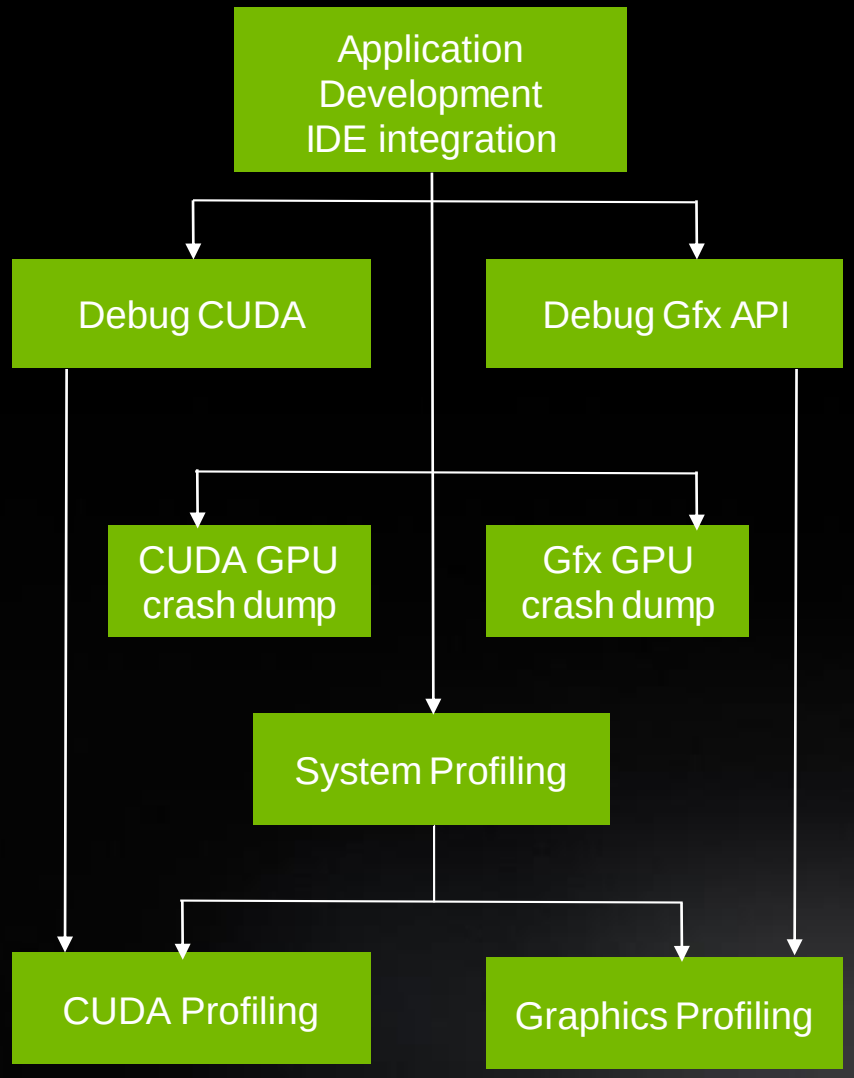
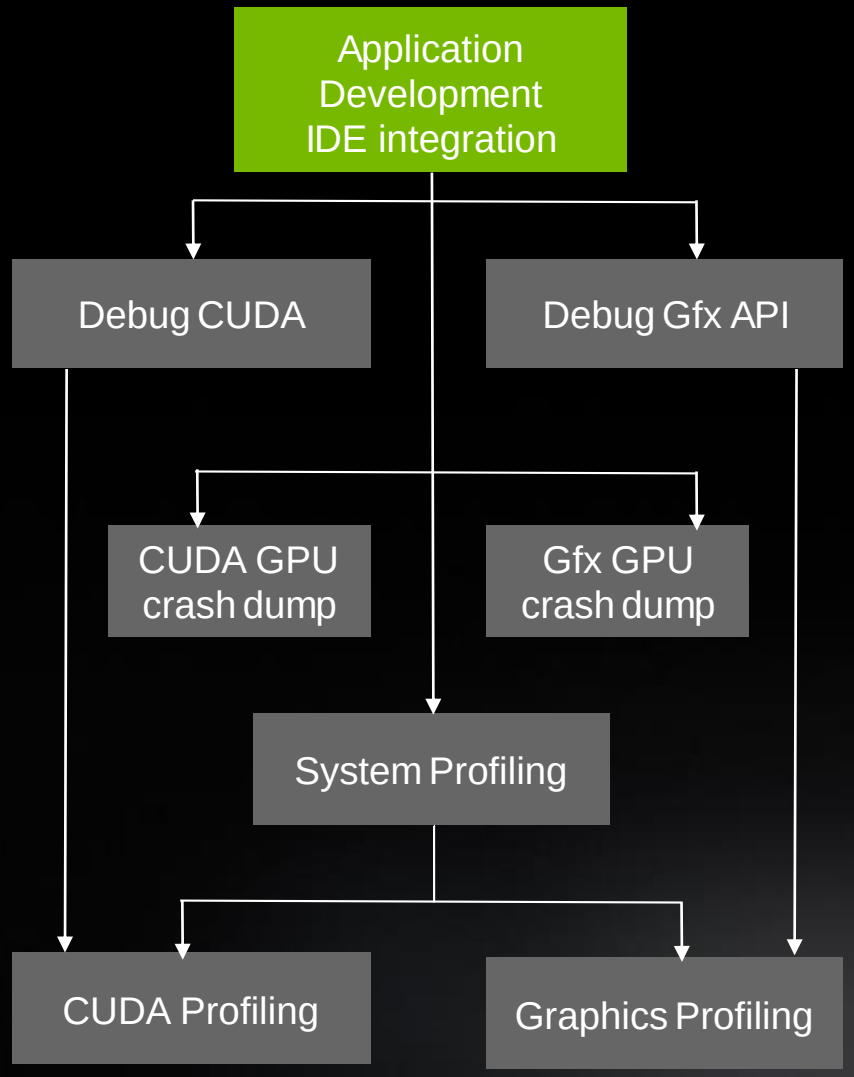
CUDA DEVELOPER TOOLS: NEW FEATURES AND CAPABILITIES

