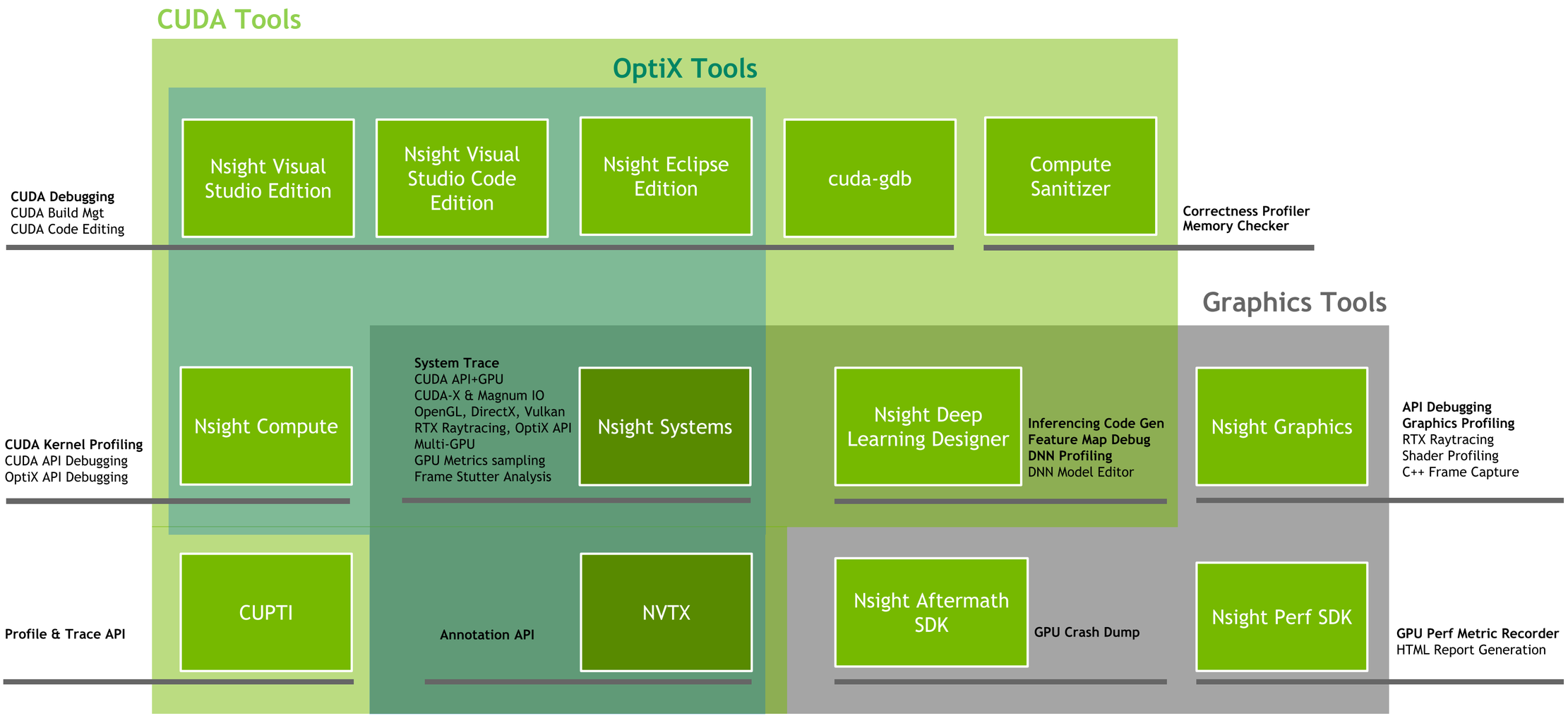
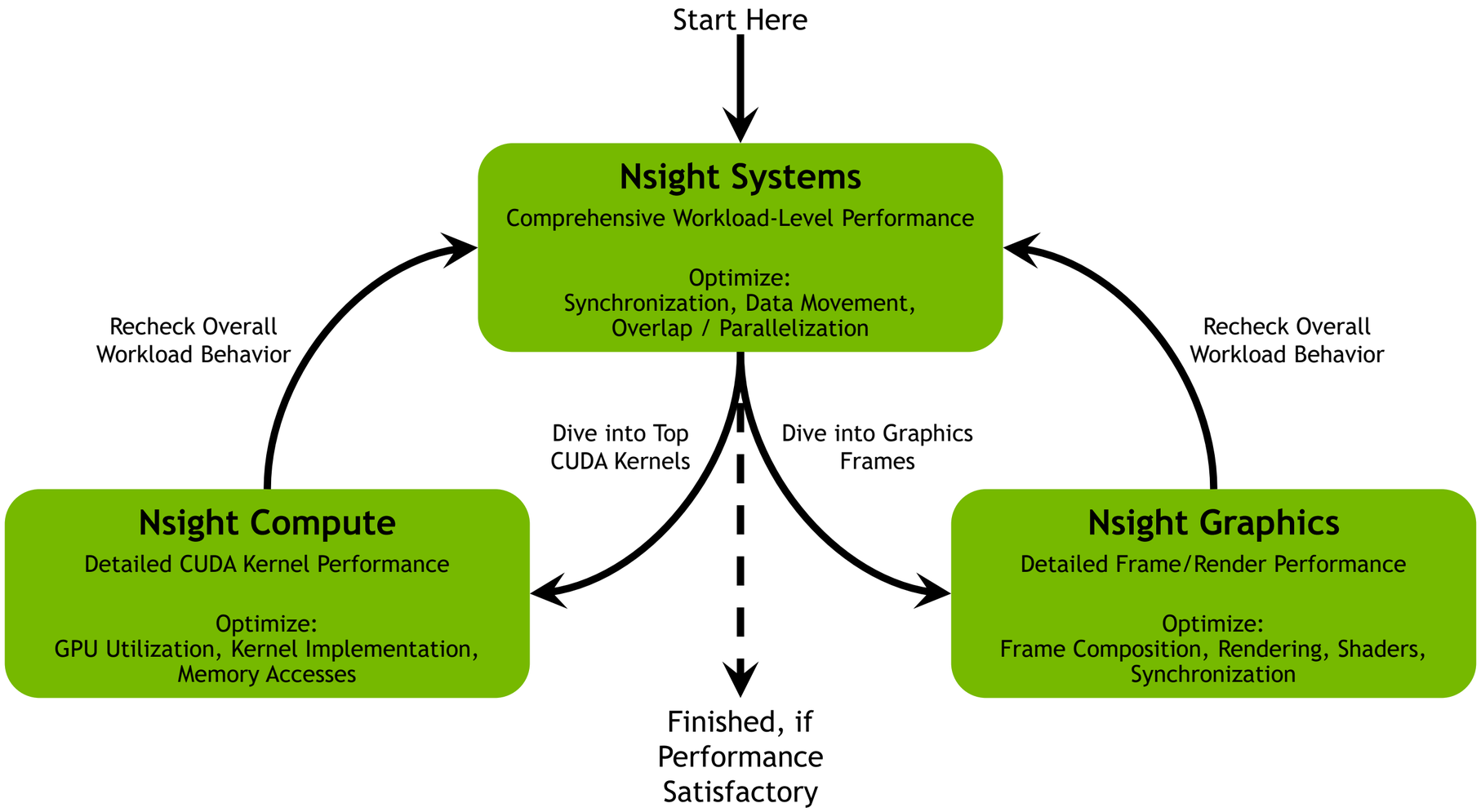
CUDA PROFILING TOOLS OVERVIEW

JACKSON MARUSARZ

NSIGHT DEVELOPER TOOLS



NSIGHT PROFILING TOOLS WORKFLOW



NSIGHT SYSTEMS

NSIGHT SYSTEMS

System Profiler

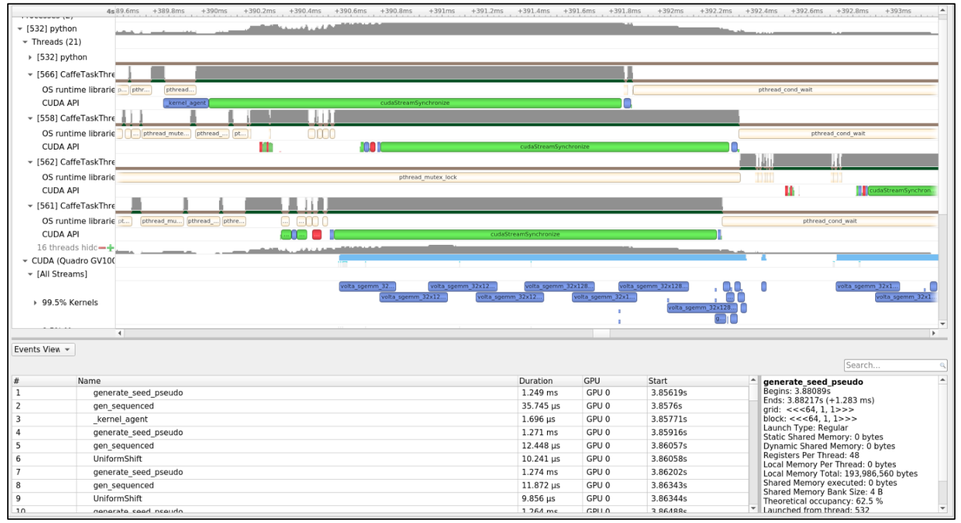
Key Features:

