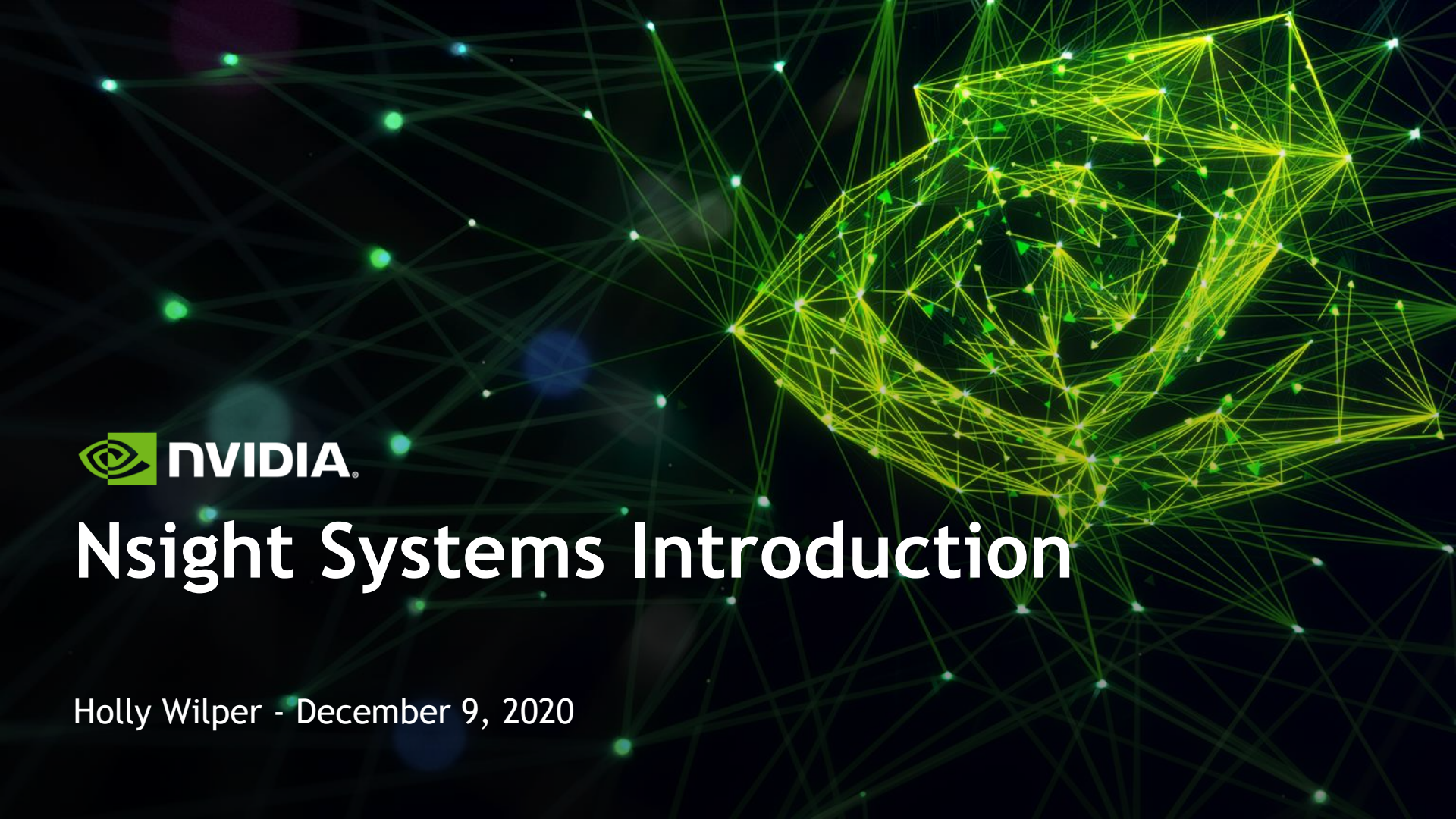


Nsight Systems Introduction

Holly Wilper - December 9, 2020

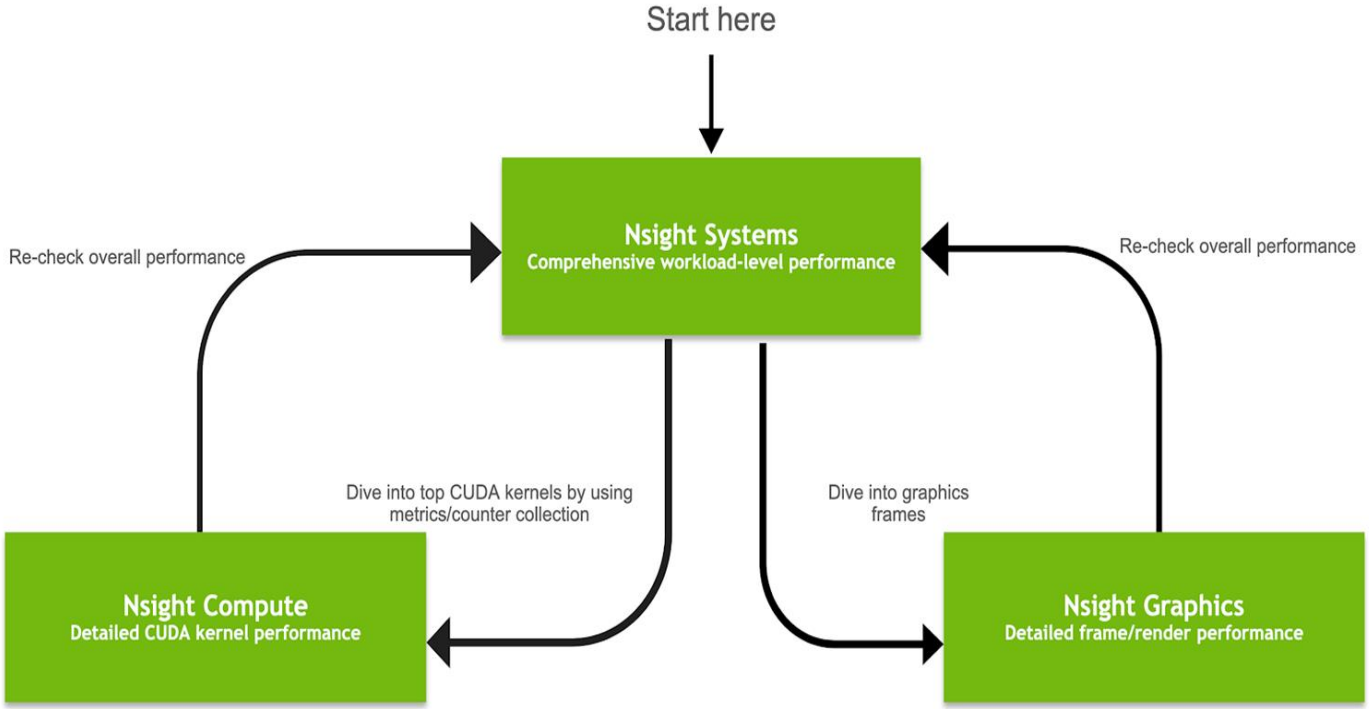
Nsight Product Family

Workflow

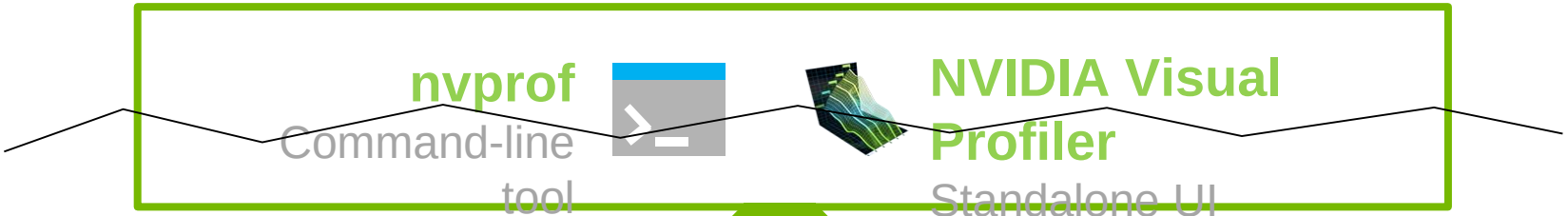
Nsight Systems -
Analyze application algorithm system-wide

Nsight Compute -
Debug/optimize CUDA kernel

Nsight Graphics -
Debug/optimize graphics workloads



Legacy Transition



Nsight Systems

Standalone GUI+CLI

- CPU-GPU interactions & triage
- Low overhead capture
- GPU compute & graphics
- Faster GUI + more data

Nsight Compute

Standalone GUI+CLI

- GPU CUDA kernel analysis & debug
- Very high freq GPU perf counters
- Compare results (diff)
- Incredible statistics & customizable

Overview

- System-wide application algorithm tuning
- Visualization
  - Locate optimization opportunities
  - Visualize millions of events on a very fast GUI timeline
  - See gaps of unused CPU and GPU time
- Balance your workload across multiple CPUs and GPUs
  - CPU algorithms, utilization, and thread state
  - GPU streams, kernels, memory transfers, etc
- Statistical analysis
  - Generate reports, output to additional processing applications
  - Direct SQLite access to data.

Timeline Trace Features

- **Compute**
  - **CUDA 9+ API & GPU workload ranges & mem transfers with correlation**
  - **Libraries: cuBLAS, cuDNN, cuDF, TensorRT, OpenACC, DirectML, MPI**
- **Graphics**
  - **Direct3D12, DXR, Direct3D11, WDDM, Vulkan, OpenGL**
  - **Long multi-iteration/frame and stutter analysis**
- **OS**
  - **Thread state and CPU utilization**
  - **OS runtime long call trace**
  - **ftrace or ETW (page faults, signal, interrupts, …)**
- **User Annotations APIs**
  - **NVTX, PIX, Vulkan debug markers, KHR_debug**

Other Key Features

- **Thread call-stack periodic sampling**
  - **Backtraces via hardware LBRs, frame pointers, or dwarf unwind**
  - **Hot functions**
- **Profiling start/stop using delay, duration, cudaProfilerStart/Stop API, NVTX annotations, or hot keys.**
- **Command Line Interface (CLI)**
  - **No host PC required to record**
  - **Works in containers, VMs, and systems with access limitations**
  - **Scriptable / interactive modes**
  - **Multiple profiling sessions can be run at once**
  - **Multiple report segments available.**

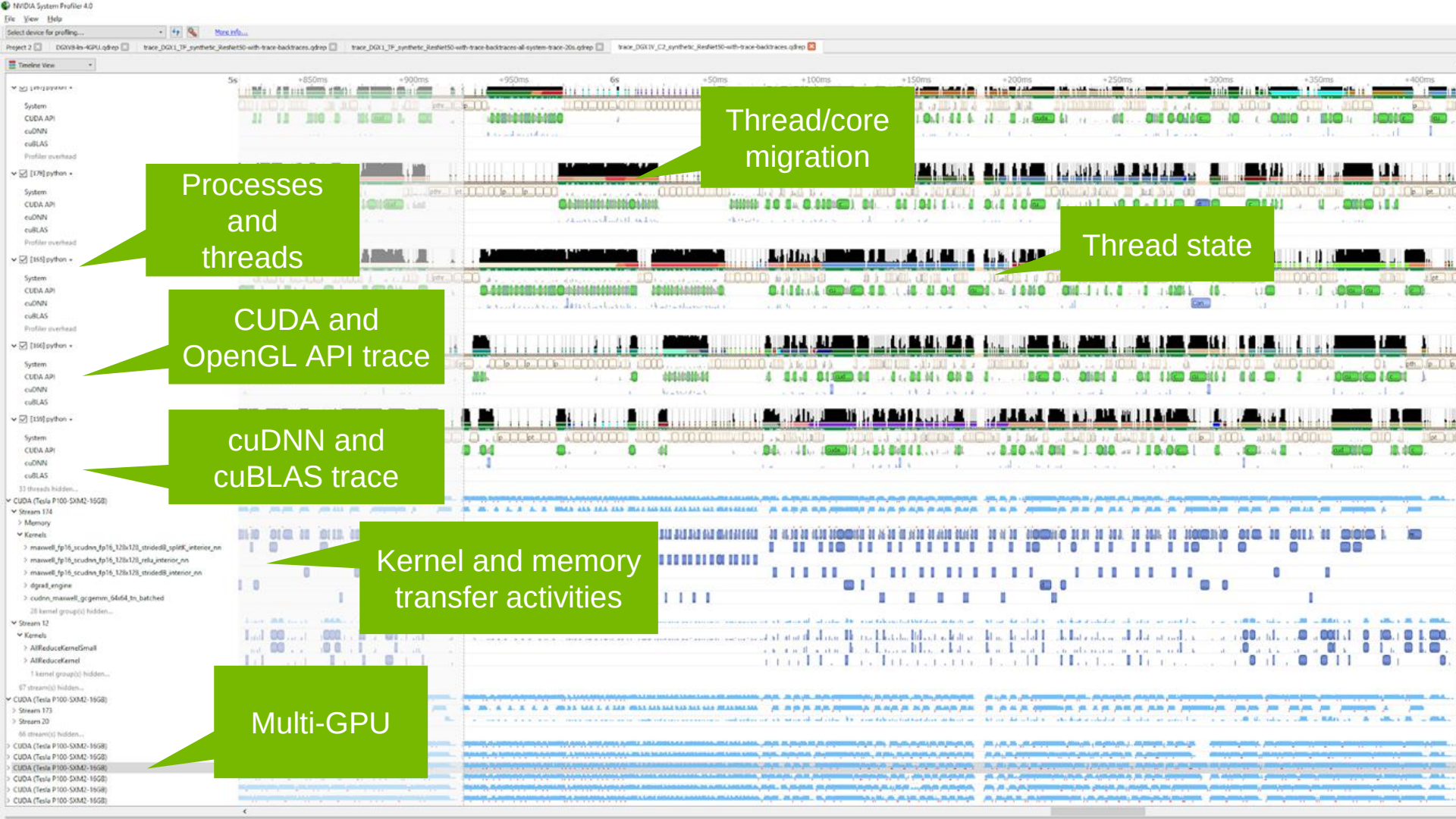
Available Platforms

OS/Arch	Host/GUI	Localhost	Target	CLI
MacOS	x			
Windows	x	x	x	x
Linux x86	x	x	x	x
Linux Power			x	x
Linux ARM Server			x	x
L4T/QNX for Tegra			x	x


NVIDIA
Demo

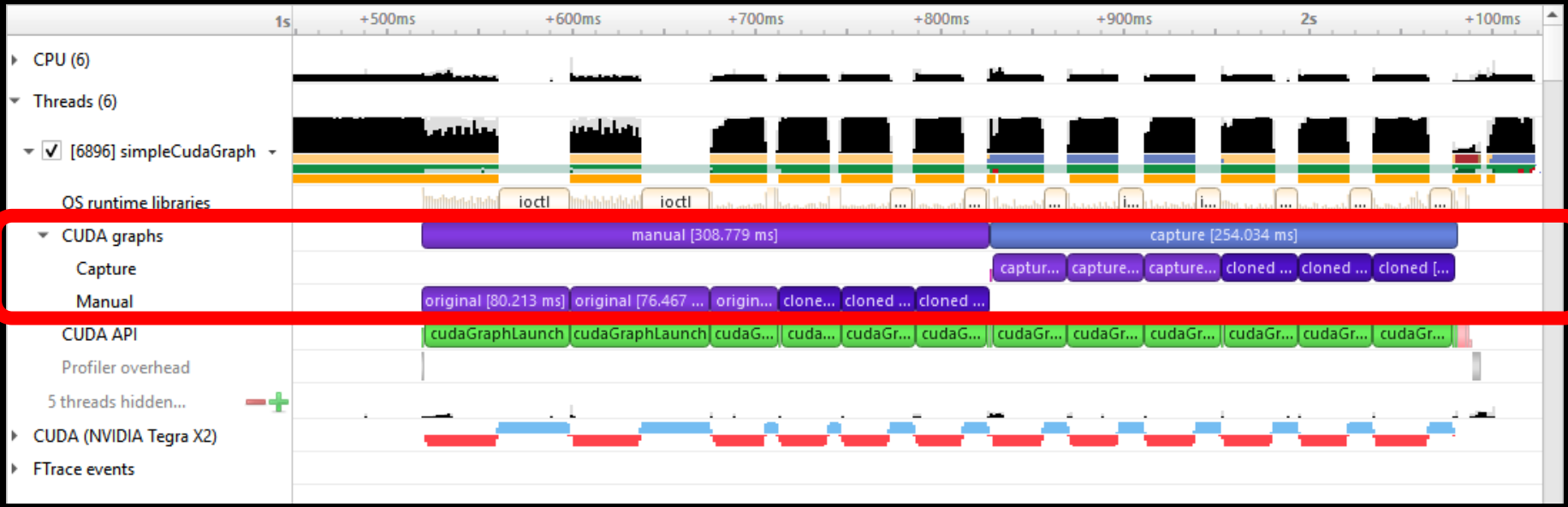
Feature Highlights

CPU and General


Thread/core migration
Processes and threads
Thread state
CUDA and OpenGL API trace
cuDNN and cuBLAS trace
Kernel and memory transfer activities
Multi-GPU

