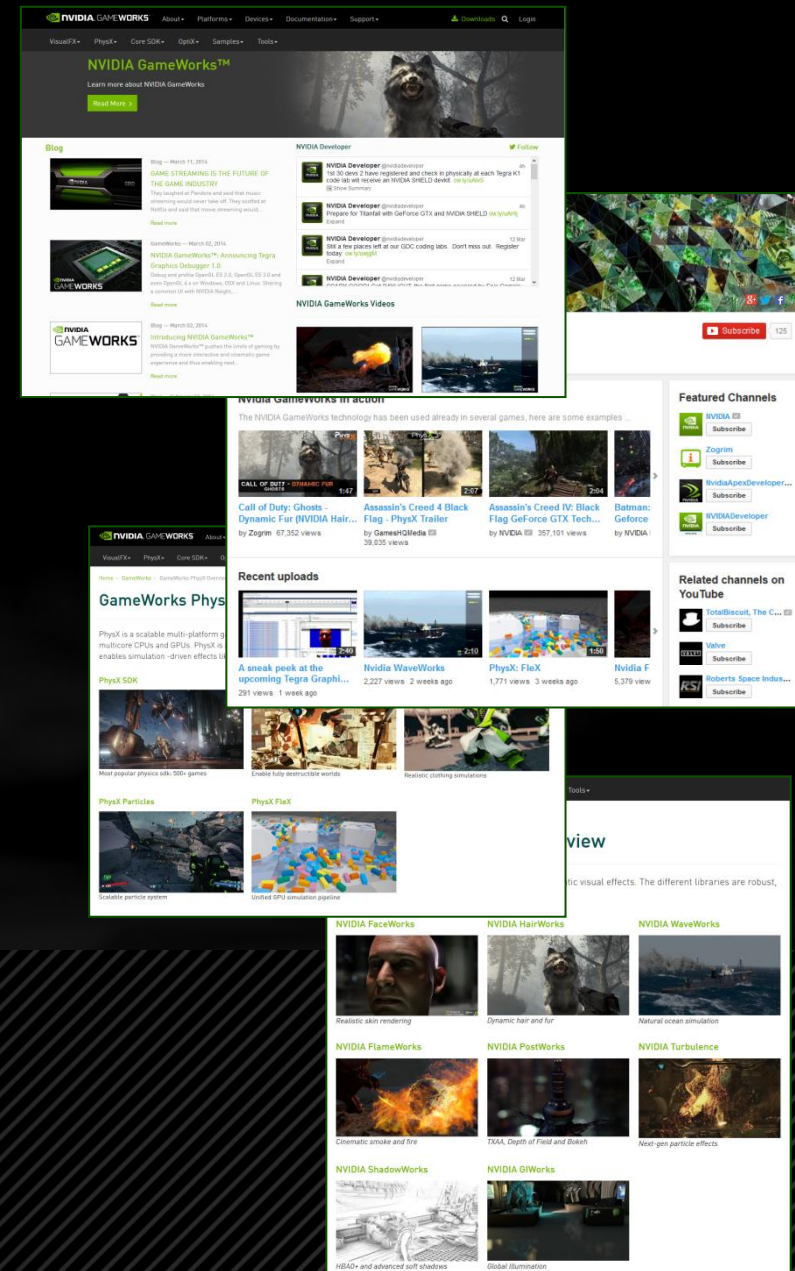
NVIDIA GAMEWORKS™

Web: <http://gameworks.nvidia.com>

Survey: <https://developer.nvidia.com/developer-tools-survey-201503>

YouTube: <https://www.youtube.com/user/nvidiaGameWorks>

Twitter: <https://twitter.com/nvidiadeveloper>



Q&A

- E-mail: devtools-support@nvidia.com
- Forum: <https://devtalk.nvidia.com/default/board/84/nsight-visual-studio-edition/>