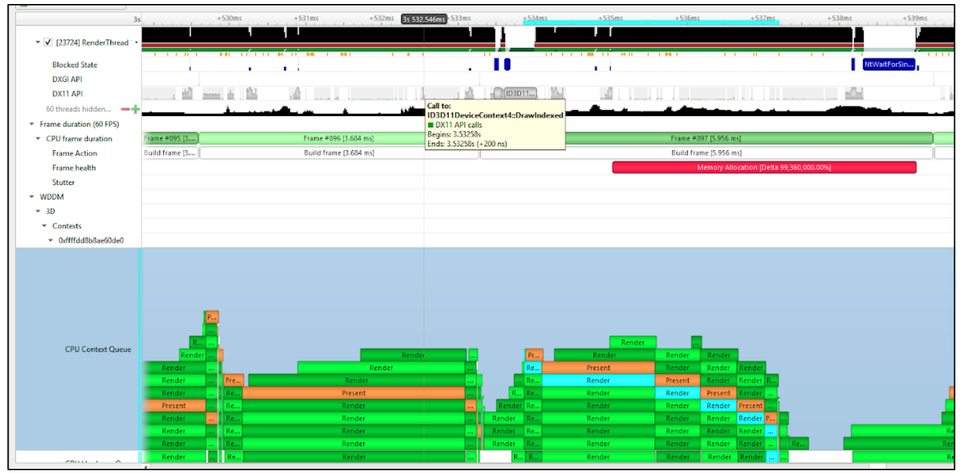
- System-wide application algorithm tuning
  - Multi-process tree support
- Locate optimization opportunities
  - Visualize millions of events on a very fast GUI timeline
  - Or gaps of unused CPU and GPU time
- Balance your workload across multiple CPUs and GPUs
  - CPU algorithms, utilization and thread state
    GPU streams, kernels, memory transfers, etc
- Command Line, Standalone, IDE Integration

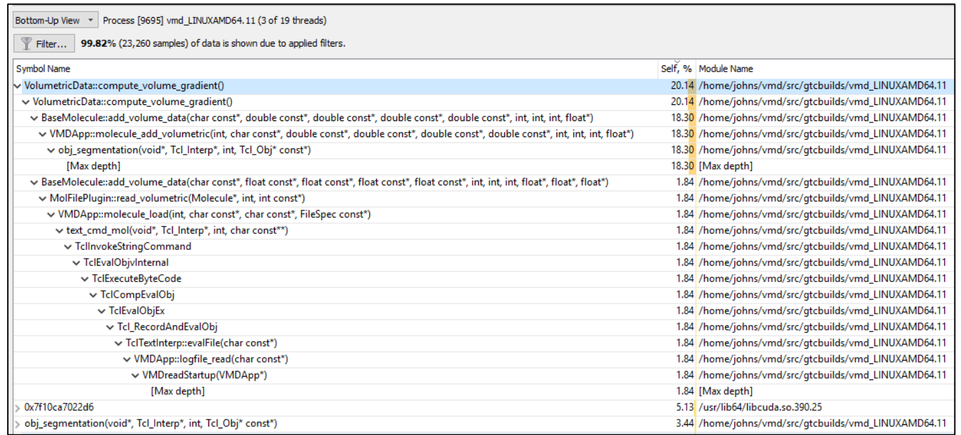
OS: Linux (x86, Power, Arm SBSA, Tegra), Windows, MacOSX (host)