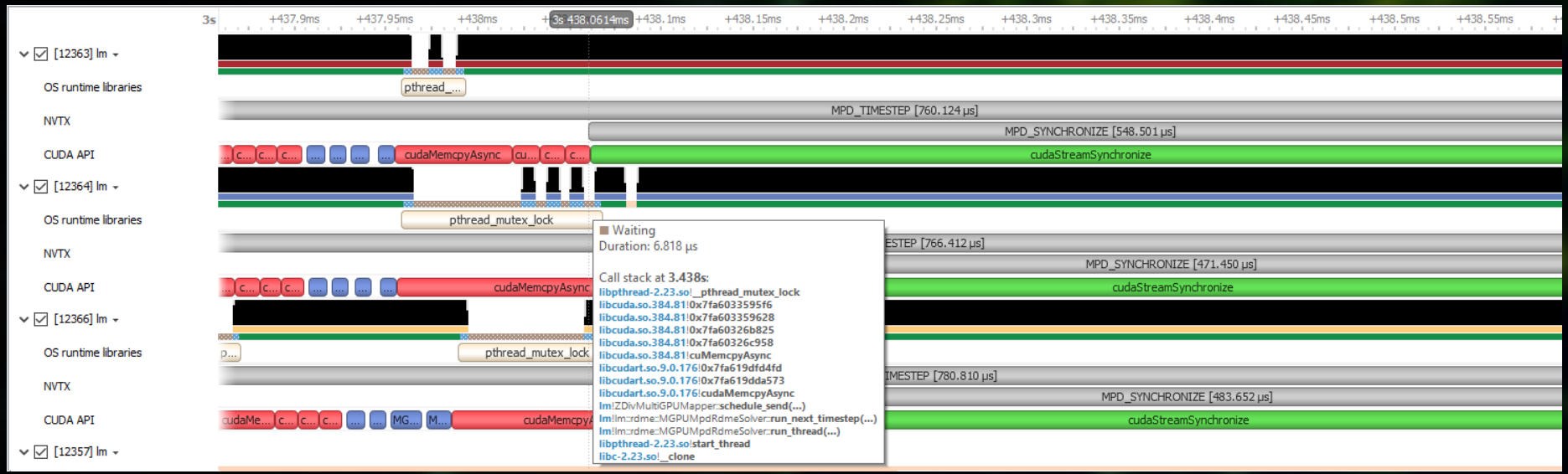
User annotations APIs for CPU & GPU
NVTX, OpenGL, Vulkan, and Direct3D performance markers

Example: Visual Molecular Dynamics (VMD) algorithms visualized with NVTX on CPU



NVTX Domains - Hoisting & Hierarchies

Look like APIs and '/' forms hierarchy



OS runtime trace (OSRT)
Includes backtraces of long running functions

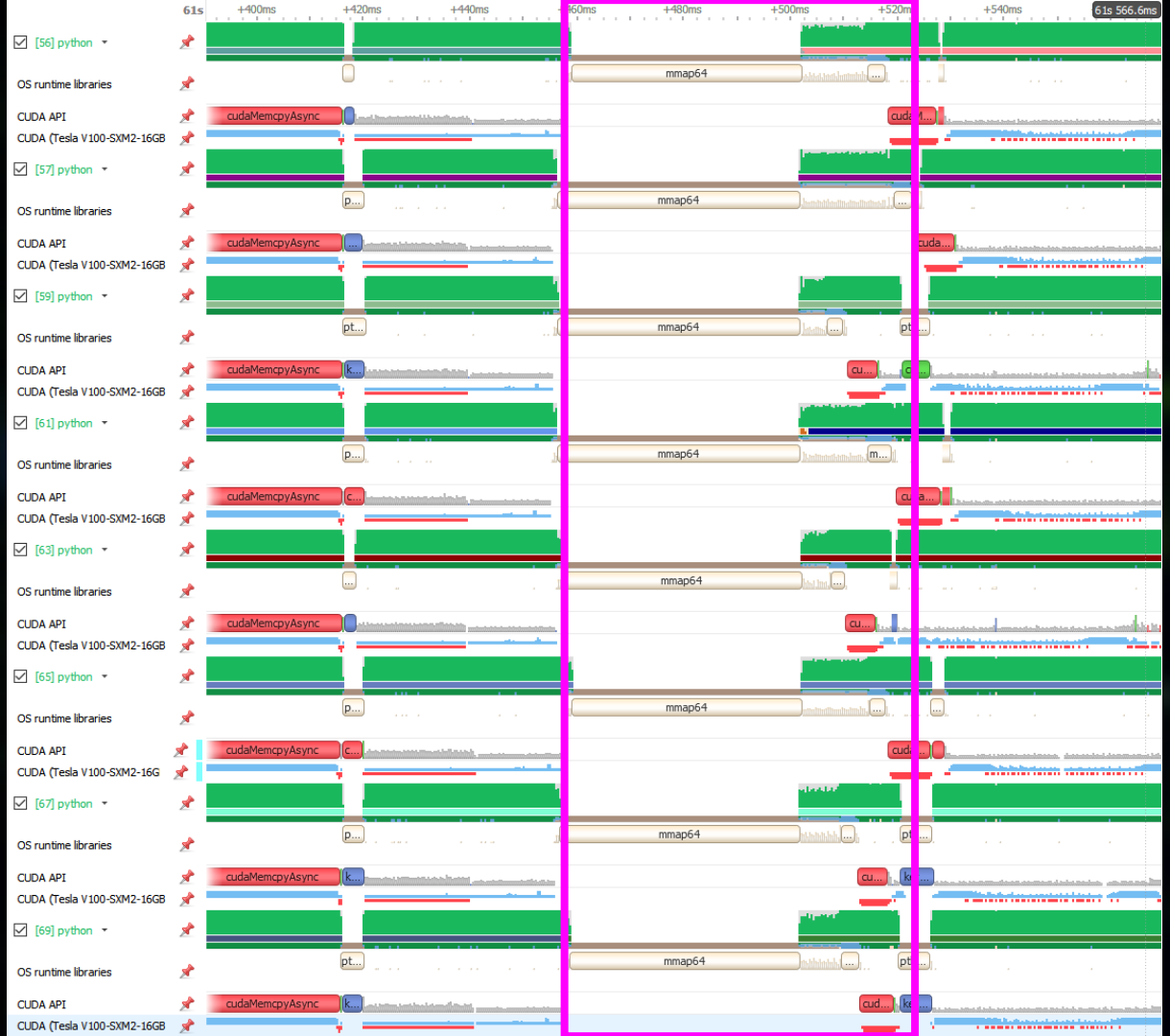
OS Runtime API Trace

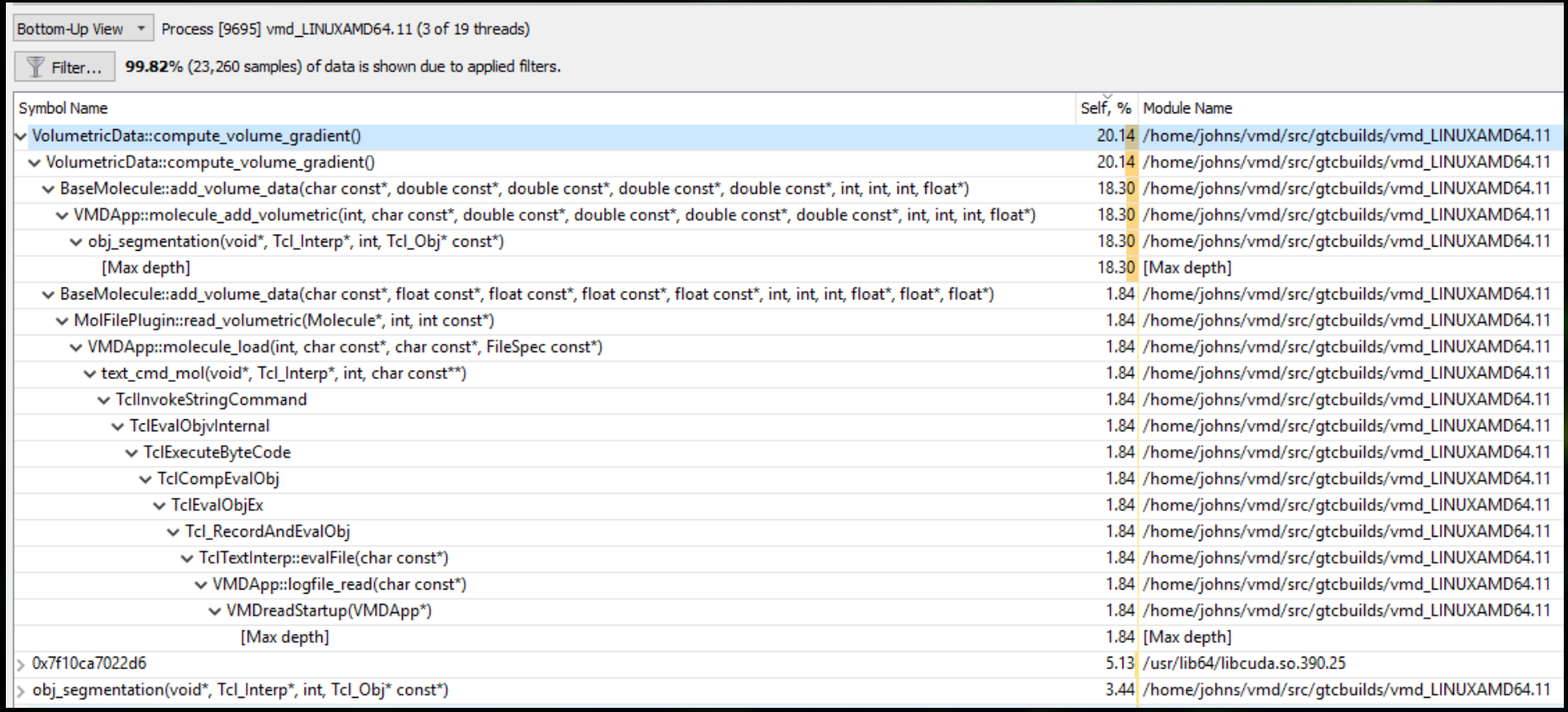
Example:Mask-RCNN

Map/unmap hiccups

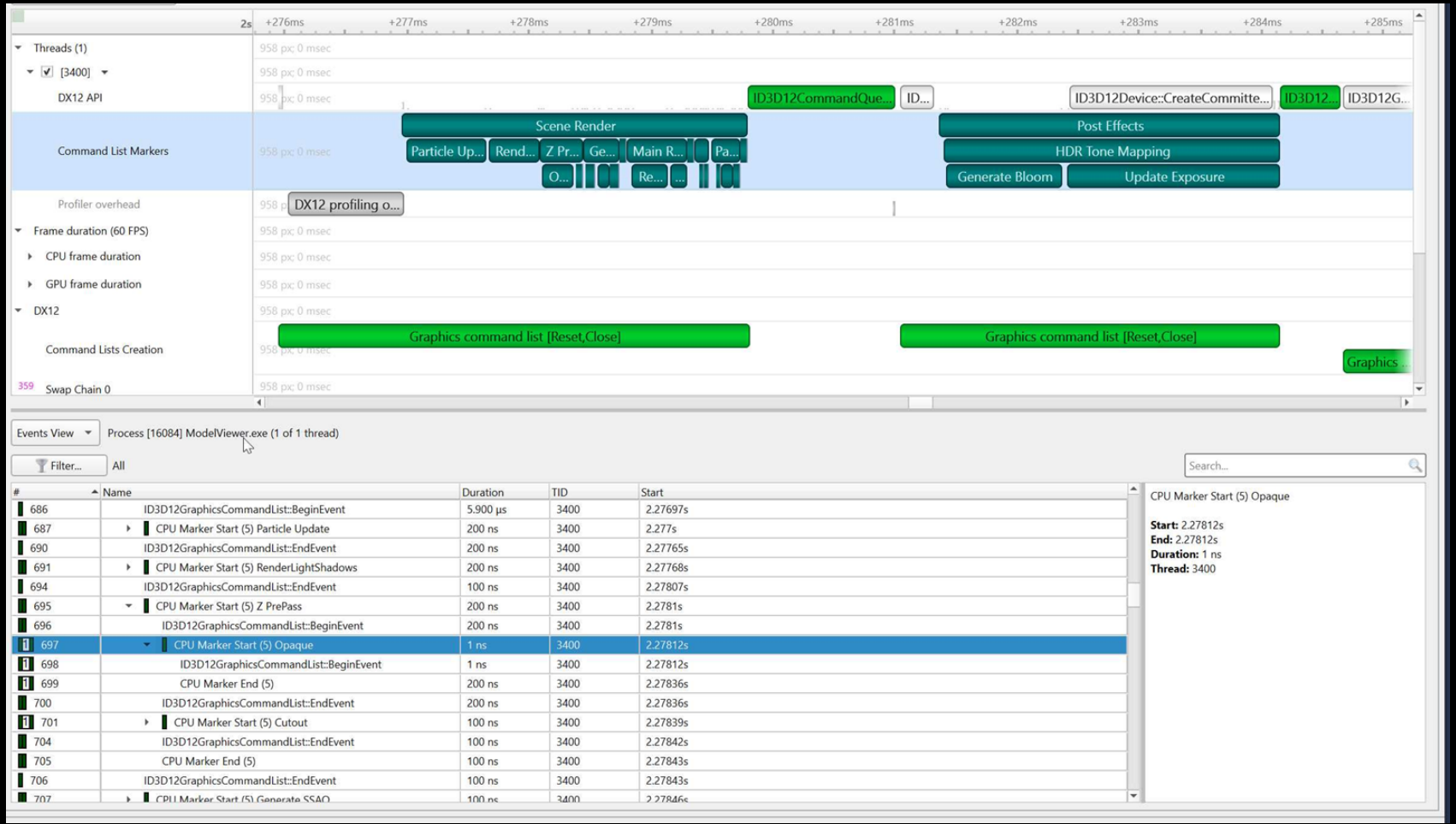
Mitigate by pipelining

- Map 1 batch ahead
- Unmap last batch
- Swap pointers here instead

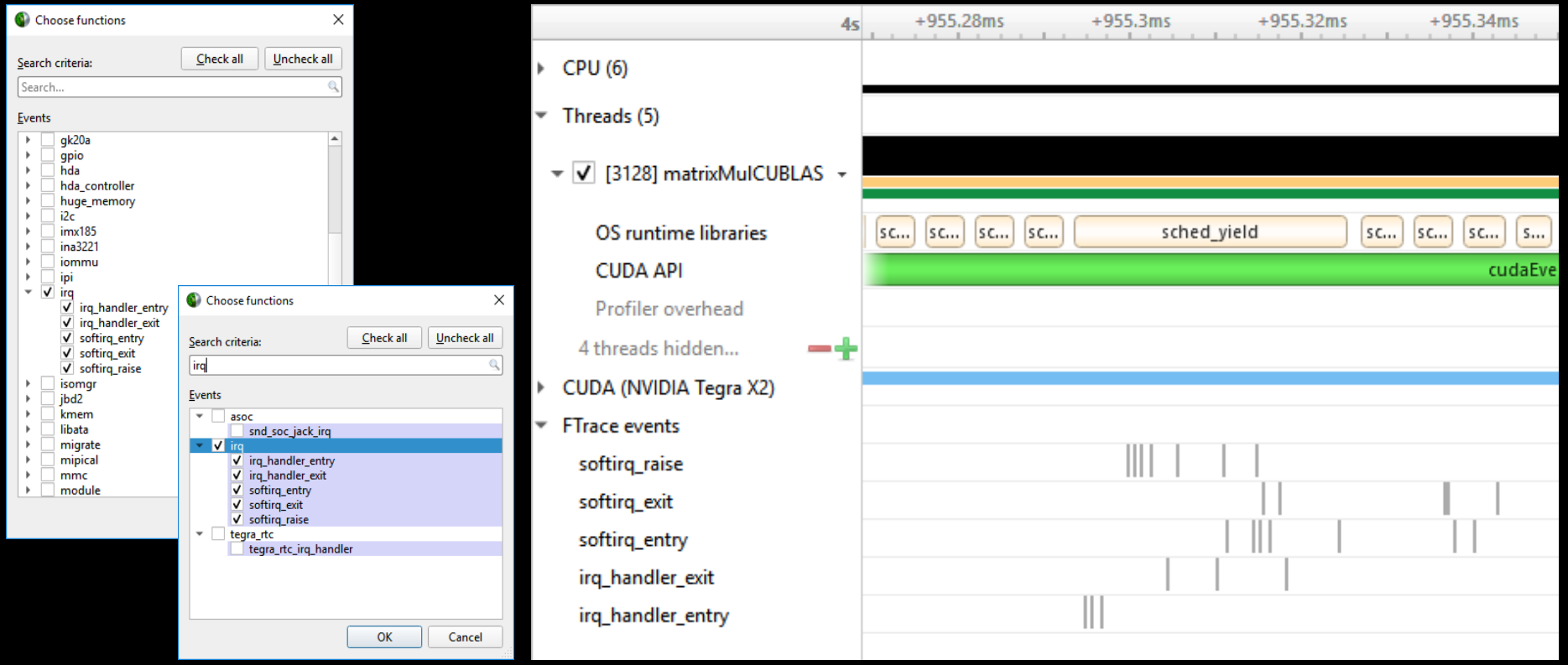




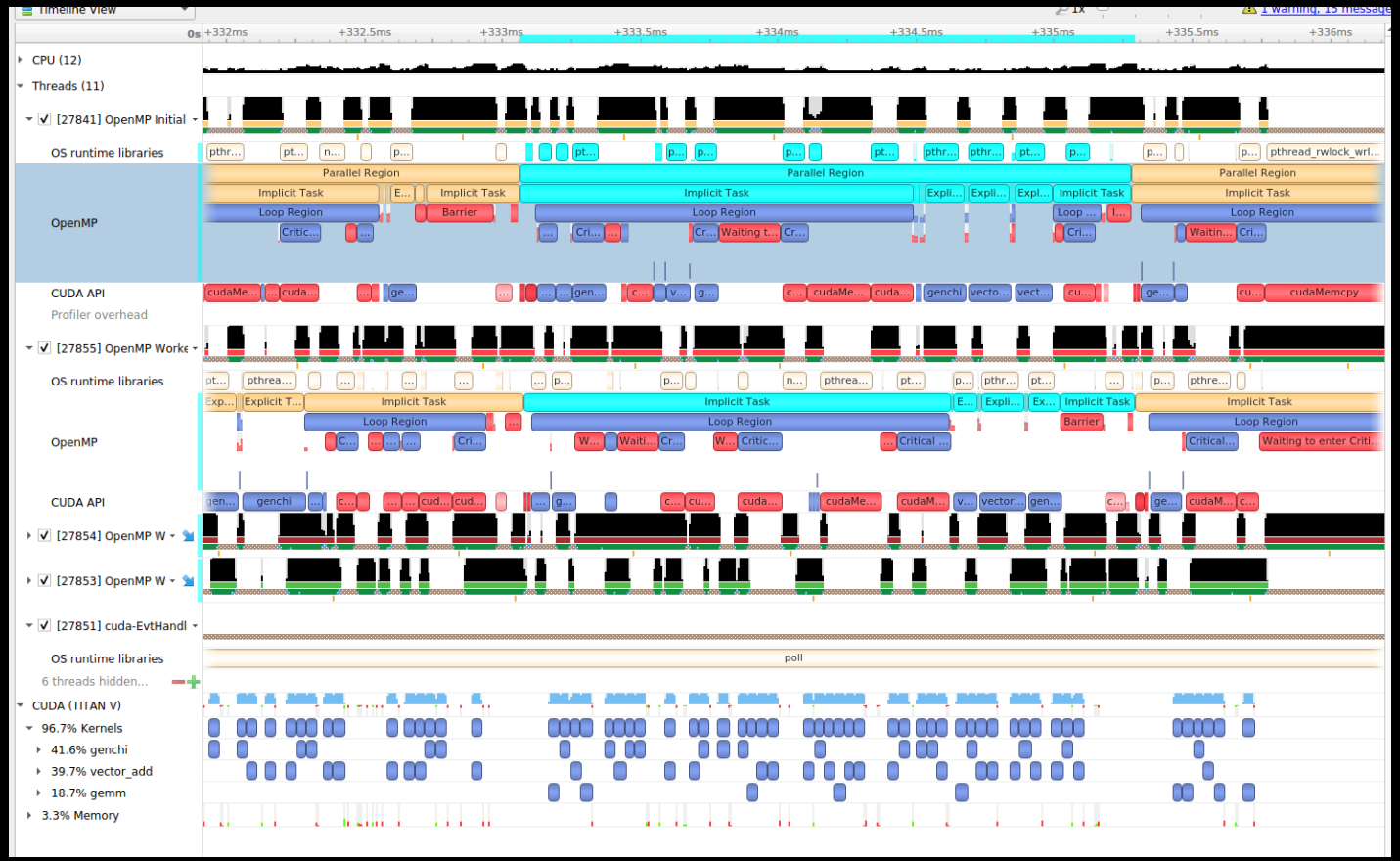
Function table shows statistics from periodic call-stack backtraces


2s
+276ms
+277ms
+278ms
+279ms
+280ms
+281ms
+282ms
+283ms
+284ms
+285ms
Threads (1)
[3400]
DX12 API
ID3D12CommandQue...
ID...
ID3D12Device::CreateCommitte...
ID3D12...
ID3D12G...
Command List Markers
Scene Render
Post Effects
Particle Up...
Rend...
Z Pr...
Ge...
Main R...
Pa...
HDR Tone Mapping
O...
Re...
Generate Bloom
Update Exposure
Profiler overhead
DX12 profiling o...
Frame duration (60 FPS)
CPU frame duration
GPU frame duration
DX12
Command Lists Creation
Graphics command list [Reset,Close]
Graphics command list [Reset,Close]
Graphics ...
359
Swap Chain 0
Events View
Process [16084] ModelViewer.exe (1 of 1 thread)
Filter...
All
Search...
Name Duration TID Start
686 ID3D12GraphicsCommandList::BeginEvent 5.900 µs 3400 2.27697s
687 CPU Marker Start (5) Particle Update 200 ns 3400 2.277s
690 ID3D12GraphicsCommandList::EndEvent 200 ns 3400 2.27765s
691 CPU Marker Start (5) RenderLightShadows 200 ns 3400 2.27768s
694 ID3D12GraphicsCommandList::EndEvent 100 ns 3400 2.27807s
695 CPU Marker Start (5) Z PrePass 200 ns 3400 2.2781s
696 ID3D12GraphicsCommandList::BeginEvent 200 ns 3400 2.2781s
697 CPU Marker Start (5) Opaque 1 ns 3400 2.27812s
698 ID3D12GraphicsCommandList::BeginEvent 1 ns 3400 2.27812s
699 CPU Marker End (5) 200 ns 3400 2.27836s
700 ID3D12GraphicsCommandList::EndEvent 200 ns 3400 2.27836s
701 CPU Marker Start (5) Cutout 100 ns 3400 2.27839s
704 ID3D12GraphicsCommandList::EndEvent 100 ns 3400 2.27842s
705 CPU Marker End (5) 100 ns 3400 2.27843s
706 ID3D12GraphicsCommandList::EndEvent 100 ns 3400 2.27843s
707 CPU Marker Start (5) Generate SSAO 100 ns 3400 2.27846s
CPU Marker Start (5) Opaque
Start: 2.27812s
End: 2.27812s
Duration: 1 ns
Thread: 3400

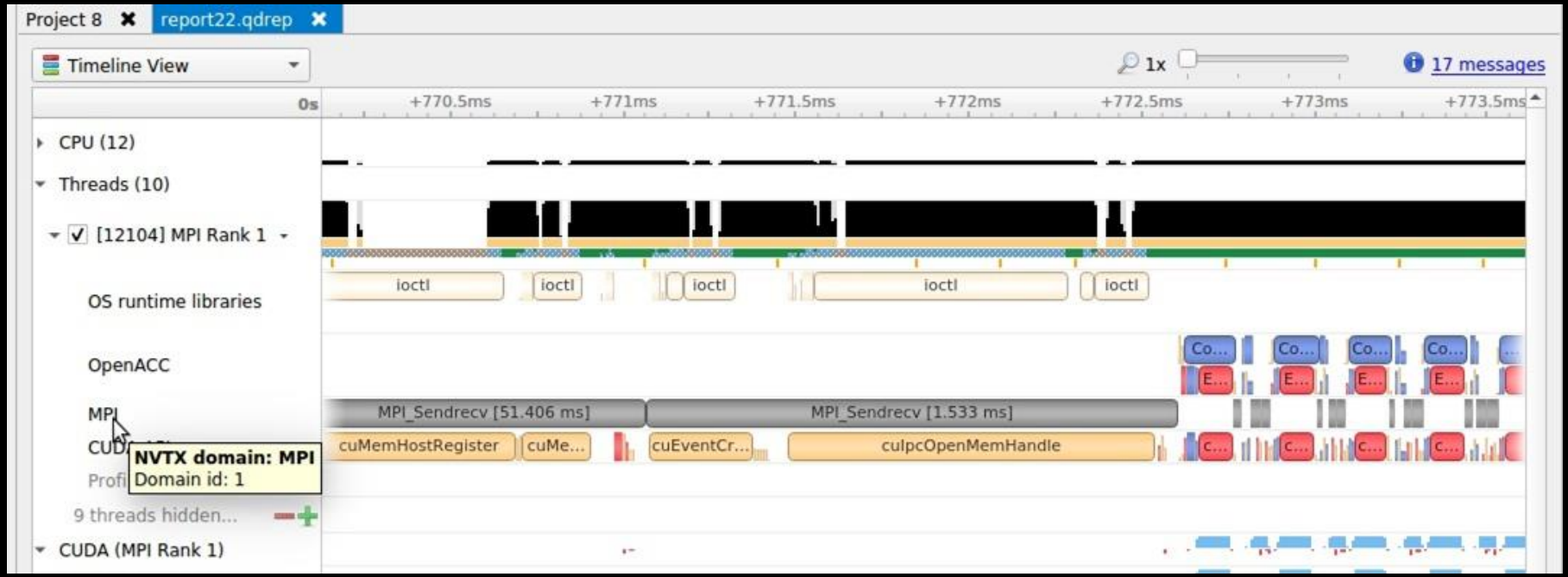
Event Table



FTrace
Example demonstrates interrupts



OpenMP5+ trace


Project 8
report22.qdrep
Timeline View
1x
17 messages
0s
+770.5ms
+771ms
+771.5ms
+772ms
+772.5ms
+773ms
+773.5ms
CPU (12)
Threads (10)
[12104] MPI Rank 1
OS runtime libraries
OpenACC
MPI
CUDA
Profi
9 threads hidden...
CUDA (MPI Rank 1)
ioctl
ioctl
ioctl
ioctl
ioctl
Co...
E...
MPI_Sendrecv [51.406 ms]
MPI_Sendrecv [1.533 ms]
cuMemHostRegister
cuMe...
cuEventCr...
cuIpcOpenMemHandle
NVTX domain: MPI
Domain id: 1