GPUs: Pascal+

Docs/product: https://developer.nvidia.com/nsight-systems







Processes & threads

OpenGL trace

NVTX annotations

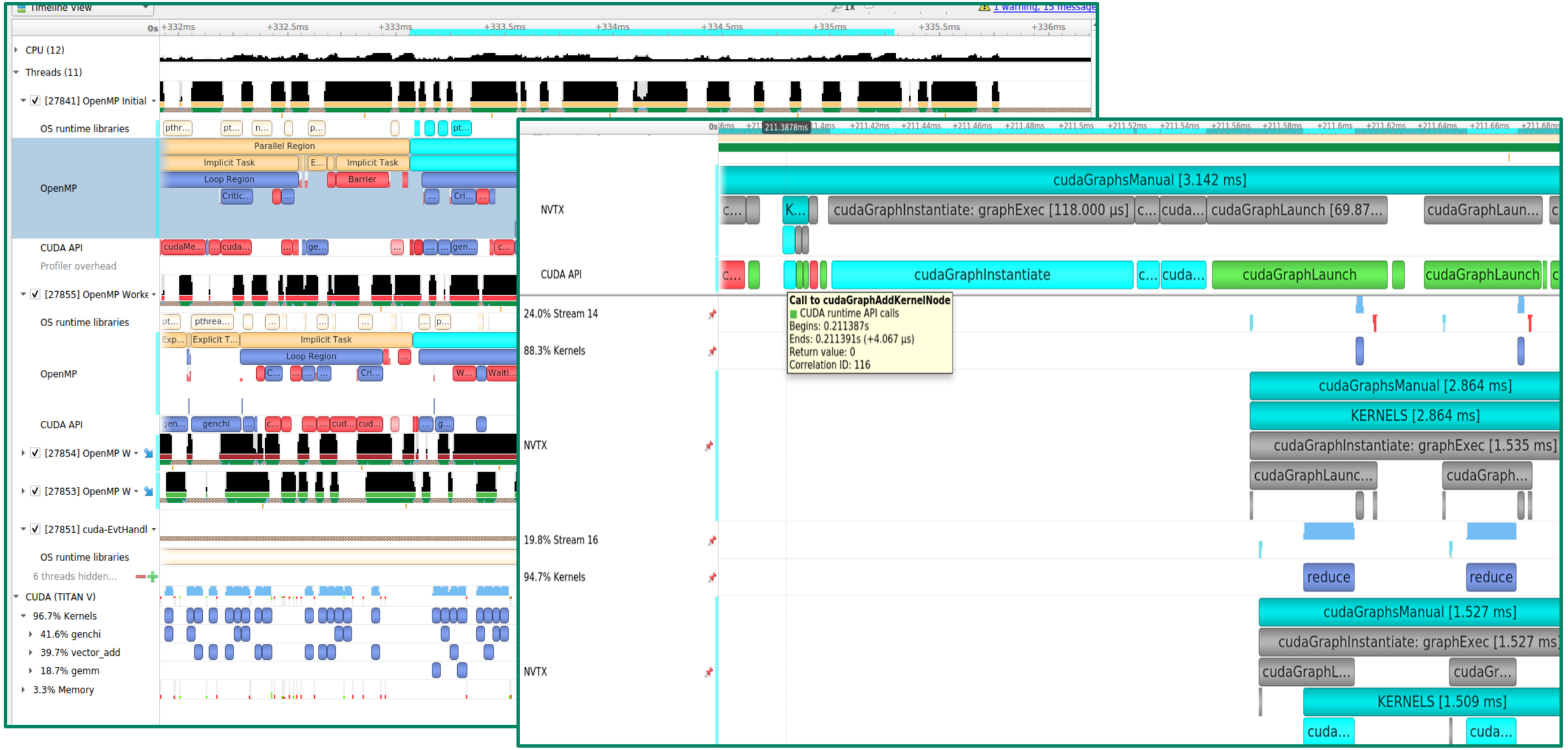
CUDA API trace

GPU CUDA Kernel & memory transfer activities

CPU call-stack samples

NVIDIA

ZOOM/FILTER TO EXACT AREAS OF INTEREST



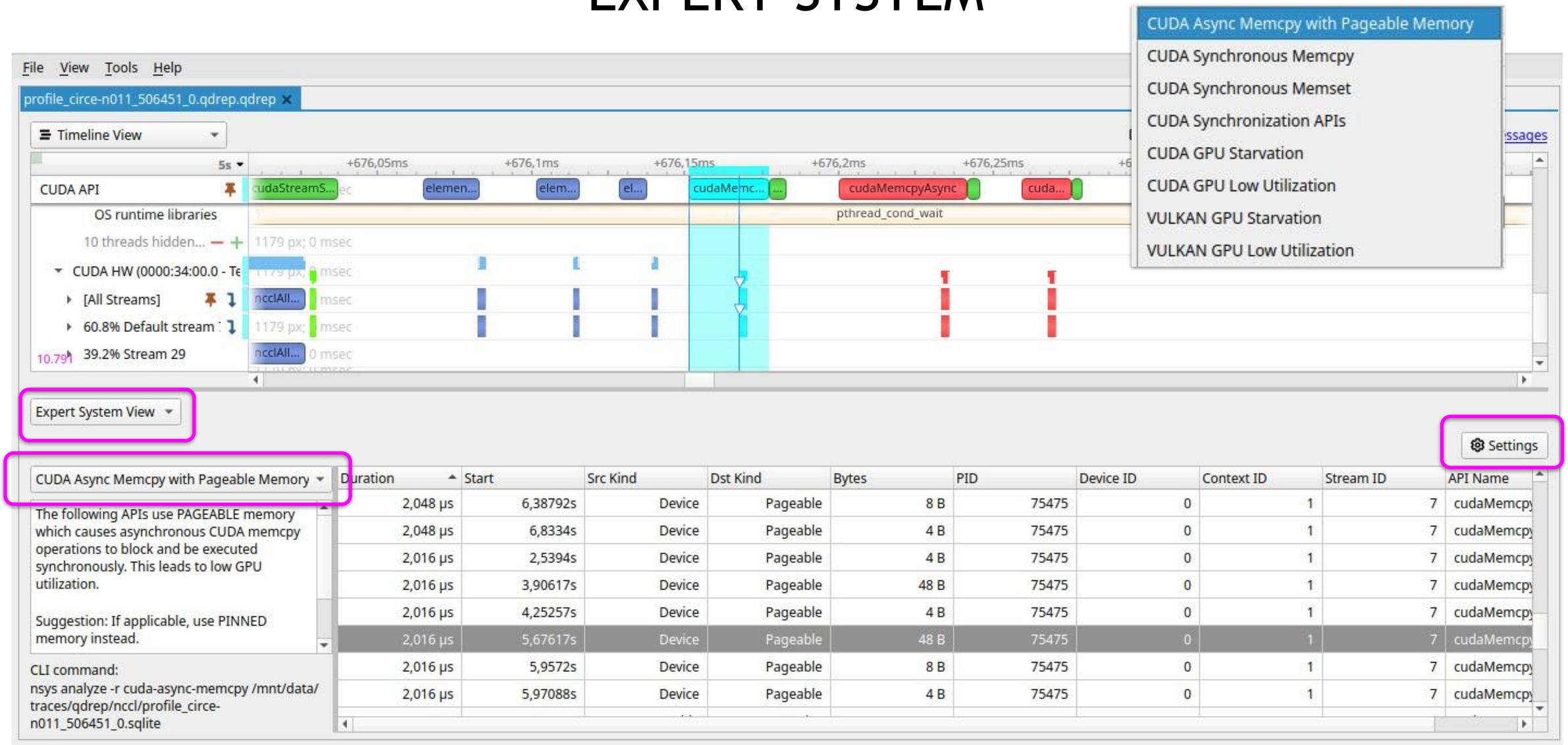
FUNCTION TABLE WITH CALL-STACK BACKTRACES

Bottom-Up View | Process [9695] vmd_LINUXAMD64.11 (3 of 19 threads)

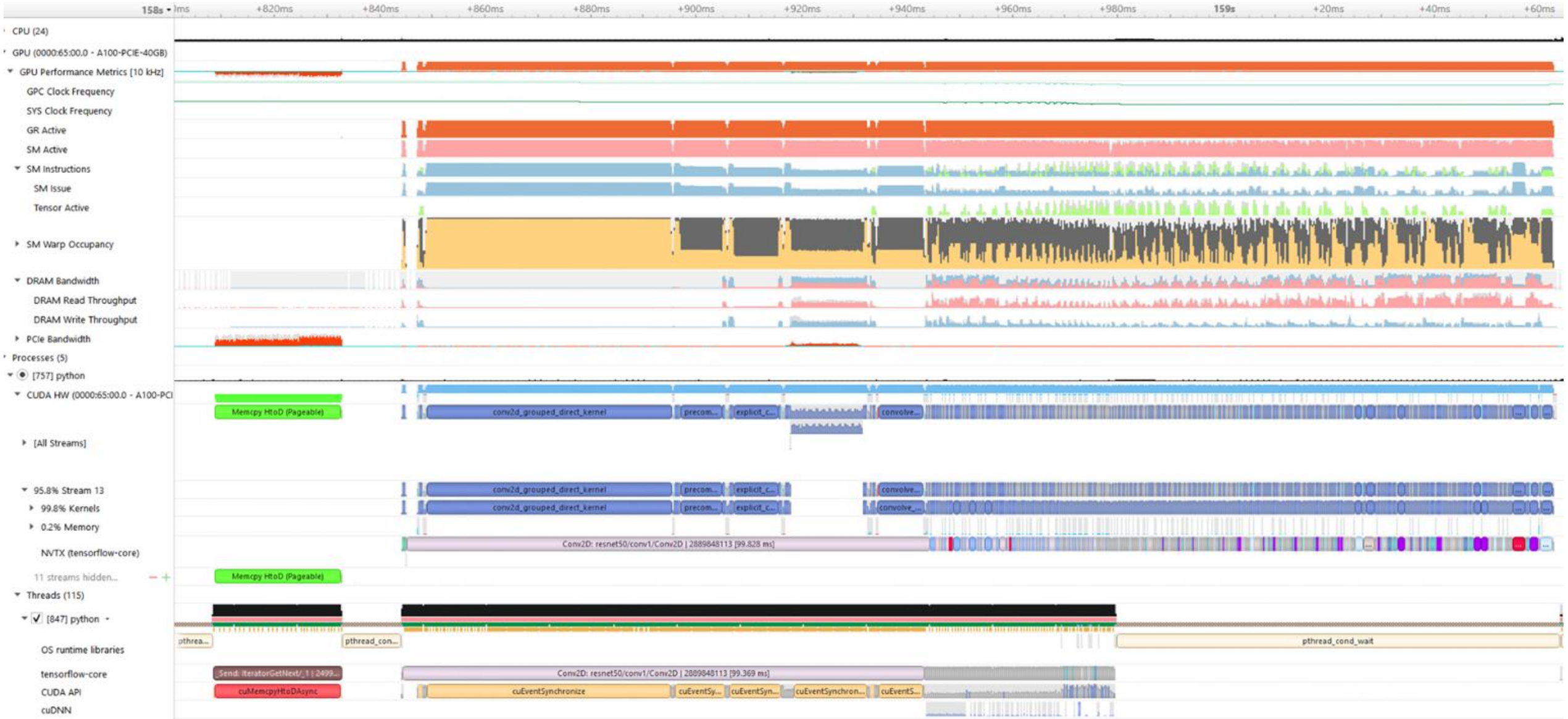
Filter... **99.82%** (23,260 samples) of data is shown due to applied filters.

Symbol Name	Self, %	Module Name
VolumetricData::compute_volume_gradient()	20.14	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
VolumetricData::compute_volume_gradient()	20.14	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
BaseMolecule::add_volume_data(char const*, double const*, double const*, double const*, double const*, int, int, int, float*)	18.30	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
VMDApp::molecule_add_volumetric(int, char const*, double const*, double const*, double const*, double const*, int, int, int, float*)	18.30	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
obj_segmentation(void*, Tcl_Interp*, int, Tcl_Obj* const*)	18.30	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
[Max depth]	18.30	[Max depth]
BaseMolecule::add_volume_data(char const*, float const*, float const*, float const*, float const*, int, int, int, float*, float*, float*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
MolFilePlugin::read_volumetric(Molecule*, int, int const*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
VMDApp::molecule_load(int, char const*, char const*, FileSpec const*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
text_cmd_mol(void*, Tcl_Interp*, int, char const**)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
TclInvokeStringCommand	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
TclEvalObjvInternal	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
TclExecuteByteCode	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
TclCompEvalObj	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
TclEvalObjEx	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
Tcl_RecordAndEvalObj	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
TclTextInterp::evalFile(char const*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
VMDApp::logfile_read(char const*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
VMDreadStartup(VMDApp*)	1.84	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11
[Max depth]	1.84	[Max depth]
0x7f10ca7022d6	5.13	/usr/lib64/libcuda.so.390.25
obj_segmentation(void*, Tcl_Interp*, int, Tcl_Obj* const*)	3.44	/home/johns/vmd/src/gtcbuilds/vmd_LINUXAMD64.11

EXPERT SYSTEM



GPU METRICS SAMPLING

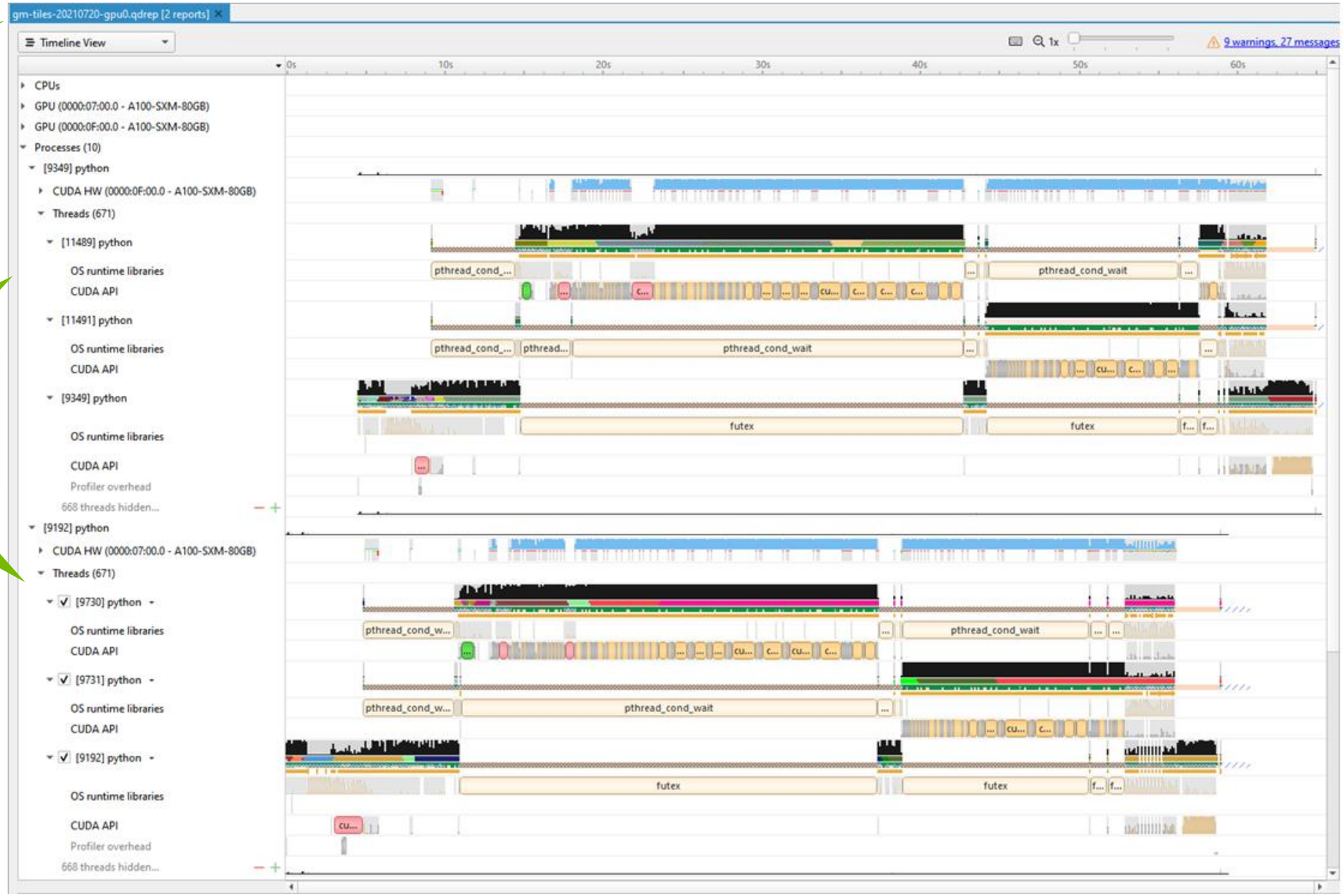


MULTI-REPORT TILING

Visualize more parallel activity

Open multiple reports

Correlated on same timeline



NVIDIA


NSIGHT COMPUTE

NSIGHT COMPUTE

Kernel Profiling Tool

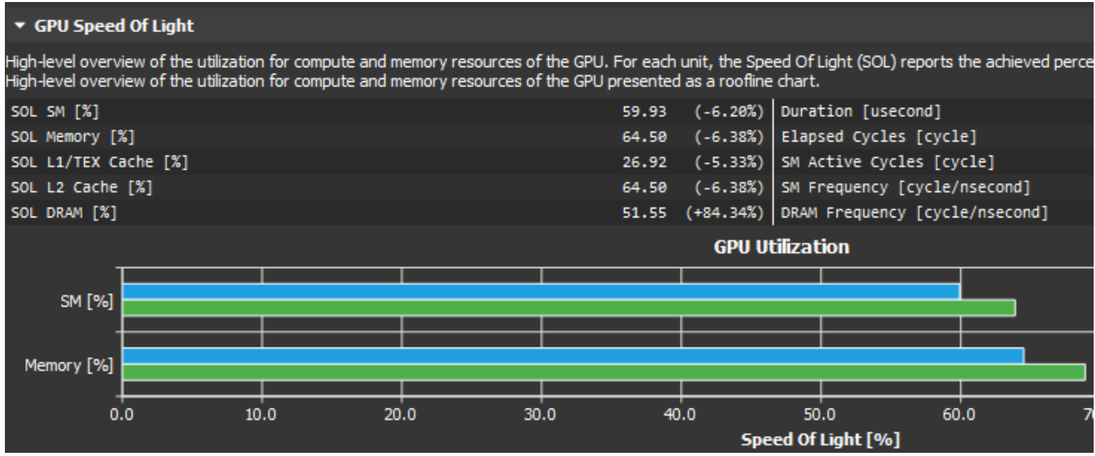
Key Features:

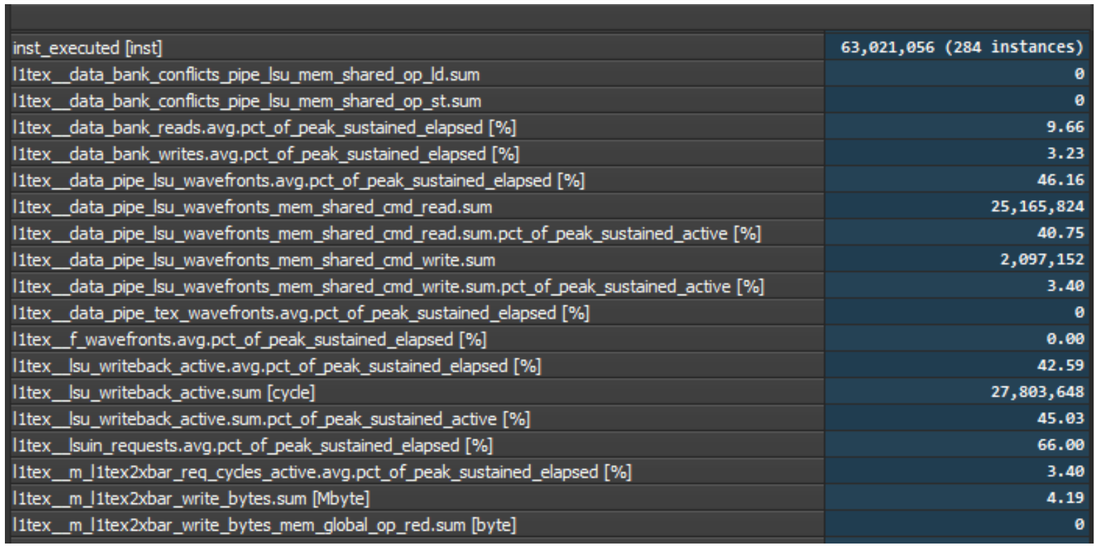
- Interactive CUDA API debugging and kernel profiling
- Built-in rules expertise
- Fully customizable data collection and display
- Command Line, Standalone, IDE Integration, Remote Targets

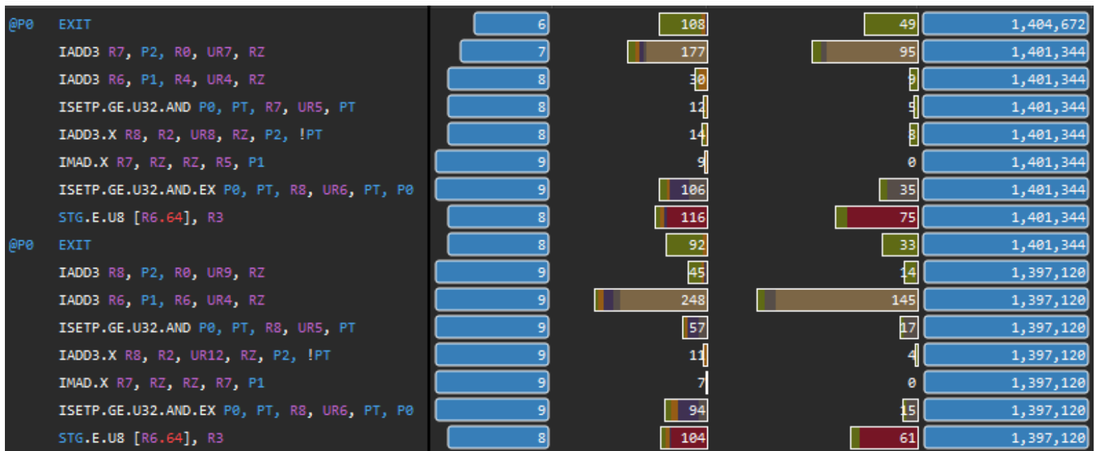
OS: Linux (x86, Power, Tegra, Arm SBSA), Windows, MacOSX (host only)