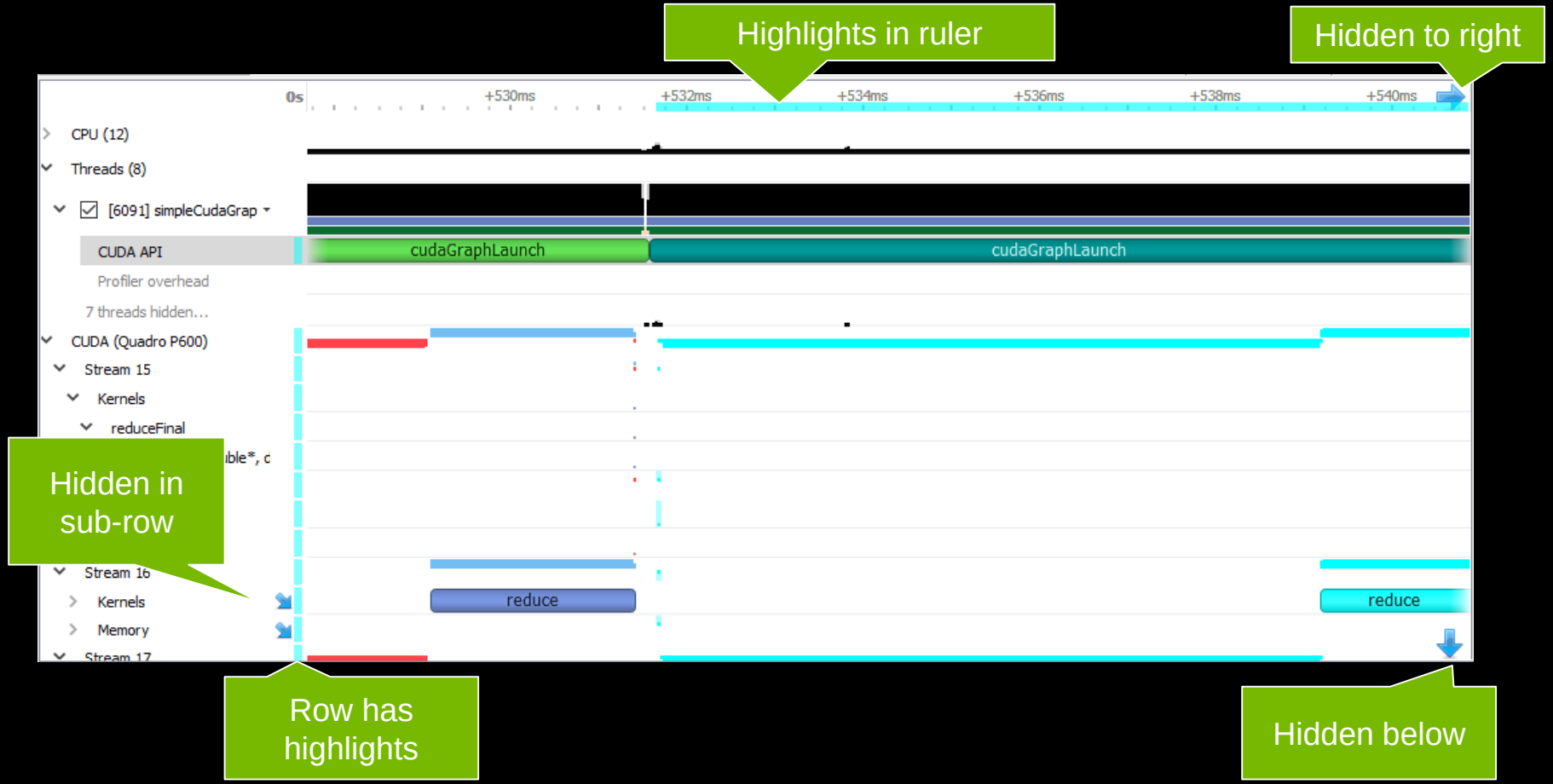
MPI trace

TensorFlow Resnet50 DNN nodes
as NVTX ranges projected onto the GPU

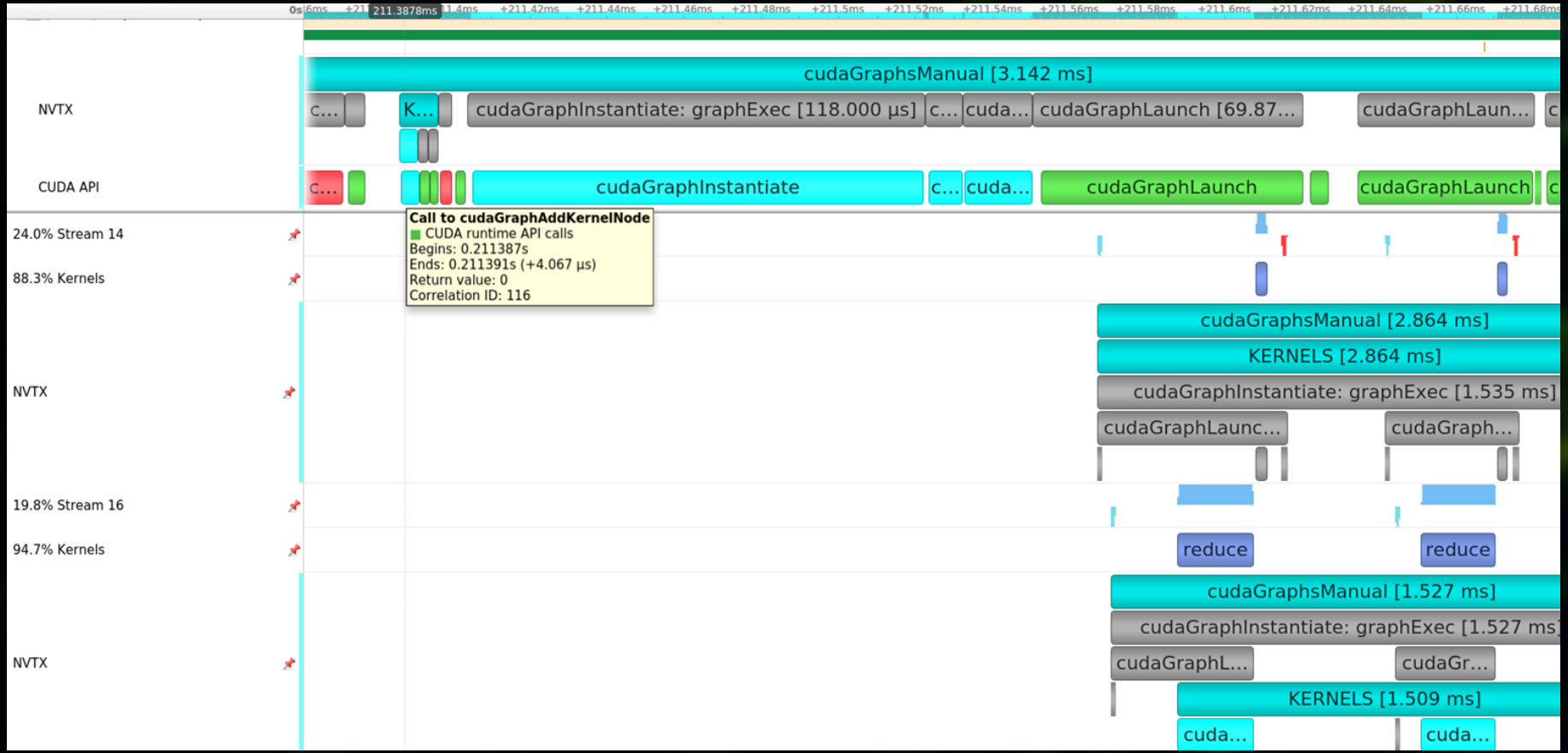
Feature Highlights

Compute Applications

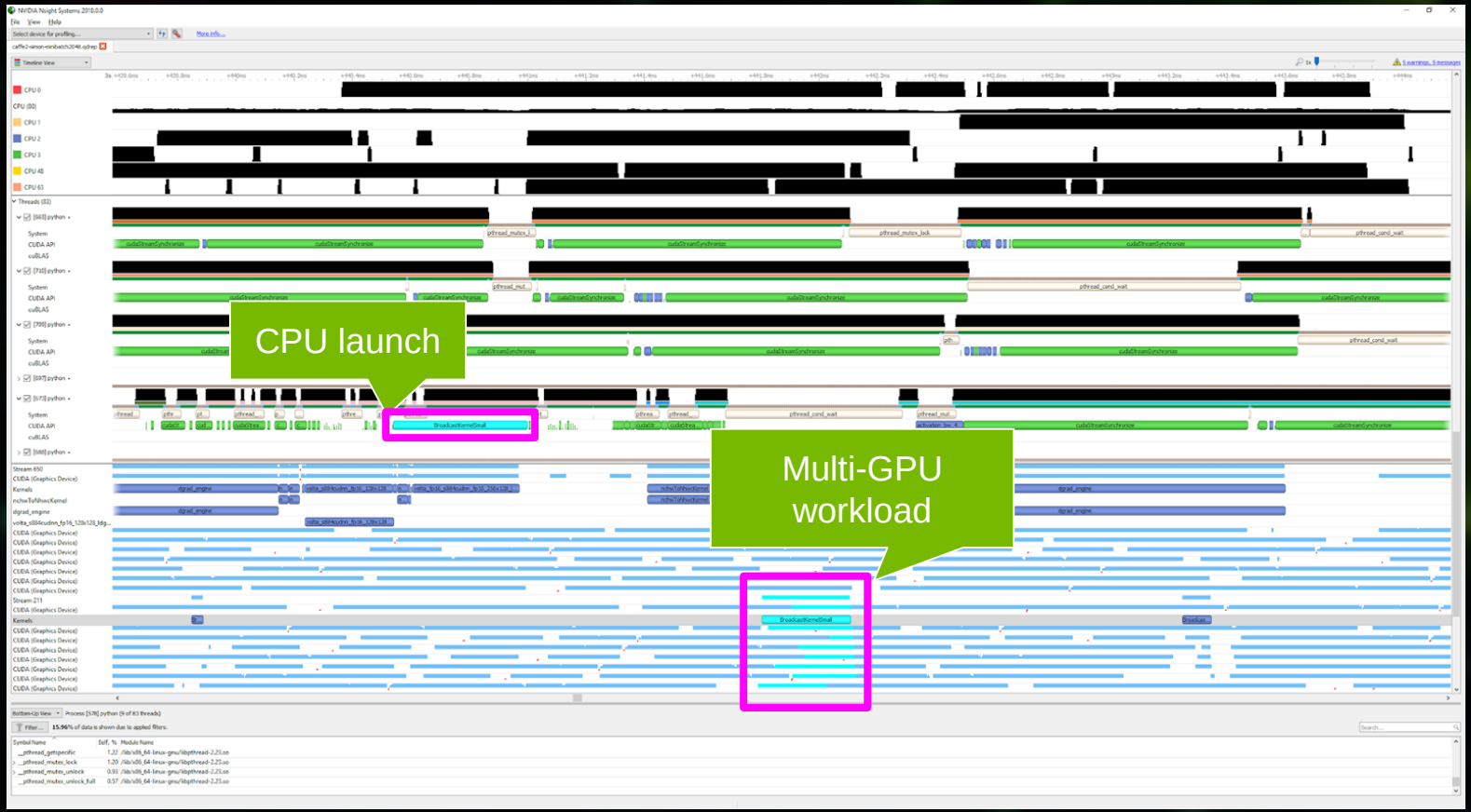
GPU API launch to HW workload correlation



CPU-GPU correlation & location assistance

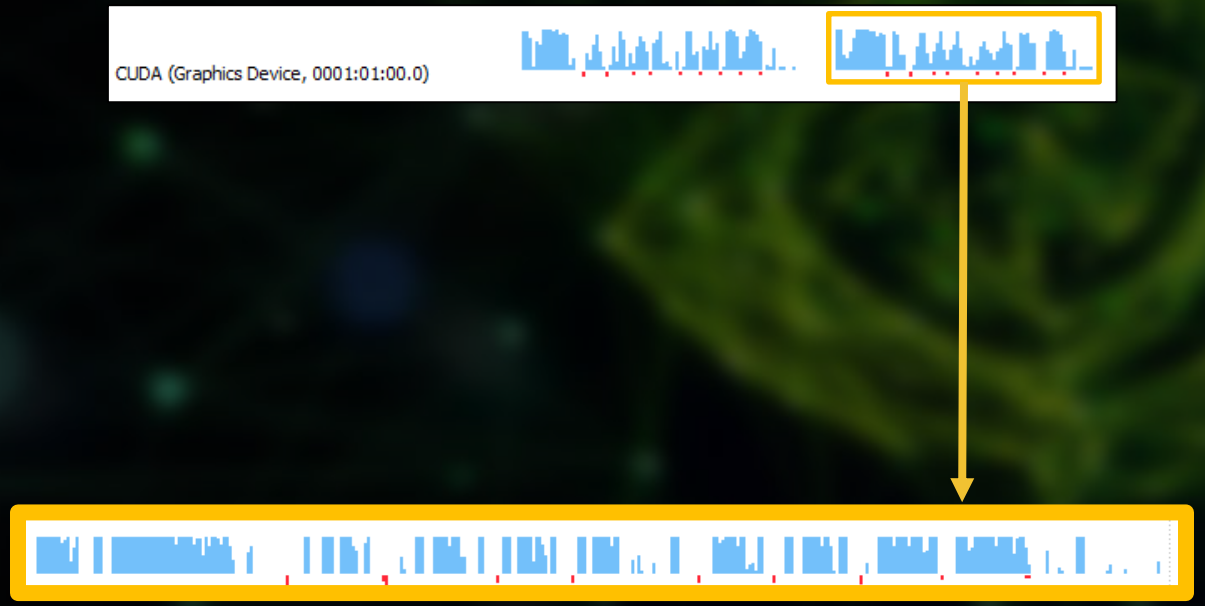


CUDA graph launches show all related GPU ranges
NVTX around nodes is also projected onto the GPU CUDA stream

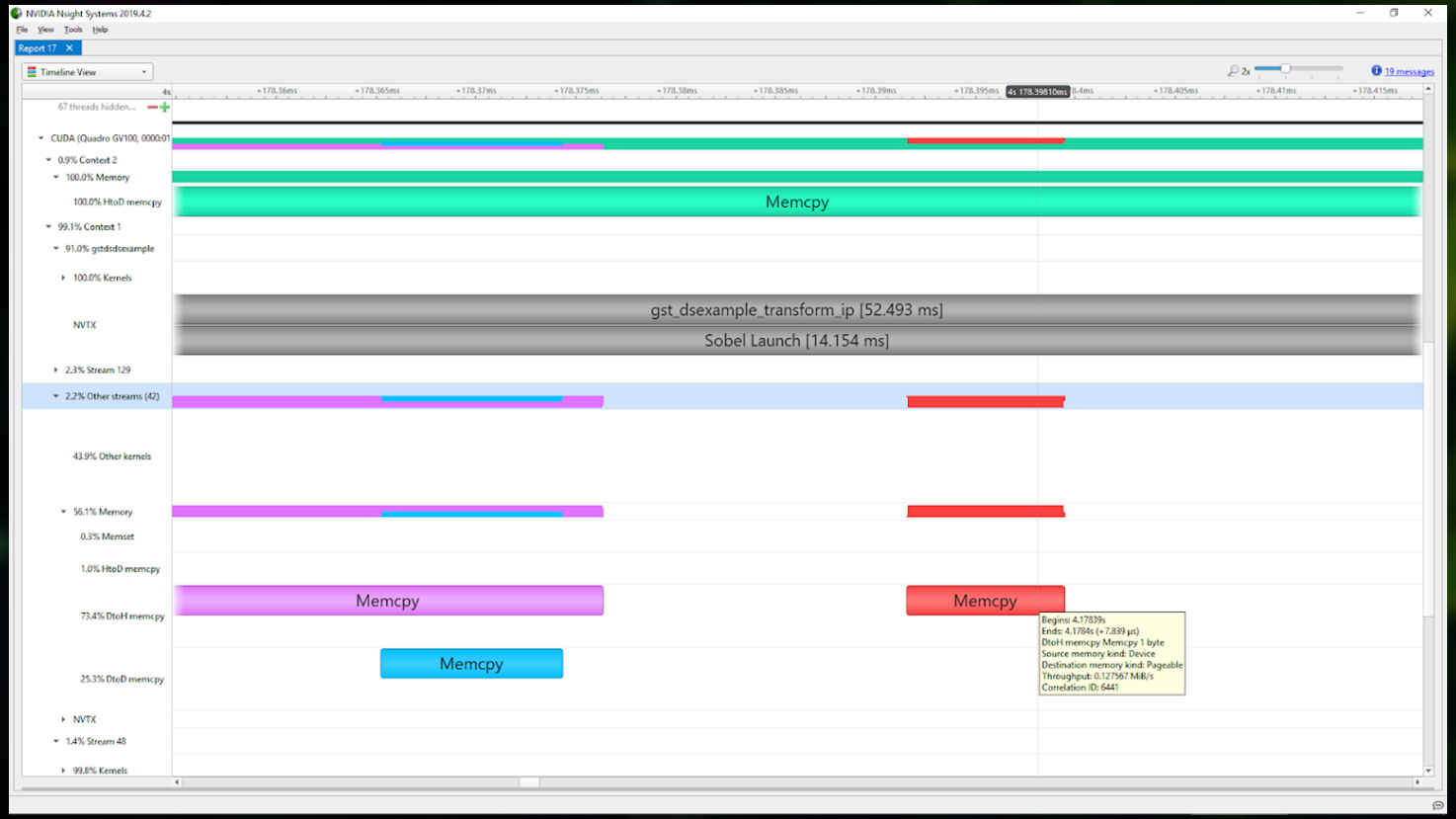


cudaLaunchCooperativeKernelMultiDevice
From Caffe2 Resnet50 within a container on a DGX-2

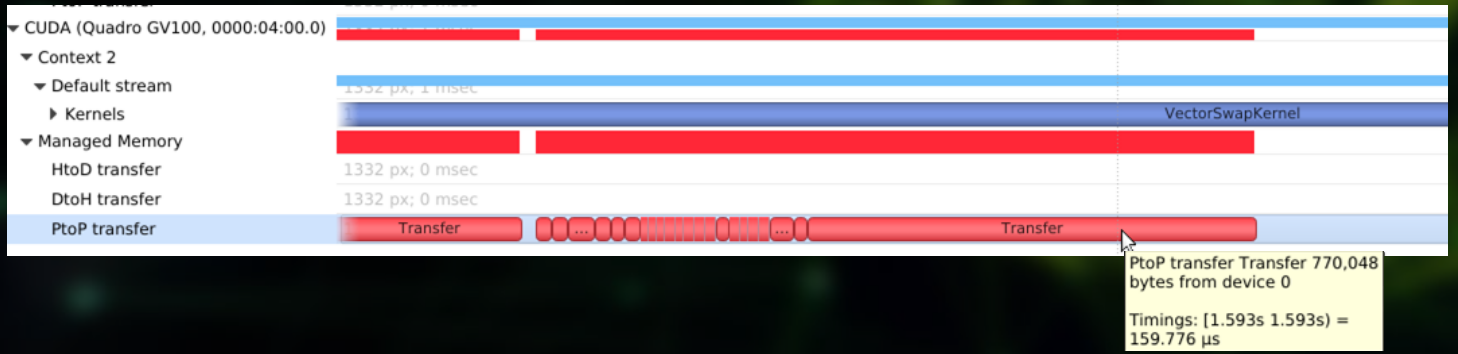
GPU utilization based on percentage time coverage



Zooming in reveals gaps where there were valleys



CUDA memory transfer color pallette
show direction and pageable memory hazards



CUDA unified virtual memory (UVM) transfers

CUDA library trace such as cuDNN and cuBLAS


CLI statistics and export
(SQLite & JSON)

Time(%)	Time (ns)	Calls	Avg (ns)	Min (ns)	Max (ns)	Name
25.3	96449693309	492166	195969.8	2590	11910256	cudaStreamSynchronize
17.3	65985078126	1337536	49333.3	6121	207018385	cudaLaunchKernel
15.3	58387659442	303066	192656.6	7710	126095339	cudaMemcpyPeerAsync
13.8	52570248446	88	597389186.9	79402	1109295234	cudaHostAlloc
7.8	29692609029	174420	170236.3	5770	133461628	cudaMemcpyAsync
5.0	19201650230	4717	4070733.6	8389	657446238	cudaMalloc
3.2	12177533056	306181	39772.3	901	124721678	cudaEventRecord
3.1	11678553105	108375	107760.6	5751	10799852	cudaMemcpy2DAsync
2.8	10525939137	120	87716159.5	2445	2684308531	cudaStreamCreateWithFlags
2.2	8474469667	78859	107463.6	6956	97743689	cudaMemsetAsync
1.5	5643462857	10	564346285.7	10283	2824251257	cudaStreamCreate
1.2	4409251983	133120	33122.4	10932	3830401	cudaLaunchCooperativeKernel
0.8	2932703617	441	6650121.6	669	727451161	cudaFree
0.6	2202242513	165248	13326.9	3346	27369081	cudaEventSynchronize
0.1	361857328	145933	2479.6	611	3467557	cudaStreamWaitEvent
0.1	210636728	19	11086143.6	88975	76110537	cudaMallocHost
0.0	16891690	864	19550.6	706	13229050	cudaEventCreateWithFlags
0.0	6833490	1256	5440.7	3696	20444	cudaEventQuery
0.0	1597827	32	49932.1	2931	366719	cudaStreamCreateWithPriority
0.0	1064579	38	28015.2	15625	72513	cudaMemcpy
0.0	1023364	184	5561.8	2589	28453	cudaBindTexture
0.0	376511	184	2046.3	1004	20137	cudaUnbindTexture
0.0	319084	81	3939.3	952	75350	cudaEventCreate
0.0	275150	138	1993.8	949	10096	cudaEventDestroy