GPUs: Volta+

Docs/product: https://developer.nvidia.com/nsight-compute



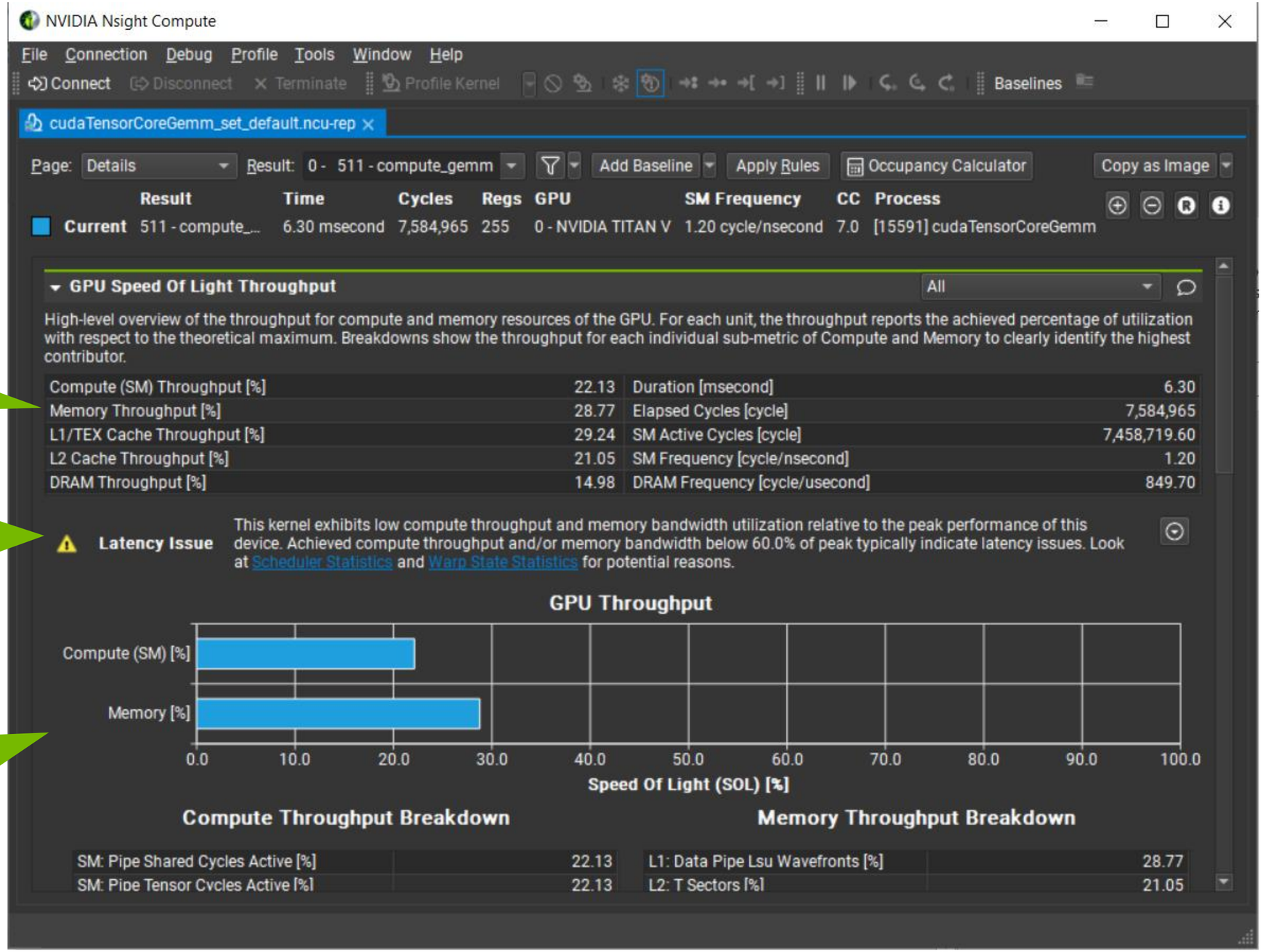




Targeted metric sections

Built-in expertise for Guided Analysis and optimization

Customizable data collection and presentation



Visual memory analysis chart

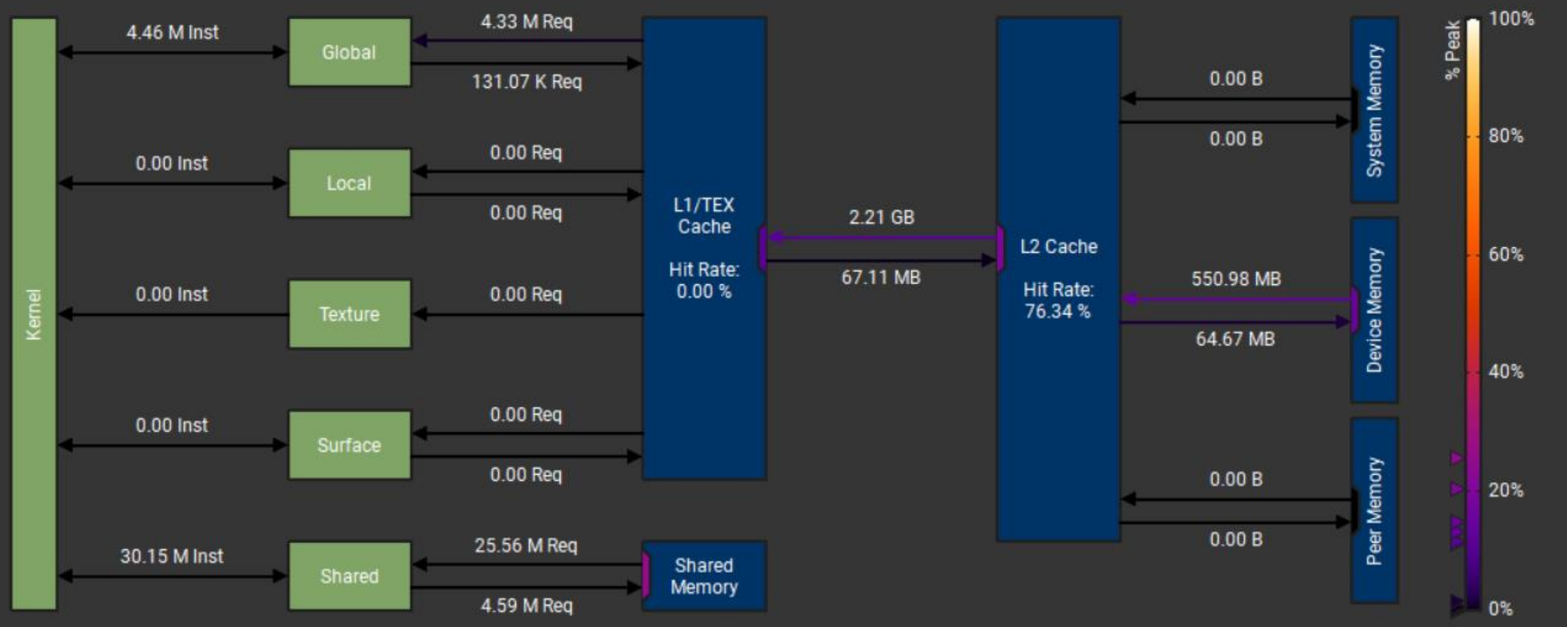
Metrics for peak performance ratios

Memory Workload Analysis — All

Detailed analysis of the memory resources of the GPU. Memory can become a limiting factor for the overall kernel performance when fully utilizing the involved hardware units (Mem Busy), exhausting the available communication bandwidth between those units (Max Bandwidth), or by reaching the maximum throughput of issuing memory instructions (Mem Pipes Busy). Detailed chart of the memory units. Detailed tables with data for each memory unit.

Memory Throughput [Gbyte/second]	97.71	Mem Busy [%]	28.78
L1/TEX Hit Rate [%]	0	Max Bandwidth [%]	20.45
L2 Hit Rate [%]	76.34	Mem Pipes Busy [%]	9.82

Memory Chart — Show As: Transfer Size

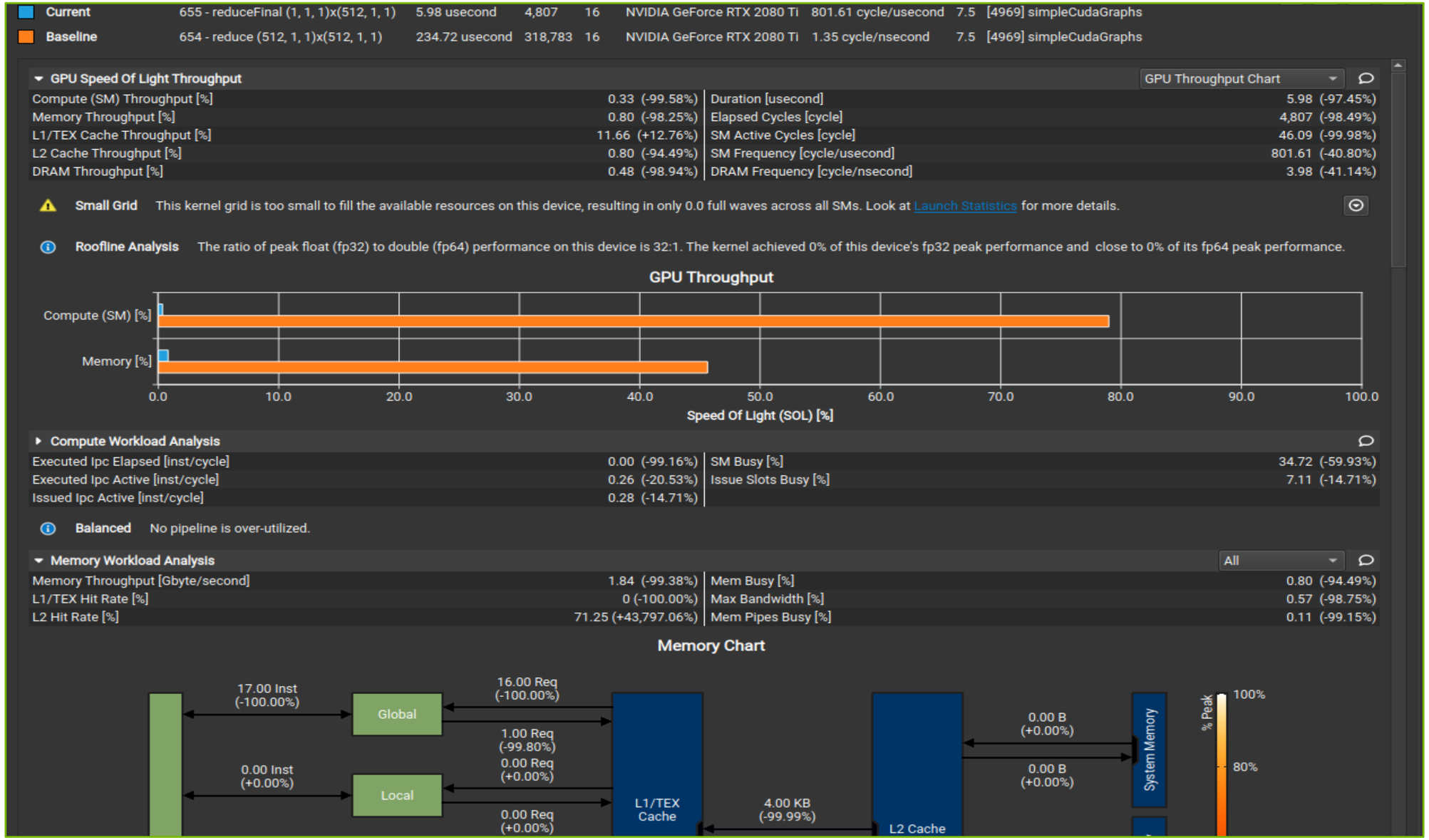


Shared Memory

	Instructions	Requests	Wavefronts	% Peak	Bank Conflicts
Shared Load	25,559,040	25,559,040	136,852,276	22.57	35,140,404
Shared Store	4,587,520	4,587,520	19,398,656	3.20	1,572,864
Shared Atomic	0	0	0	0	0
Other	-	-	147,957	0.02	506
Total	30,146,560	30,146,560	156,398,889	25.80	36,713,774

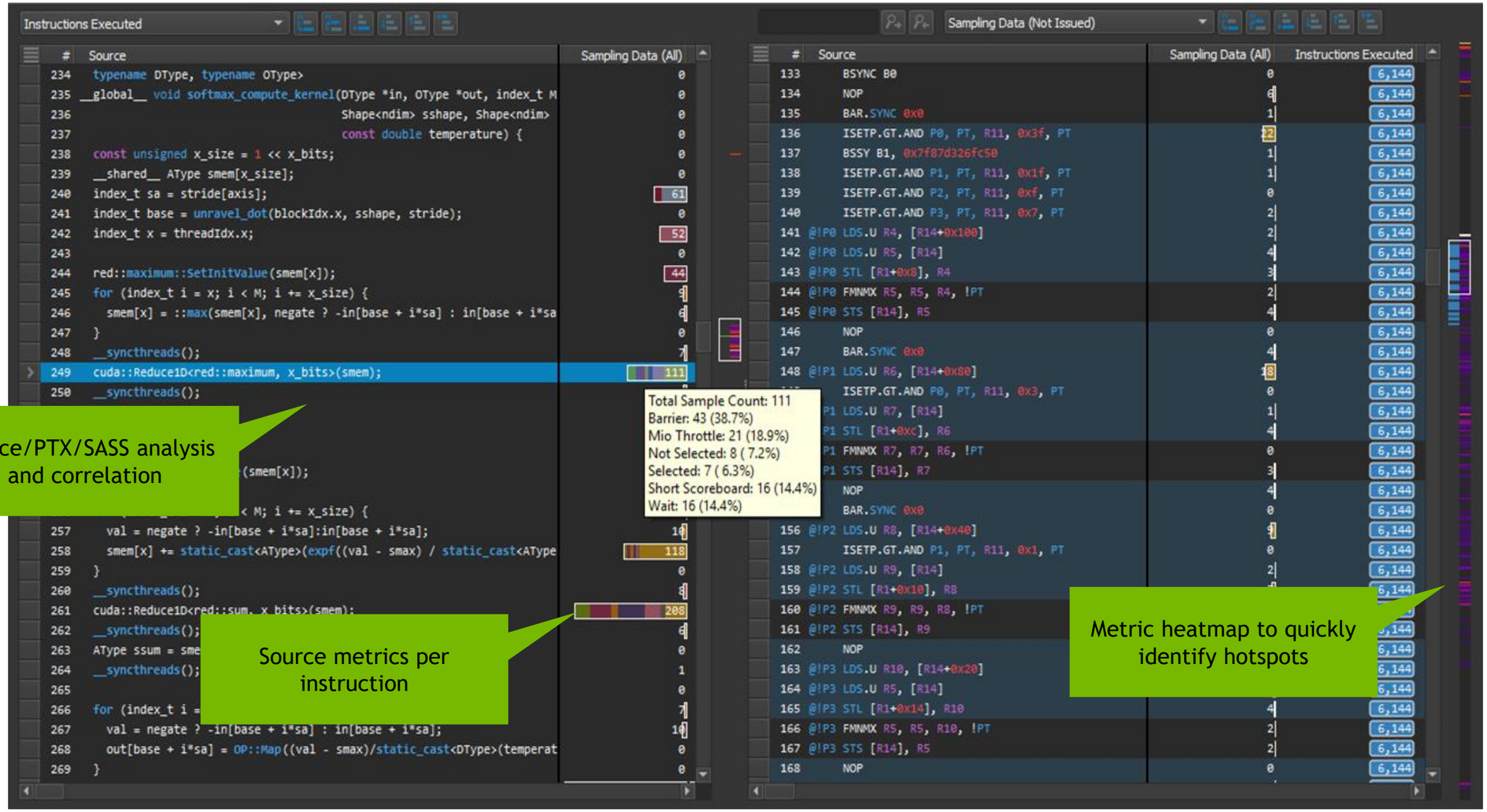
BASELINES

- Comparison of results directly within the tool with "Baselines"
- Supported across kernels, reports, and GPU architectures