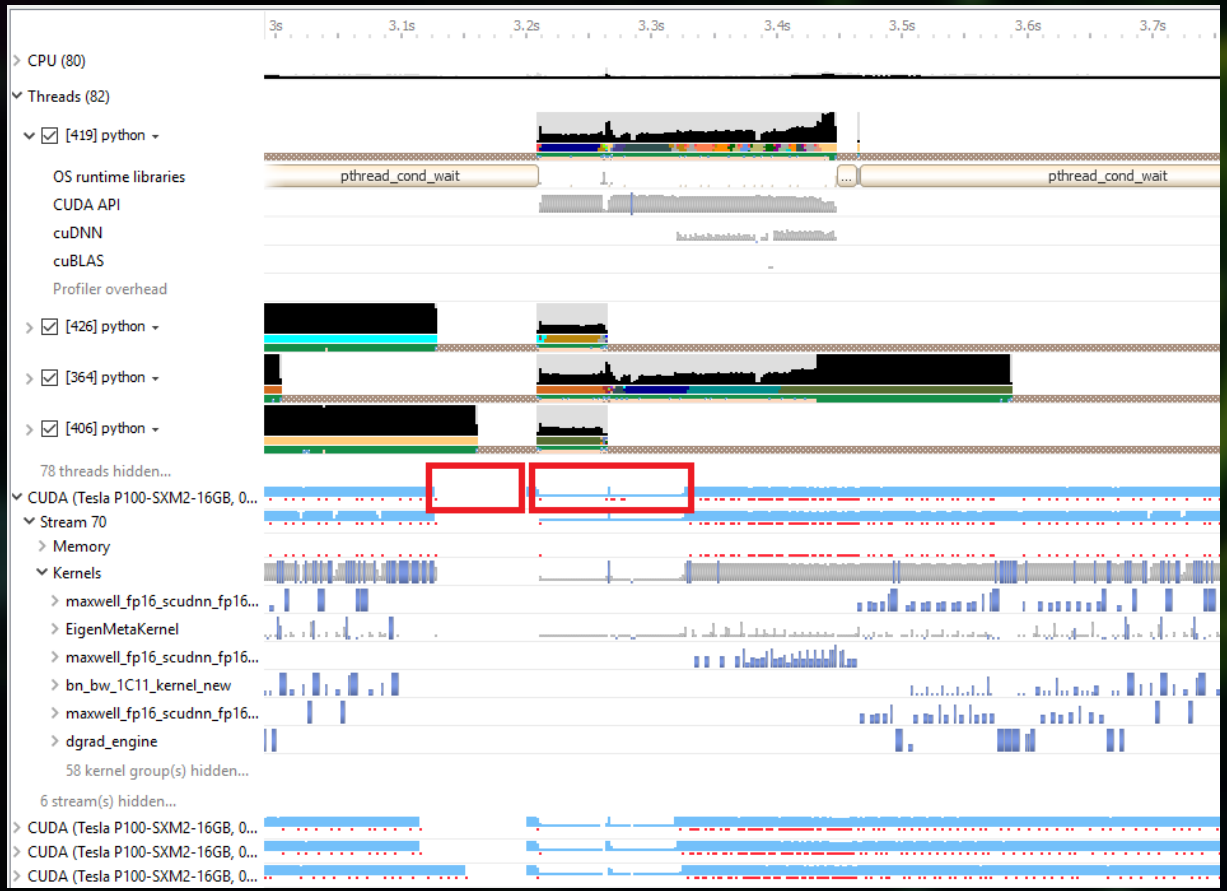
Stats/Export - CUDA API summary

Time(%)	Time (ns)	Instances	Avg (ns)	Min (ns)	Max (ns)	Range
4.0	162789746	70	2325567.8	248462	10421312	TensorRT:ExecutionContext::enqueue
0.4	16335803	24	680658.5	94707	2036983	TensorRT:predictions
0.3	10447691	70	149252.7	16094	3604518	TensorRT:conv1 + activation_1/Relu
0.2	10169055	70	145272.2	25936	1169143	TensorRT:conv1 + activation_1/Relu input reformatter 0
0.2	8698016	24	362417.3	9581	2970733	TensorRT:block_3b_conv_2
0.2	8625565	24	359398.5	28257	1242590	TensorRT:average_pooling2d_1
0.2	7733550	46	168120.7	30474	3498224	TensorRT:conv2d_cov/Sigmoid
0.2	6672310	70	95318.7	11271	1151050	TensorRT:block_1a_conv_1 + activation_2/Relu
0.1	5312256	70	75889.4	8316	2133255	TensorRT:block_4a_conv_2
0.1	5197452	24	216560.5	9572	1543570	TensorRT:block_3b_conv_1 + activation_12/Relu
0.1	4922521	70	70321.7	11732	1281643	TensorRT:block_1a_conv_shortcut + add_1 + activation_3/Relu
0.1	4151987	24	172999.5	10592	1625630	TensorRT:block_1b_conv_2
0.1	4121931	70	58884.7	10616	1307464	TensorRT:block_2a_conv_2
0.1	4108970	70	58699.6	9311	834374	TensorRT:block_1a_conv_2
0.1	4027181	24	167799.2	10564	1728250	TensorRT:block_1b_conv_shortcut + add_2 + activation_5/Relu
0.1	4018713	24	167446.4	9636	2431179	TensorRT:block_2b_conv_1 + activation_8/Relu
0.1	3799363	70	54276.6	8595	481048	TensorRT:block_3a_conv_2
0.1	3762886	24	156786.9	13969	1008096	TensorRT:block_4b_conv_shortcut + add_8 + activation_17/Relu
0.1	3376998	24	140708.2	12773	1641372	TensorRT:block_4b_conv_2
0.1	3361499	24	140062.5	11195	1248129	TensorRT:block_2b_conv_2
0.1	3227736	24	134489.0	15949	736287	TensorRT:predictions/Softmax
0.1	3023238	24	125968.2	11972	827474	TensorRT:block_2a_conv_shortcut + add_3 + activation_7/Relu
0.1	2998494	24	124937.2	12327	1217984	TensorRT:block_2a_conv_1 + activation_6/Relu
0.1	2853998	46	62043.4	10971	1983123	TensorRT:block_3a_conv_shortcut + add_3 + activation_7/Relu
0.1	2773491	24	115562.1	12019	809395	TensorRT:block_4a_conv_shortcut + add_7 + activation_15/Relu
0.1	2744711	24	114363.0	10474	892787	TensorRT:block_2b_conv_shortcut + add_4 + activation_9/Relu
0.1	2147200	24	89466.7	12037	645408	TensorRT:block_3a_conv_1 + activation_10/Relu
0.0	1914194	24	79758.1	10599	842947	TensorRT:block_4b_conv_1 + activation_16/Relu
0.0	1732011	24	72167.1	9659	488562	TensorRT:block_3a_conv_shortcut + add_5 + activation_11/Relu
0.0	1675232	24	69801.3	10073	623120	TensorRT:block_4a_conv_1 + activation_14/Relu
0.0	1525283	24	63553.5	10386	540733	TensorRT:block_3b_conv_shortcut + add_6 + activation_13/Relu
0.0	1358781	46	29538.7	10645	241099	TensorRT:block_3a_conv_1 + activation_6/Relu
0.0	1301126	46	28285.3	11929	244583	TensorRT:conv2d_cov

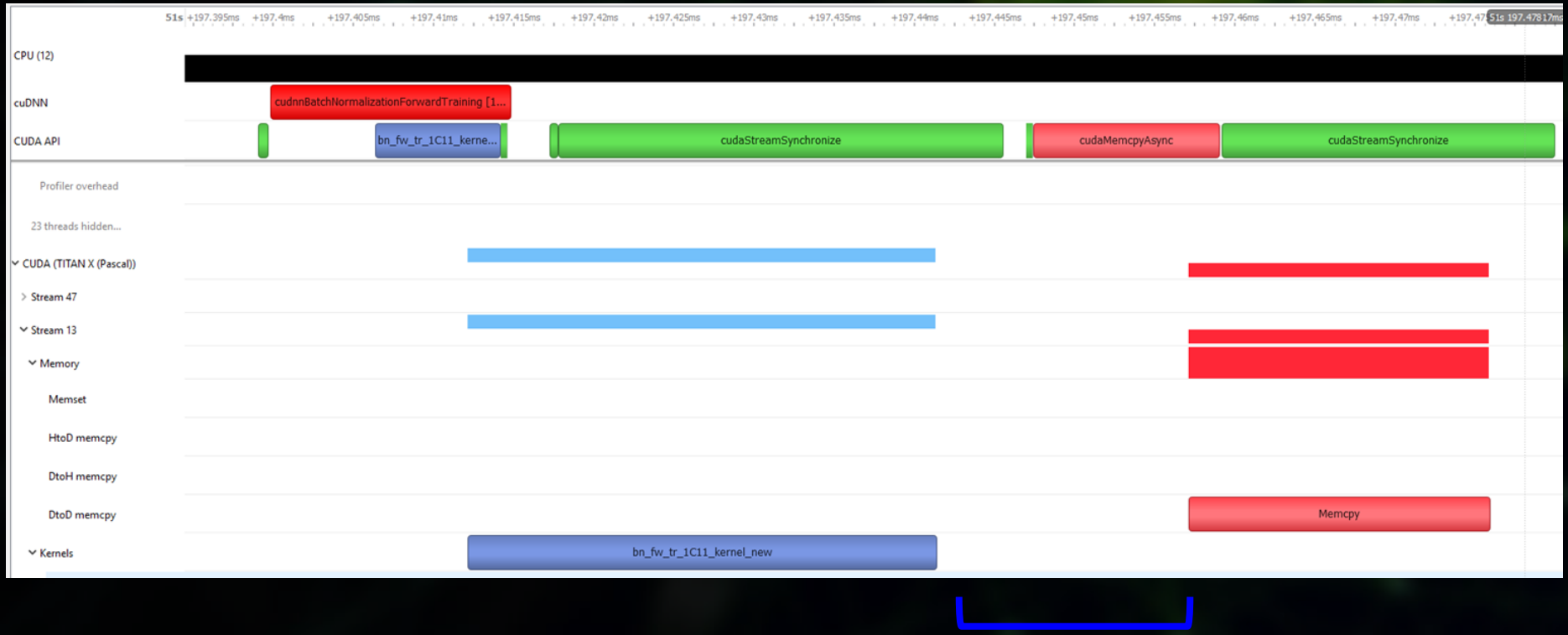
Stats/Export - NVTX code annotations

Note this includes TensorRT domains

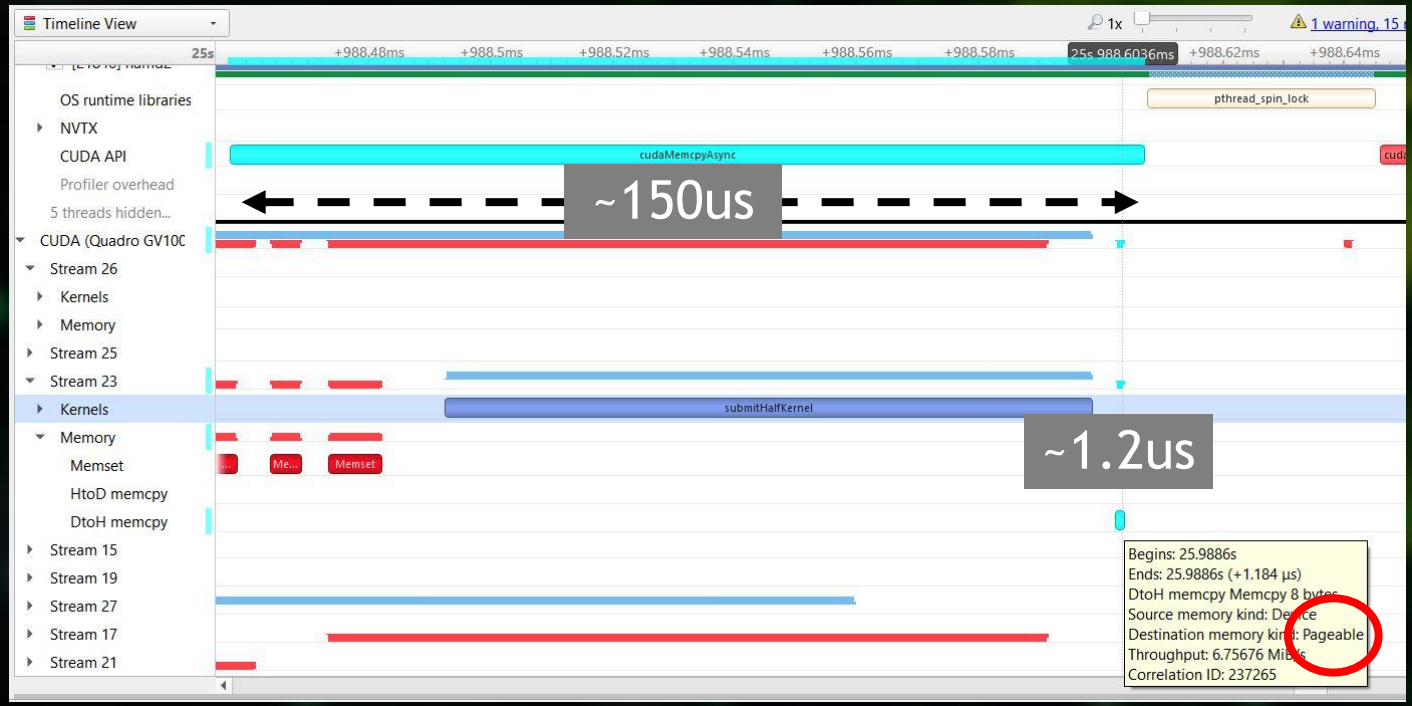

NVIDIA
Spotting Common Issues



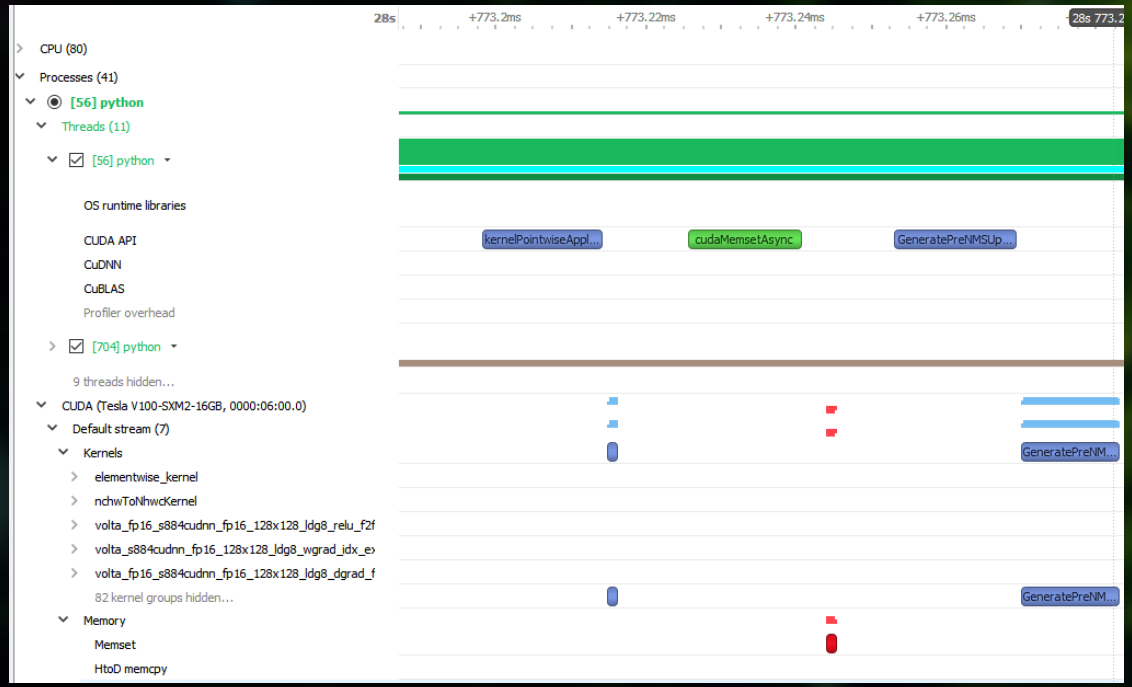
GPU idle and low utilization level of detail



Example GPU idle caused by stream synchronization



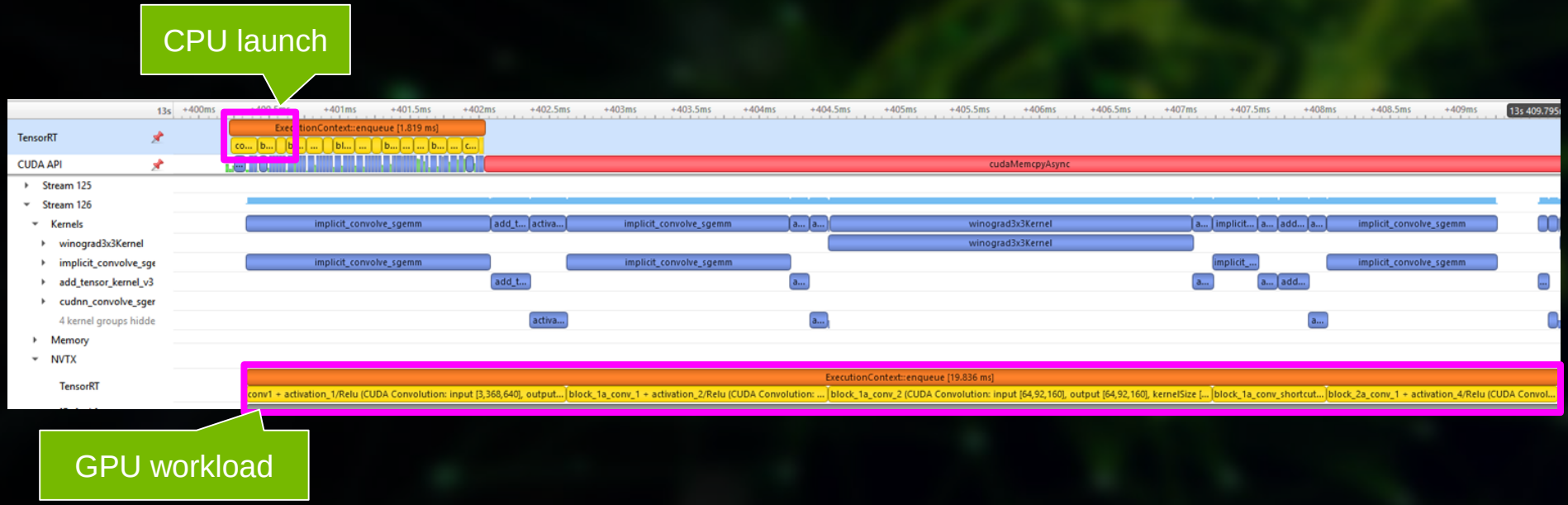
cudaMemcpyAsync behaving synchronous
Device to host pageable memory
Mitigate with pinned memory