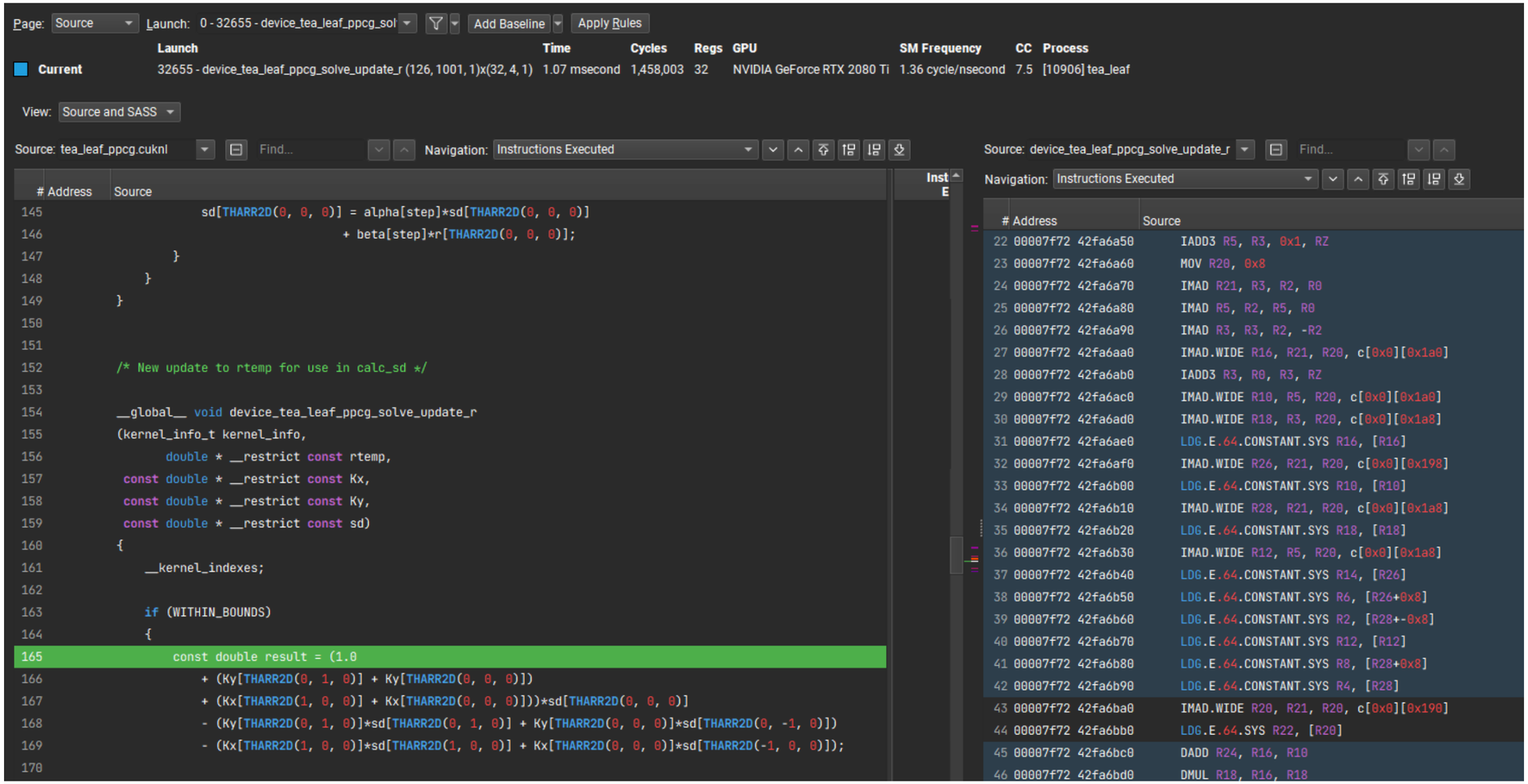
SOURCE VIEW

- Source/PTX/SASS analysis and correlation
- Source metrics per instruction and aggregated (e.g. PC sampling data)
- Metric heatmap



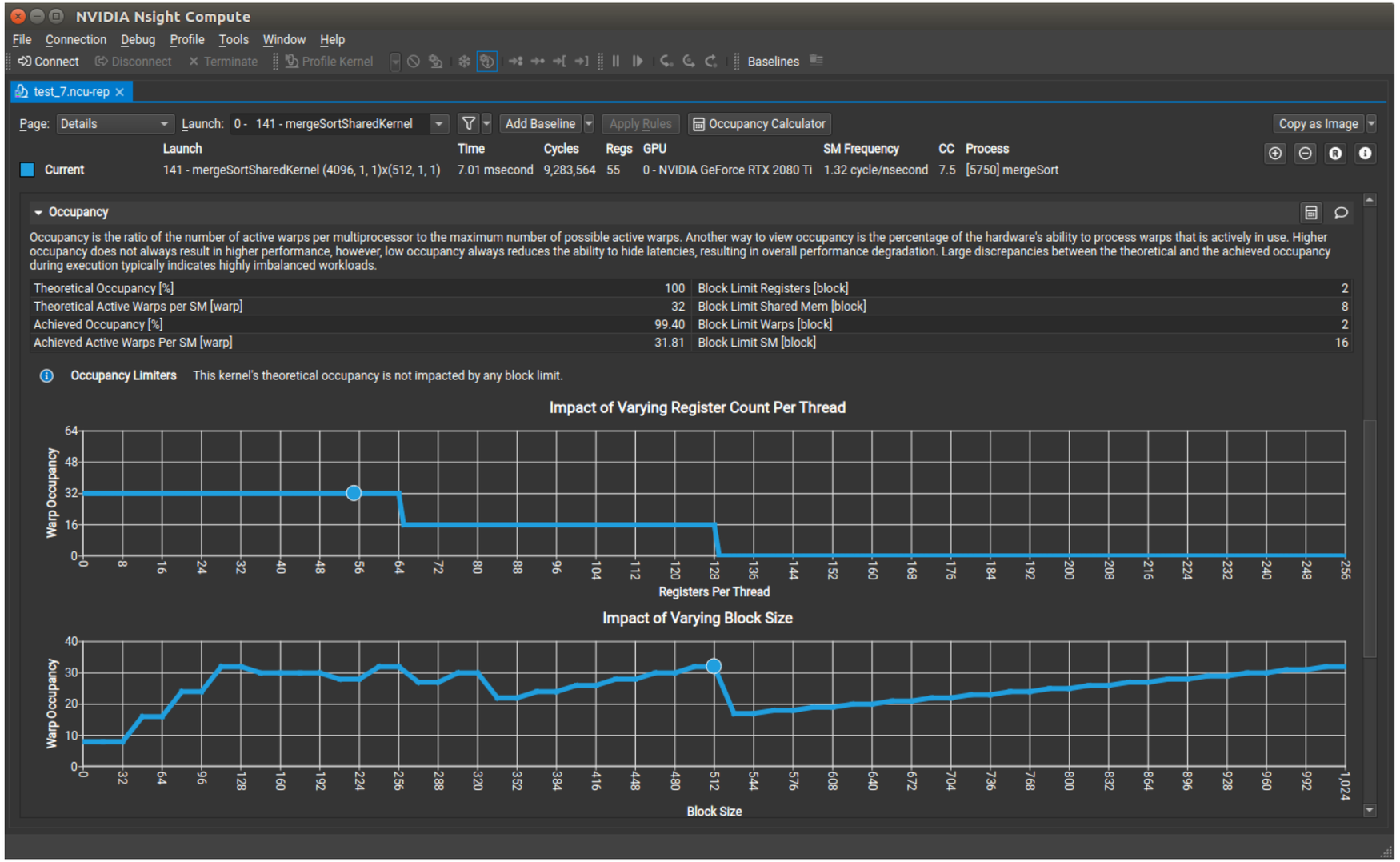
STANDALONE SOURCE VIEWER

- View of side-by-side assembly and correlated source code for CUDA kernels
- No profile required
- Open .cubin files directly
- Helps identify compiler optimizations and inefficiencies



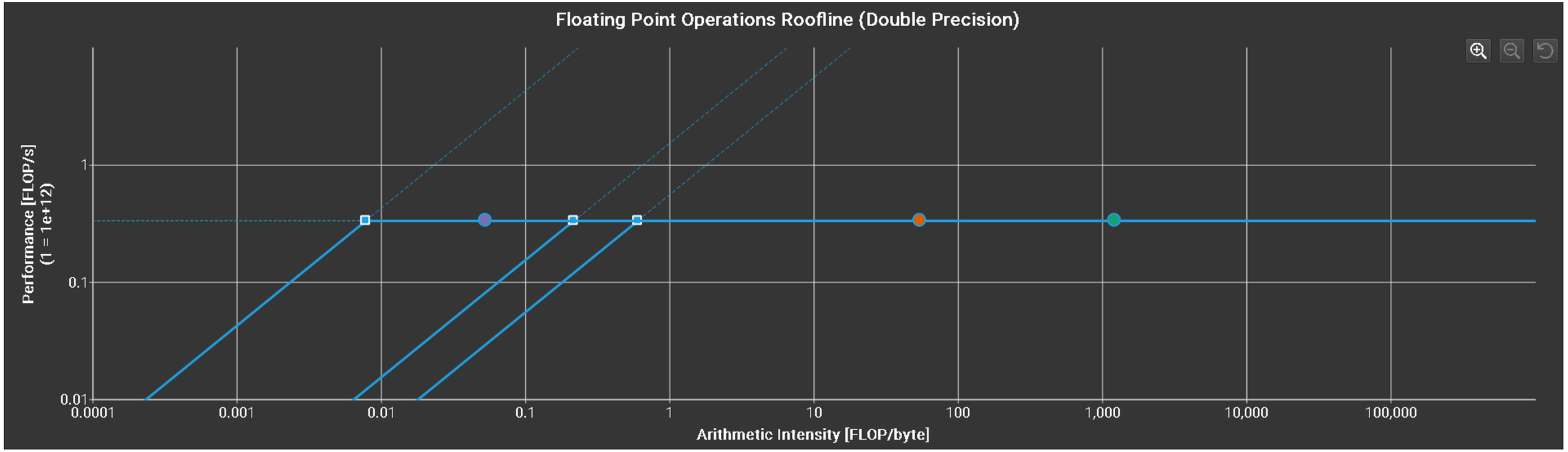
OCCUPANCY CALCULATOR

Model Hardware Usage and Identify Limiters



- Model theoretical hardware usage
- Understand limitations from hardware vs. kernel parameters
- Configure model to vary HW and kernel parameters
- Opened from an existing report or as a new activity

HIERARCHICAL ROOFLINE



- Visualize multiple levels of the memory hierarchy
- Identify bottlenecks caused by memory limitations
- Determine how modifying algorithms may (or may not) impact performance

Sections/Rules Info

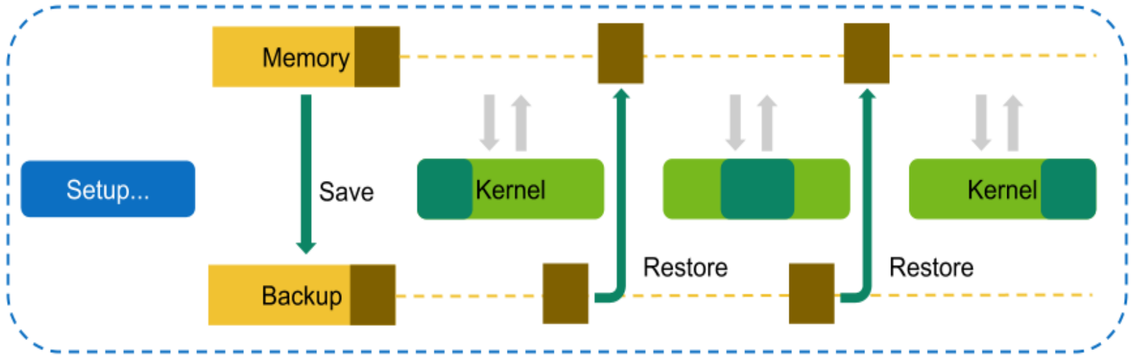
Sections/Rules | Reload | Enable All | Disable All | Restore

Enter filter

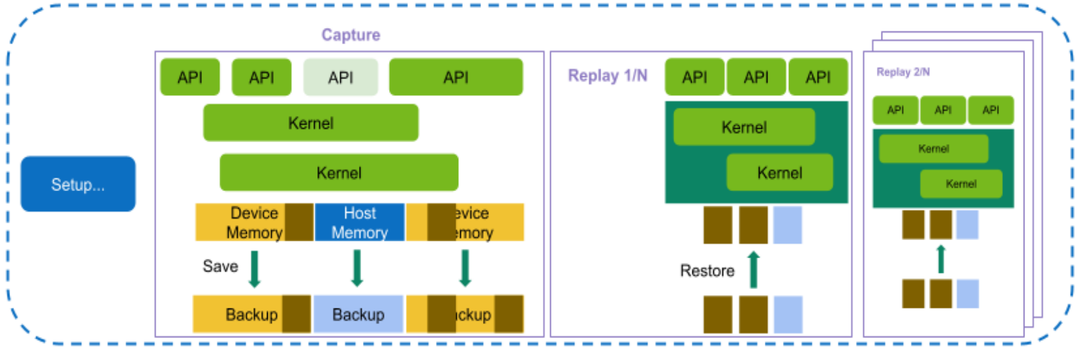
	Name	Priority	Description
☑	GPU Speed Of Light Throughput (1)	10	High-level overview of the throughput for compu...
☑	GPU Speed Of Light Roofline Chart (1)	11	High-level overview of the utilization for comput...
☑	GPU Speed Of Light Hierarchical Roofline Chart (Double Precision)	12	High-level overview of the utilization for comp...
☑	GPU Speed Of Light Hierarchical Roofline Chart (Half Precision)	12	High-level overview of the utilization for comput...
☑	GPU Speed Of Light Hierarchical Roofline Chart (Single Precision)	12	High-level overview of the utilization for comput...
☑	GPU Speed Of Light Hierarchical Roofline Chart (Tensor Core)	12	High-level overview of the utilization for comput...
☐	Compute Workload Analysis (2)	20	Detailed analysis of the compute resources of t...

REPLAY MODES

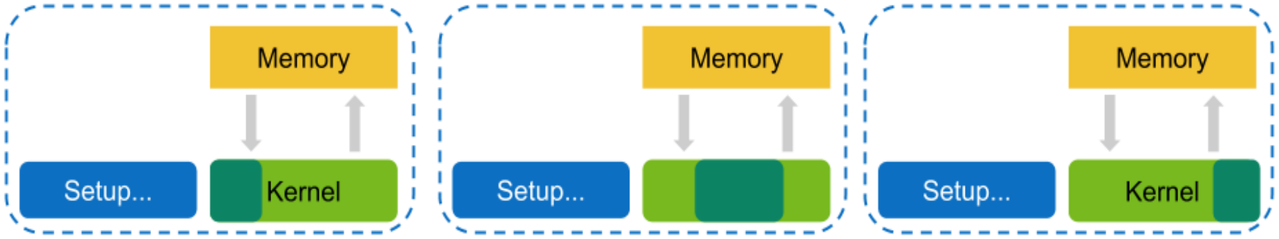
https://docs.nvidia.com/nsight-compute/ProfilingGuide/index.html#replay



Kernel Replay
(interactive and non-interactive)



Range Replay
(non-interactive)



Application Replay
(non-interactive)

FURTHER INFORMATION

Download

https://developer.nvidia.com/cuda-downloads (packaged in the CUDA Toolkit)

https://developer.nvidia.com/nsight-systems

https://developer.nvidia.com/nsight-compute

Documentation

https://docs.nvidia.com/nsight-systems/

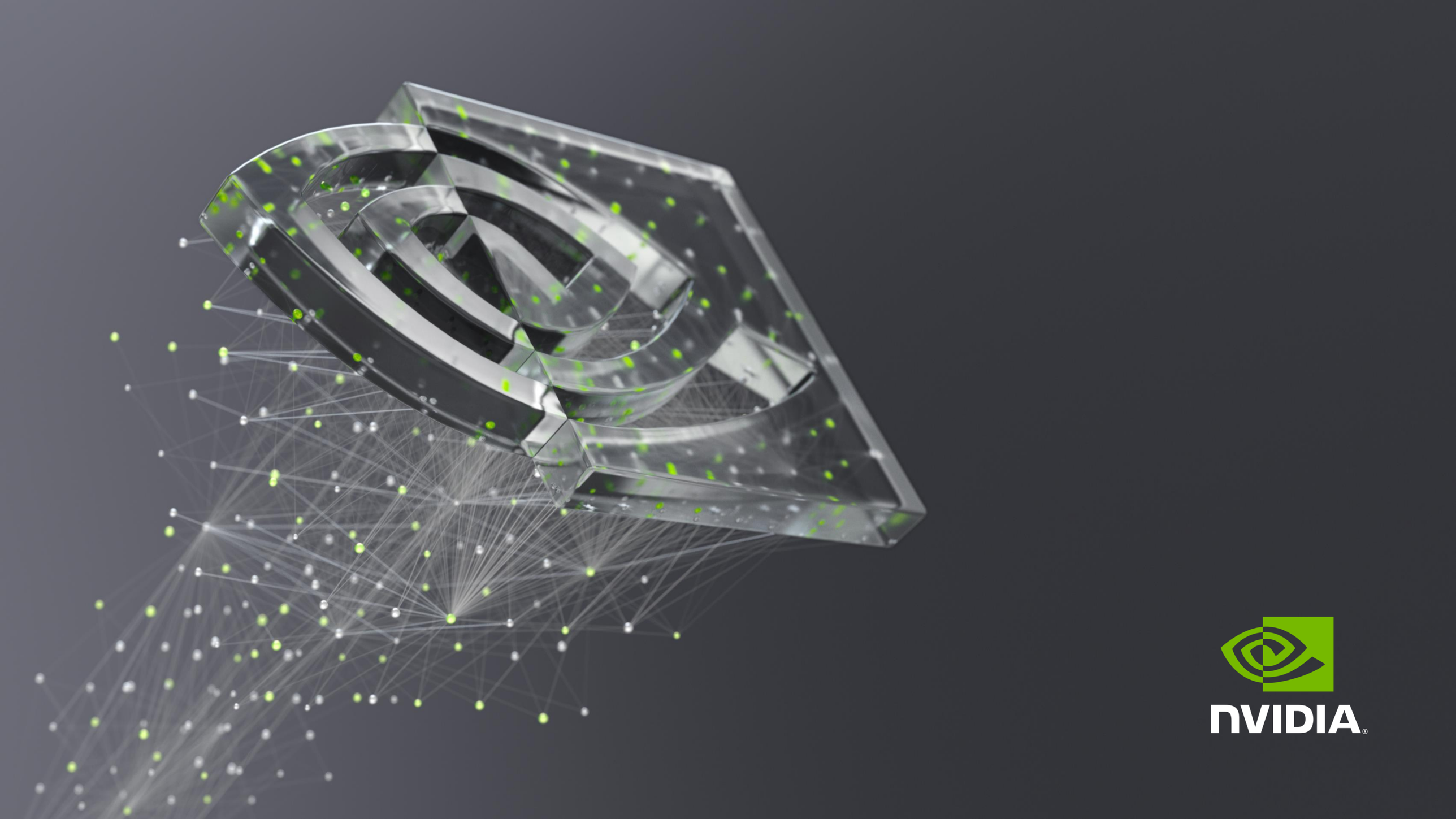
https://docs.nvidia.com/nsight-compute

Support is available via:

https://forums.developer.nvidia.com/c/development-tools/

More information at:

https://developer.nvidia.com/tools-overview


NVIDIA