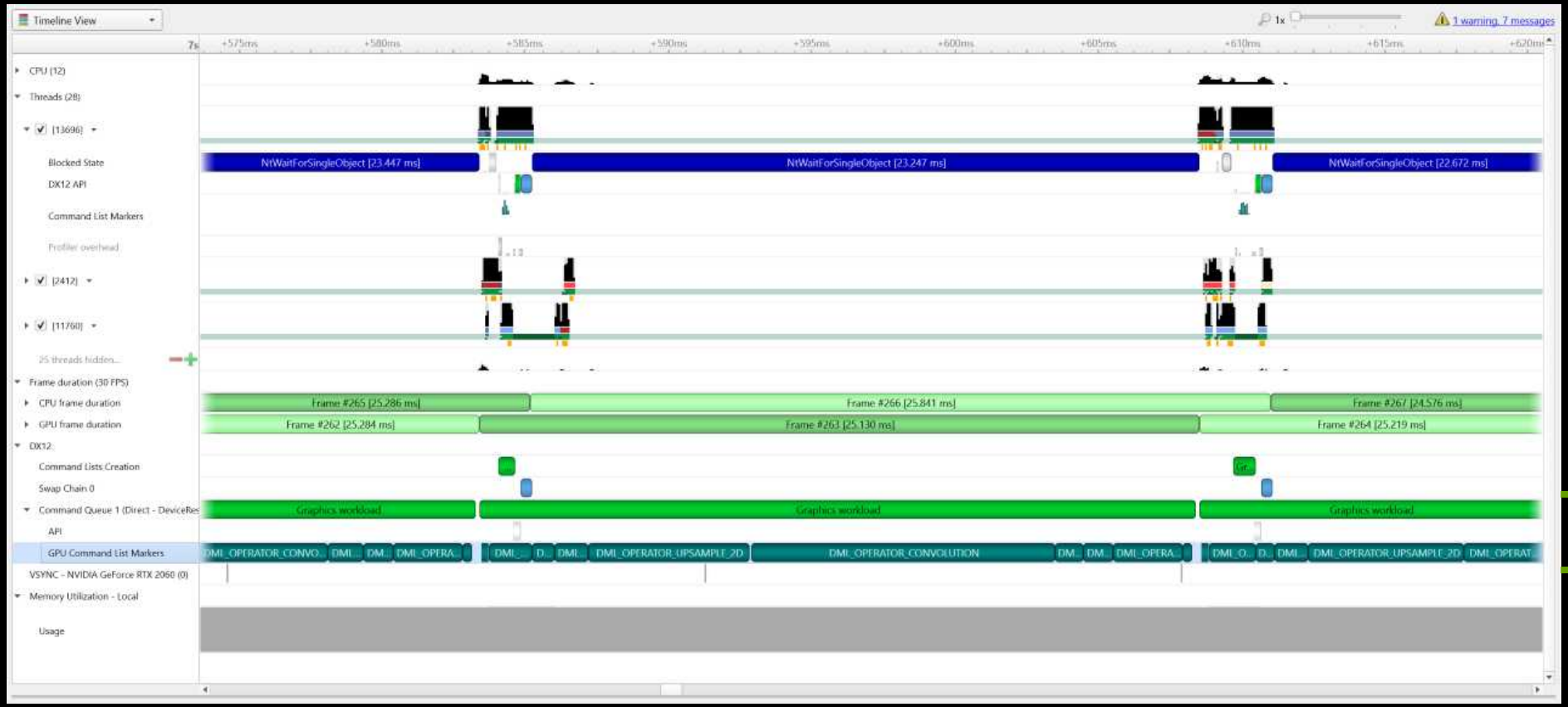
Fusion opportunities

CPU launch cost + small GPU work size ≈ GPU sparse idle

This can apply to DNN nodes/layers



NVTX ranges projected onto the GPU
From DeepStream



Windows DirectML

See graphics applications for more related info

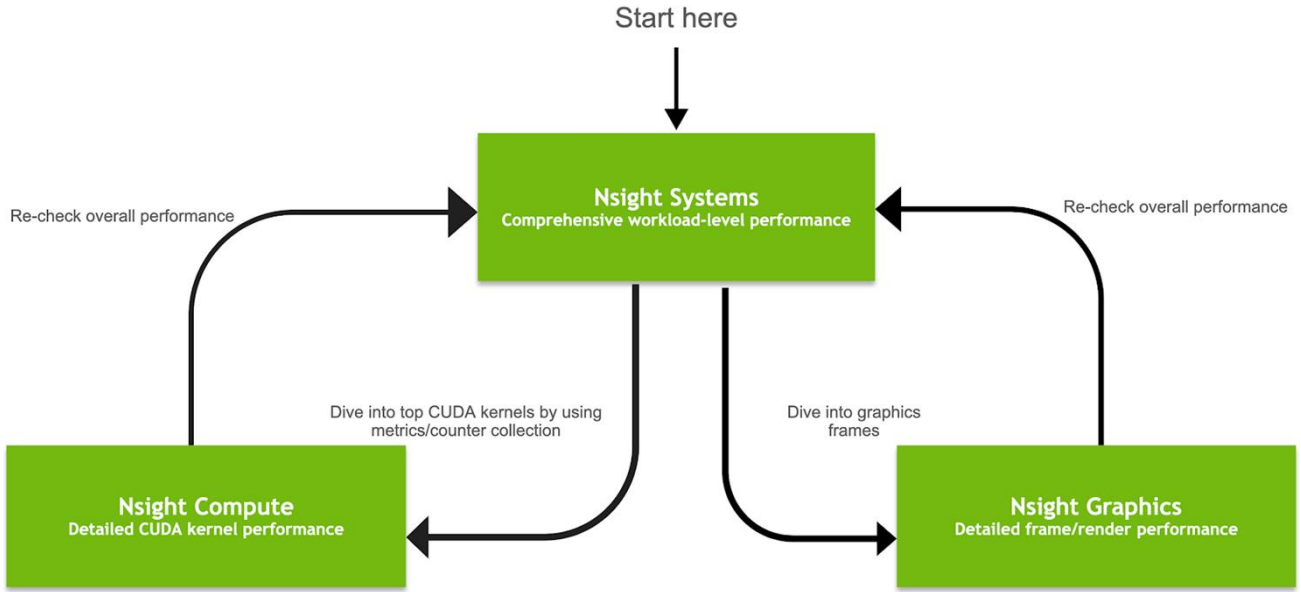
Nsight Product Family

Workflow

Nsight Systems -
Analyze application algorithm system-wide

Nsight Compute -
Debug/optimize CUDA kernel

Nsight Graphics -
Debug/optimize graphics workloads



Advice

Save time! Avoid false positive optimizations. Intuition frequently leads astray.

Most tools focus on active work

Few consider the empty areas

The big picture = hot-spot analysis + cold-spot analysis

Look at “the big picture” and ask:

- What’s the effort vs reward of shrinking (a) vs filling (b)?
- How will the critical path be affected?

THANK YOU!

Download	**https://developer.nvidia.com/nsight-systems** **NOTE: Website versions newer than CUDA Toolkit**
Forums	**https://devtalk.nvidia.com**
Email	**nsight-systems@nvidia.com**


NVIDIA
Backup

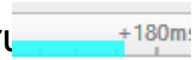
Tips & Tricks

Profile with sh or tmux and show-output for extra flexibility/interactivity

Tooltips!!! - if you want to know what the data is, then hover

Correlation

Select NVTX and CUDA ranges on CPU or GPU ... bidirectional

See locators in time-ru 

row hea 

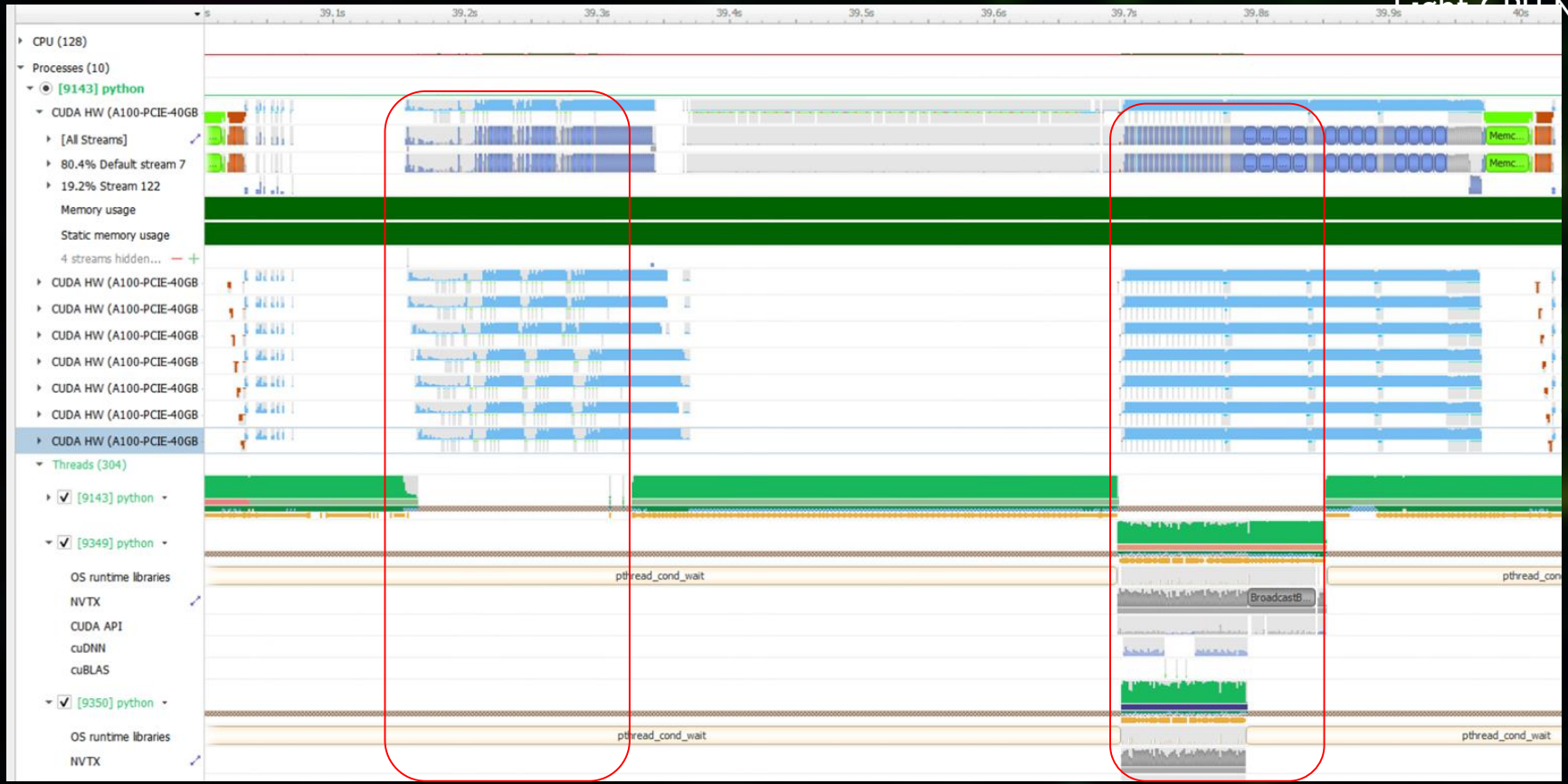
+arrow if sub-row

Keyboard Shortcuts

Zoom: +, -, CTRL+wheel, drag+context menu, backspace, trackpad pinch

Scroll sideways: left or right arrows, trackpad swipe

Pin rows: CTRL+P



Highly parallel sections
not all thread on screen

Overhead?

Yes, but can't quantify with one number, as it depends on features and app behavior

We all know that even the OS systems such as perf, ftrace, ETW are not free

Each feature enabled == more data!

- So nsys allows you to control it, and you could multi-pass

Sampling overhead is typically consistent

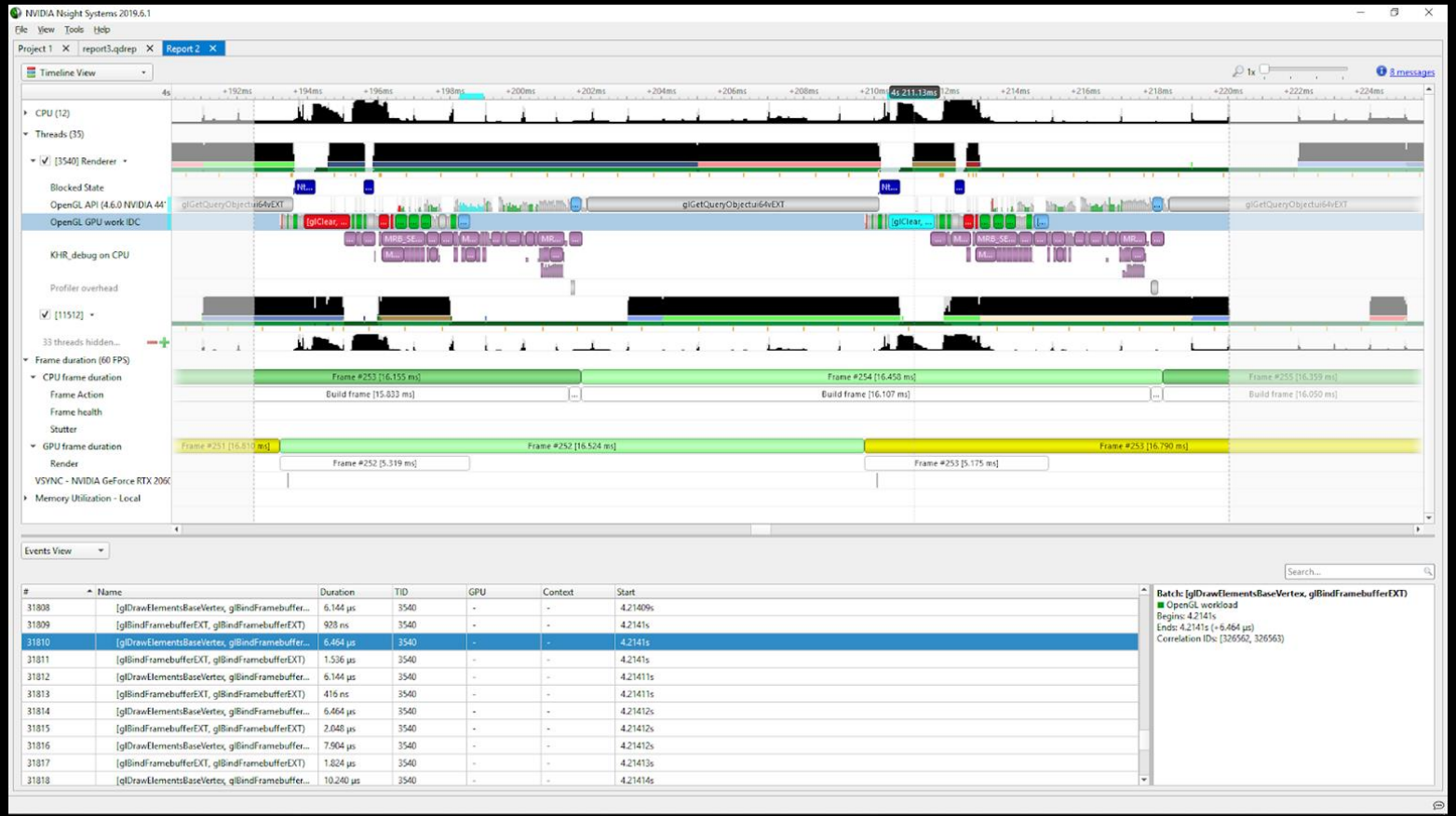
Trace overhead varies by how many events are generated, but is parallelized

- Ex: NVTX (in our next release v2020.2 + always improving)
  - ~180-190ns when recording
  - ~25-30ns when launched but not recording range

Misc overheads for some features

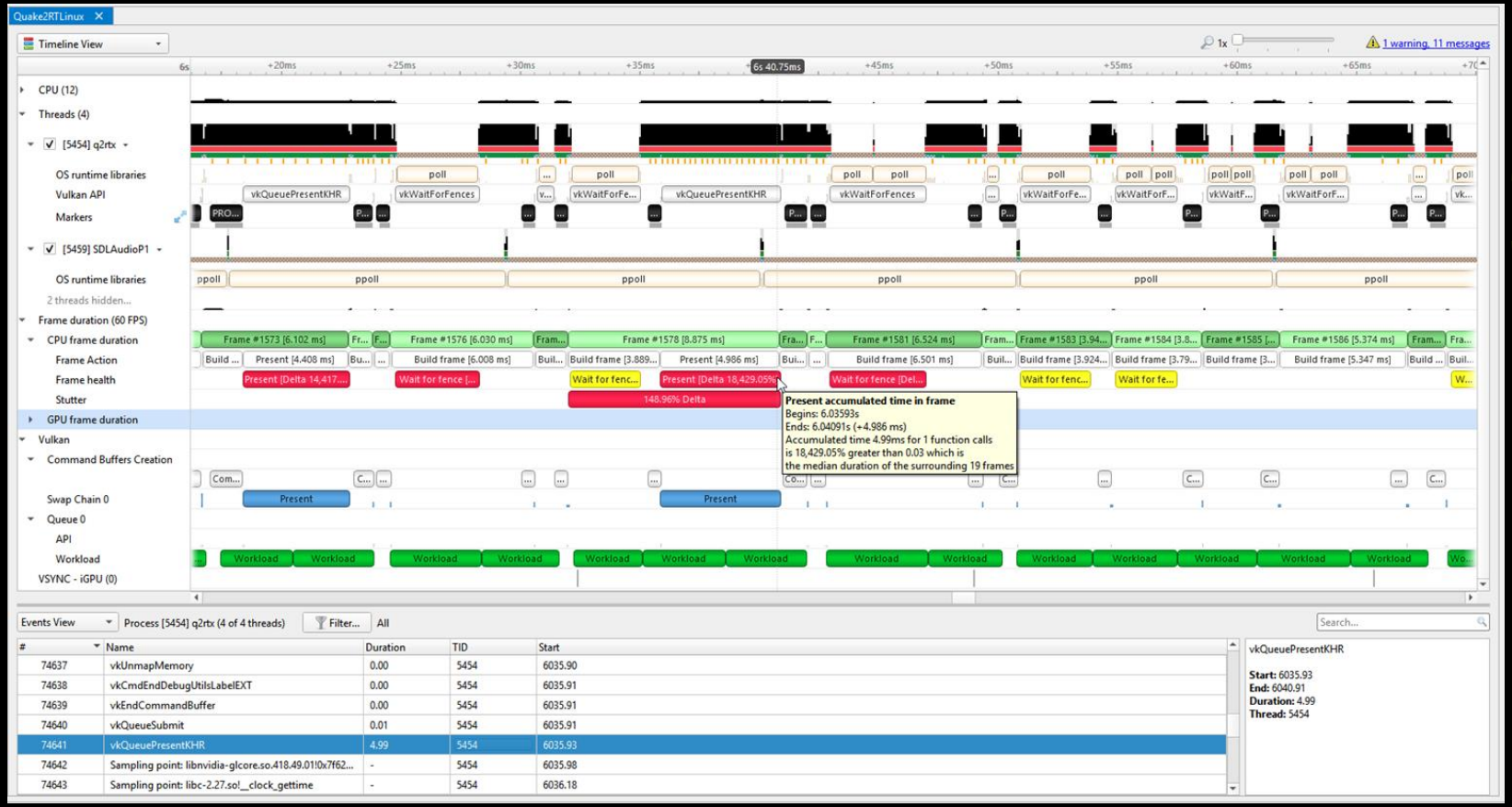
- Startup
- Recording start or end
- Periodic flush


Graphics Applications

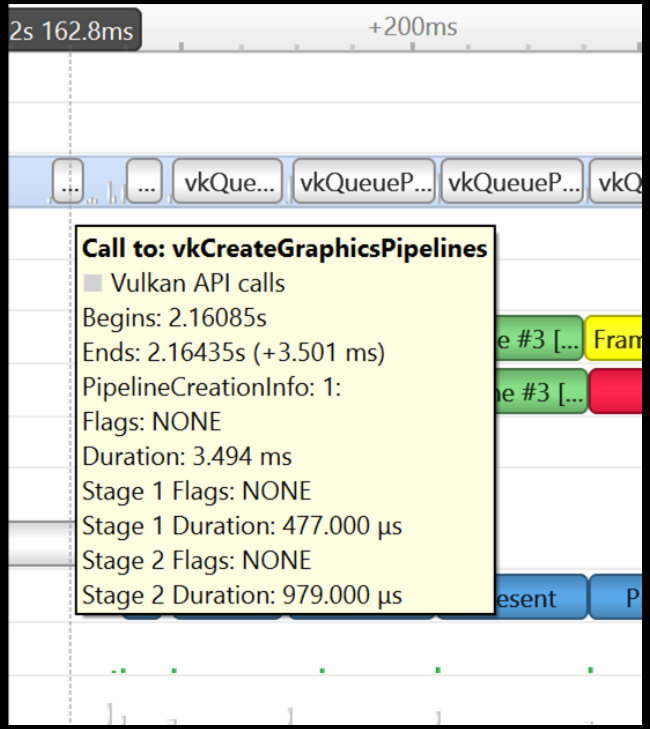


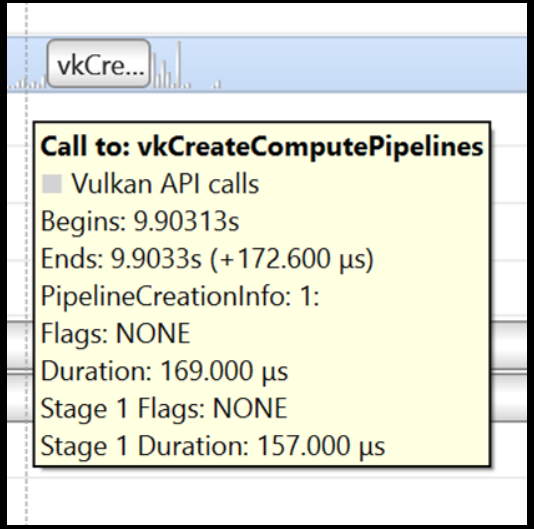
#	Name	Duration	TID	GPU	Context	Start
31808	[glDrawElementsBaseVertex, glBindFramebuffer...	6.144 µs	3540	-	-	4.21409s
31809	[glBindFramebufferEXT, glBindFramebufferEXT)	928 ns	3540	-	-	4.2141s
31810	[glDrawElementsBaseVertex, glBindFramebuffer...	6.464 µs	3540	-	-	4.2141s
31811	[glBindFramebufferEXT, glBindFramebufferEXT)	1.536 µs	3540	-	-	4.2141s
31812	[glDrawElementsBaseVertex, glBindFramebuffer...	6.144 µs	3540	-	-	4.21411s
31813	[glBindFramebufferEXT, glBindFramebufferEXT)	416 ns	3540	-	-	4.21411s
31814	[glDrawElementsBaseVertex, glBindFramebuffer...	6.464 µs	3540	-	-	4.21412s
31815	[glBindFramebufferEXT, glBindFramebufferEXT)	2.048 µs	3540	-	-	4.21412s
31816	[glDrawElementsBaseVertex, glBindFramebuffer...	7.904 µs	3540	-	-	4.21412s
31817	[glBindFramebufferEXT, glBindFramebufferEXT)	1.824 µs	3540	-	-	4.21413s
31818	[glDrawElementsBaseVertex, glBindFramebuffer...	10.240 µs	3540	-	-	4.21414s

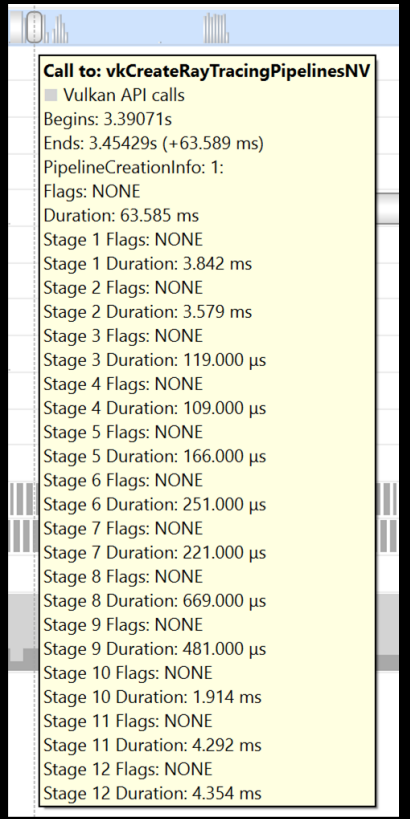
OpenGL
API + GPU workloads trace



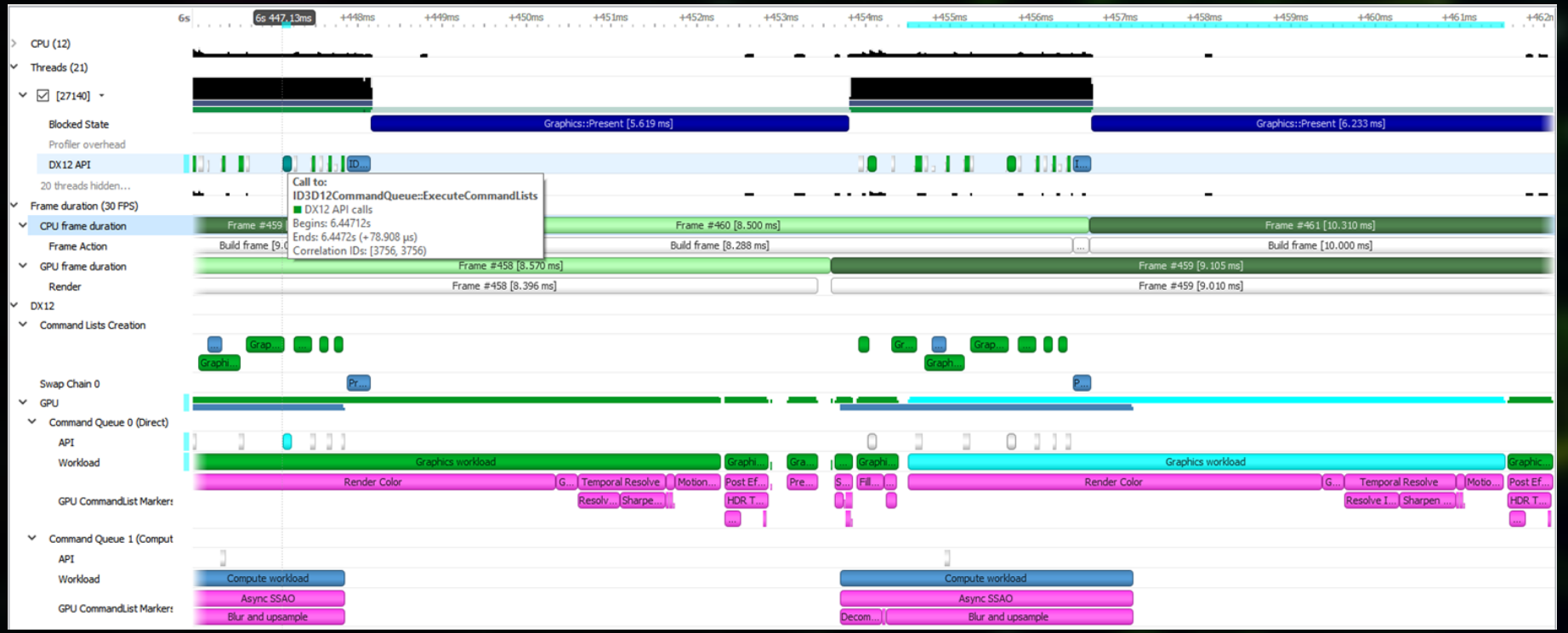
Vulkan
API + GPU command buffer workload trace
Example: Quake2RTX on Linux



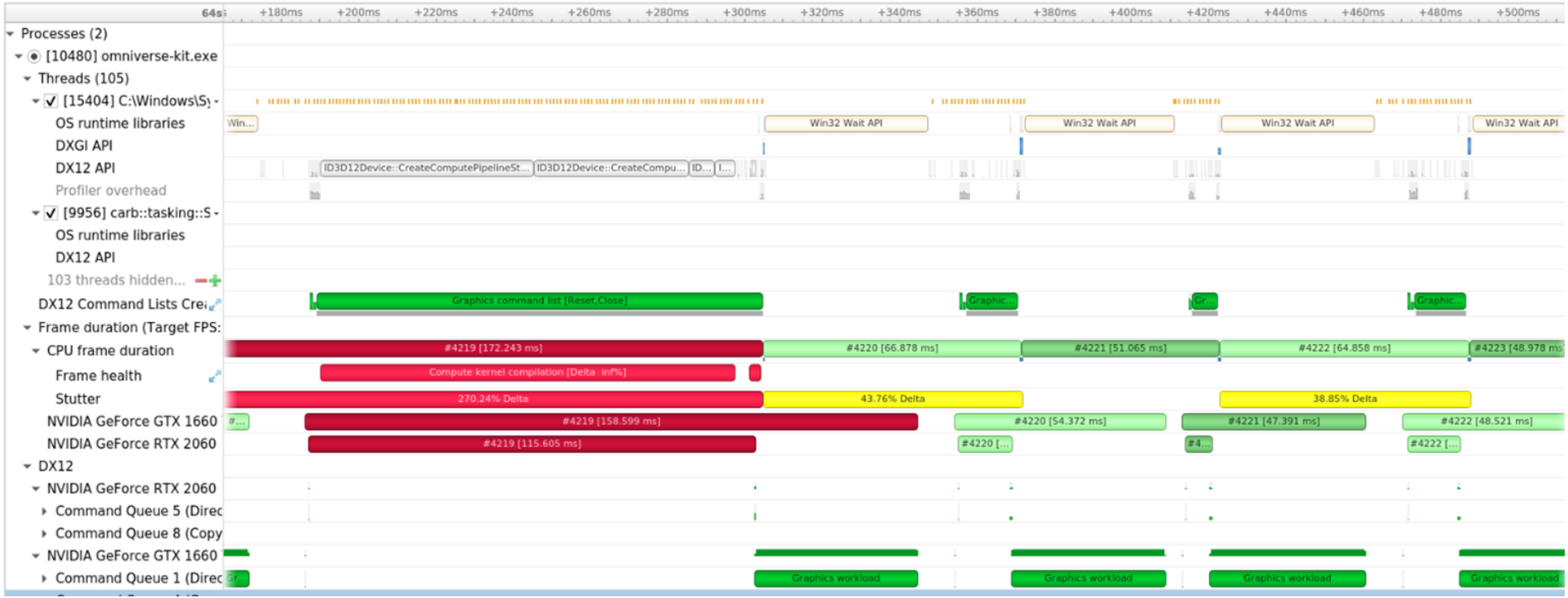




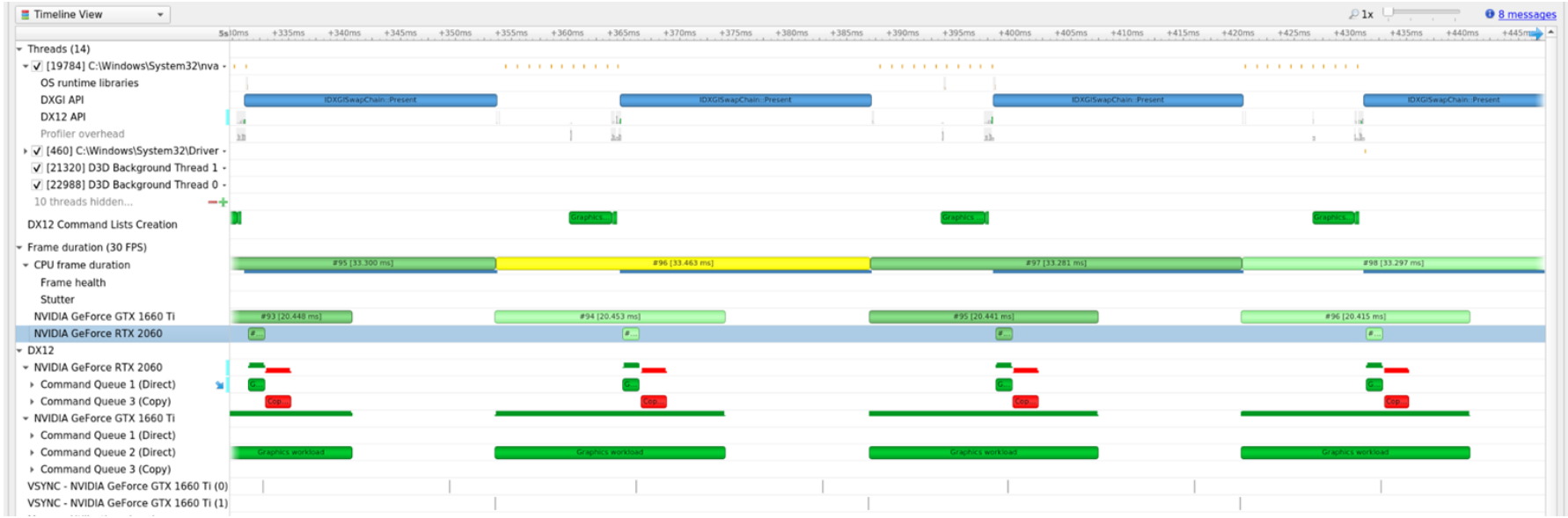
Vulkan Pipeline Creation Details



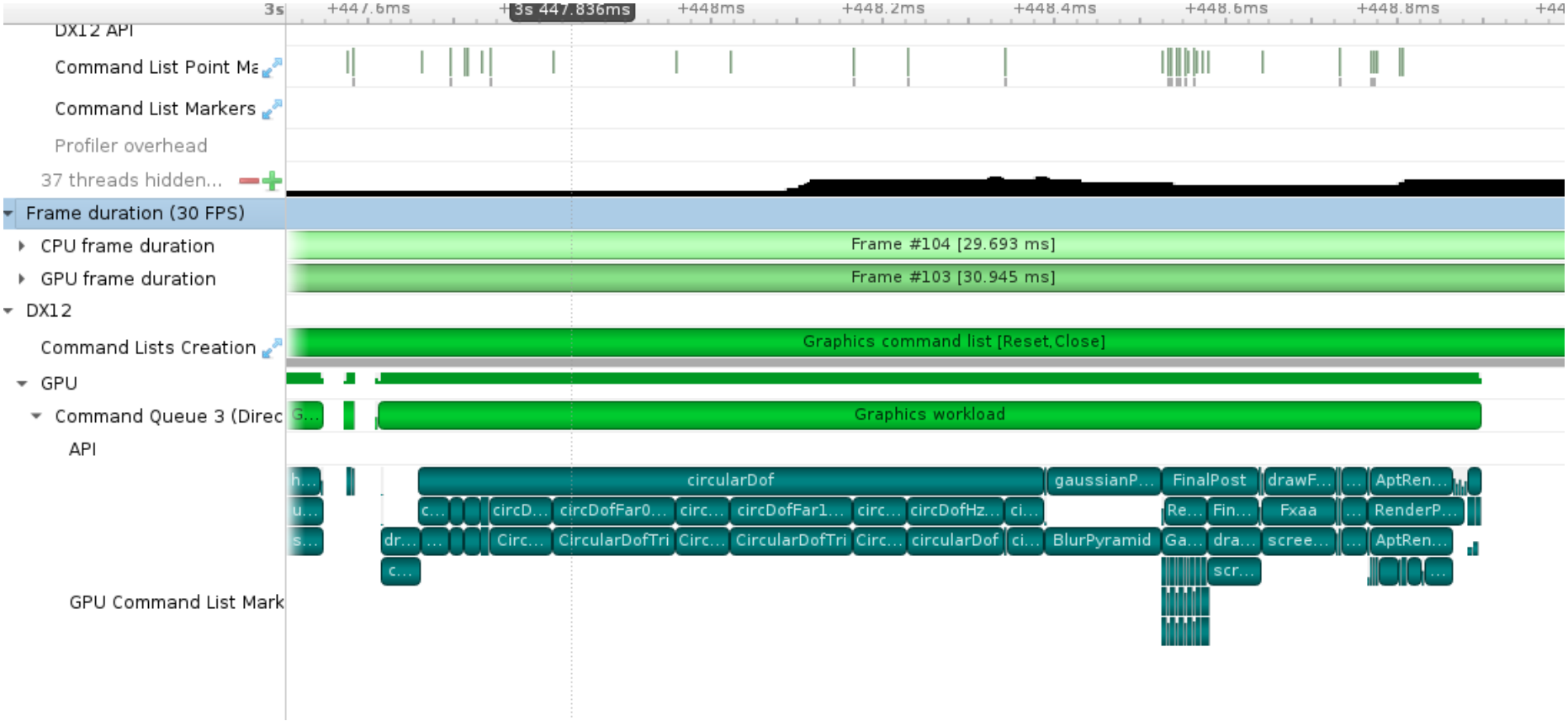
Direct3D 12 & DXR
API + GPU command list workload trace



D3D12 mGPU support

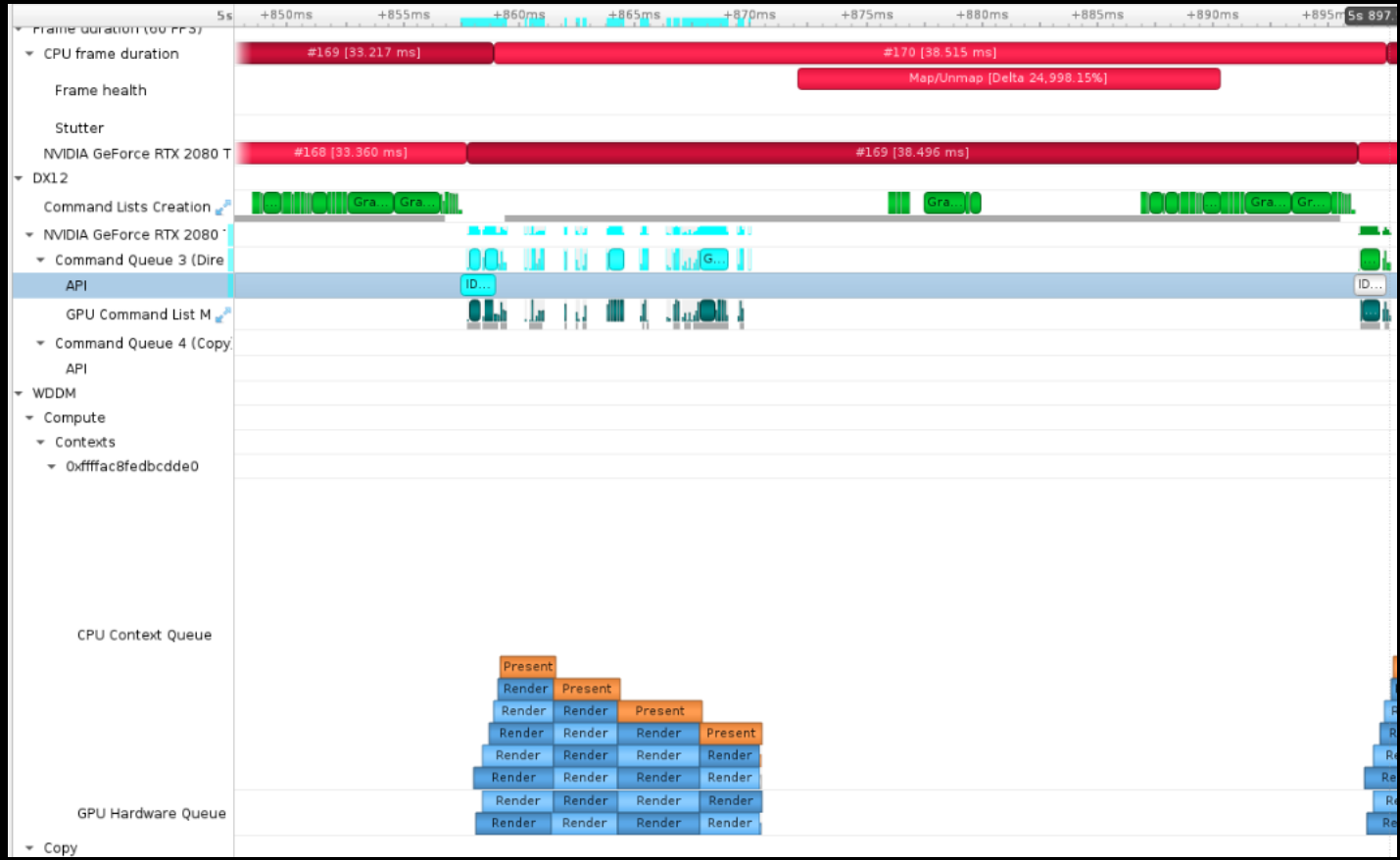


D3D12 mGPU support

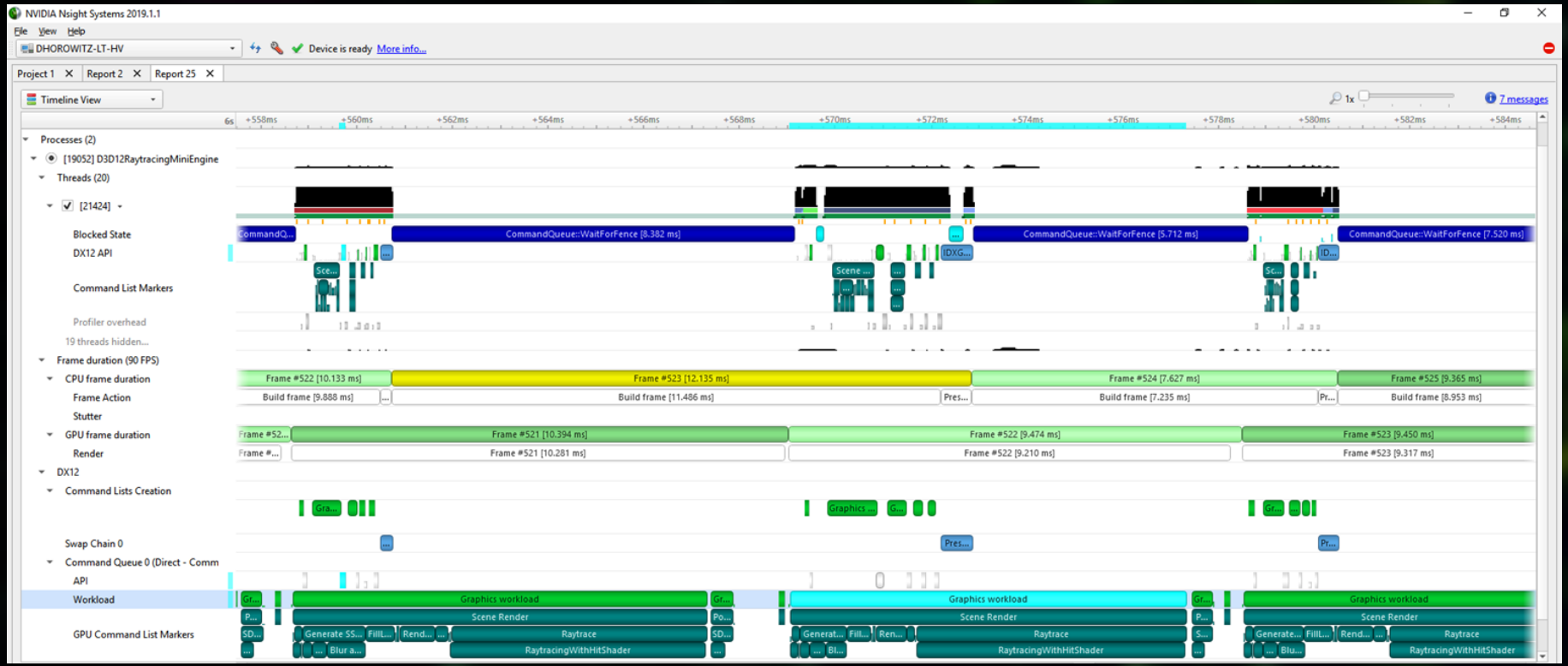


PIX marker times on the GPU

Direct3D 11 API + WDDM queues trace



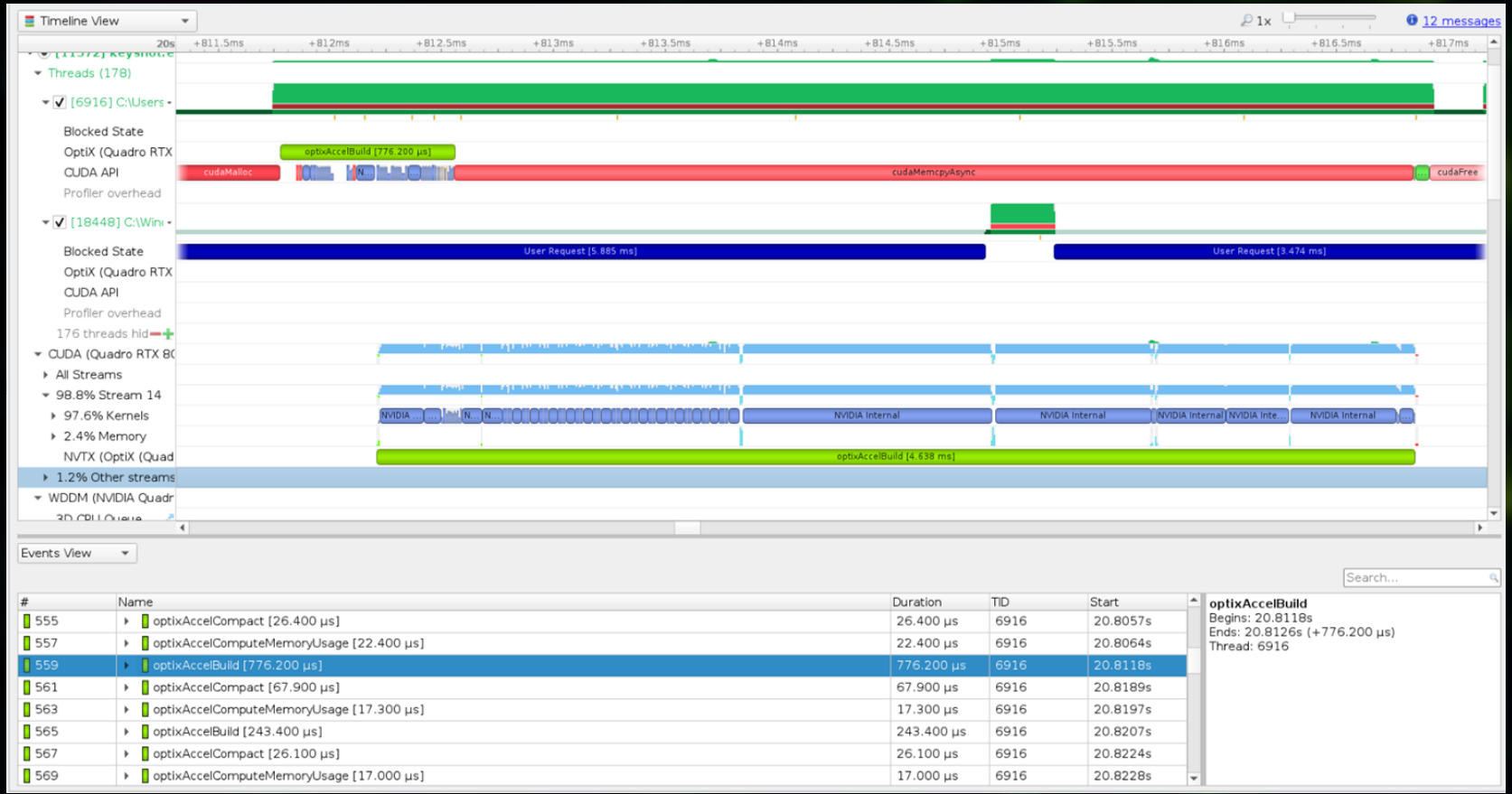
A deeper understanding of what's in your WDDM ranges



Frame timing for graphics APIs

Stutter analysis for graphics APIs
Local stutter and frame health

D3D12 mGPU support (coming soon)


Timeline View
1x
12 messages
+811.5ms
+812ms
+812.5ms
+813ms
+813.5ms
+814ms
+814.5ms
+815ms
+815.5ms
+816ms
+816.5ms
+817ms
Threads (178)
[6916] C:\Users
Blocked State
OptiX (Quadro RTX
CUDA API
Profiler overhead
optixAccelBuild [776.200 µs]
cudaMalloc
cudaMemcpyAsync
cudaFree
[18448] C:\Win
User Request [5.885 ms]
User Request [3.474 ms]
176 threads hid
CUDA (Quadro RTX 8
All Streams
98.8% Stream 14
97.6% Kernels
2.4% Memory
NVTX (OptiX (Quad
NVIDIA Internal
optixAccelBuild [4.638 ms]
1.2% Other streams
WDDM (NVIDIA Quadr
Events View
Search...
Name Duration TID Start
555 optixAccelCompact [26.400 µs] 26.400 µs 6916 20.8057s
557 optixAccelComputeMemoryUsage [22.400 µs] 22.400 µs 6916 20.8064s
559 optixAccelBuild [776.200 µs] 776.200 µs 6916 20.8118s
561 optixAccelCompact [67.900 µs] 67.900 µs 6916 20.8189s
563 optixAccelComputeMemoryUsage [17.300 µs] 17.300 µs 6916 20.8197s
565 optixAccelBuild [243.400 µs] 243.400 µs 6916 20.8207s
567 optixAccelCompact [26.100 µs] 26.100 µs 6916 20.8224s
569 optixAccelComputeMemoryUsage [17.000 µs] 17.000 µs 6916 20.8228s
optixAccelBuild
Begins: 20.8118s
Ends: 20.8126s (+776.200 µs)
Thread: 6916

OptiX trace