Rafael Campana, Nov. 19, 2019

DEVELOPER TOOLS PORTFOLIO

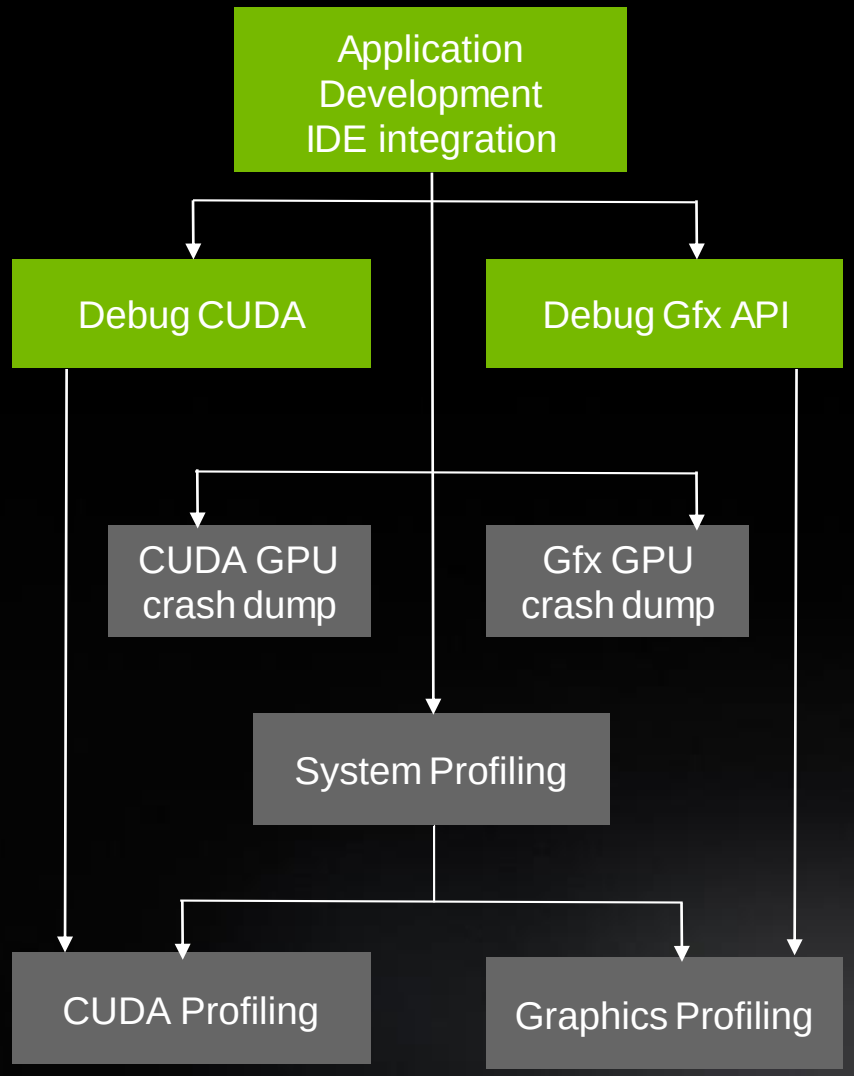


DEVELOPER TOOLS PORTFOLIO



Nsight Eclipse Edition
Nsight Visual Studio Edition

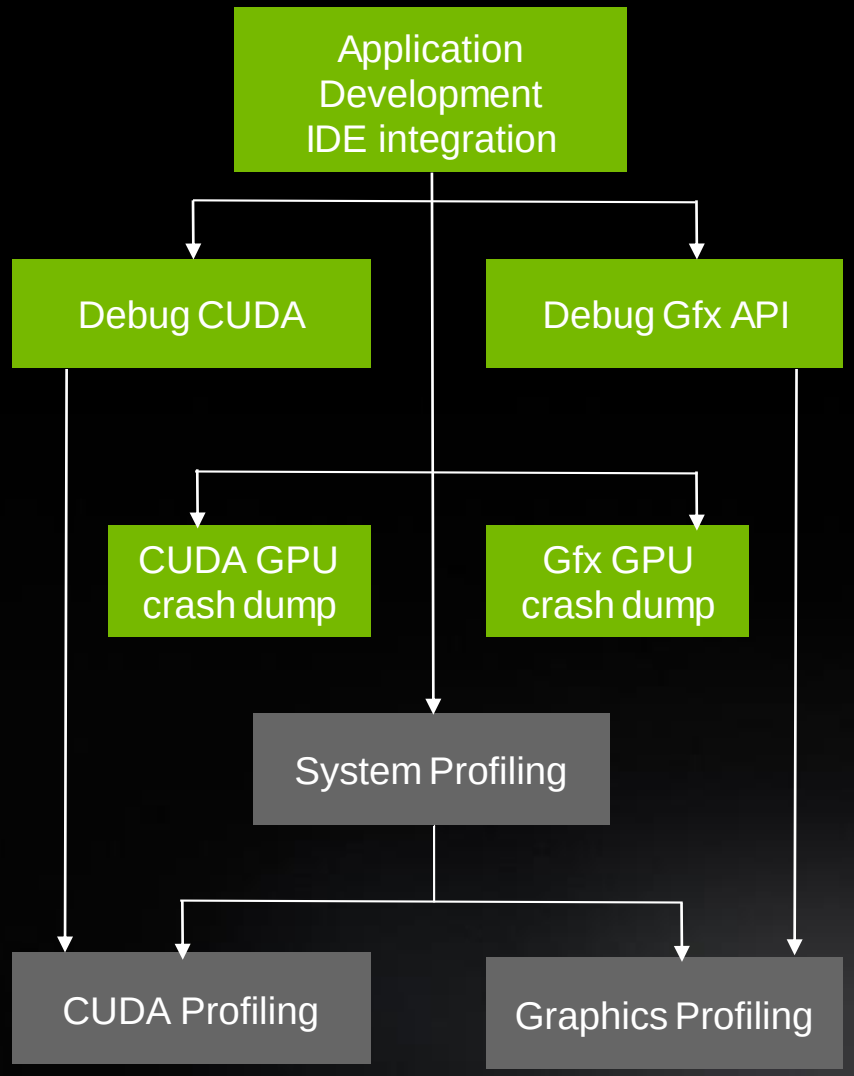
DEVELOPER TOOLS PORTFOLIO



Nsight Eclipse Edition
Nsight Visual Studio Edition

Nsight Eclipse Edition
cuda-gdb
Nsight Visual Studio Edition
Nsight Compute
cuda-memcheck & Compute Sanitizer API
Nsight Graphics

DEVELOPER TOOLS PORTFOLIO



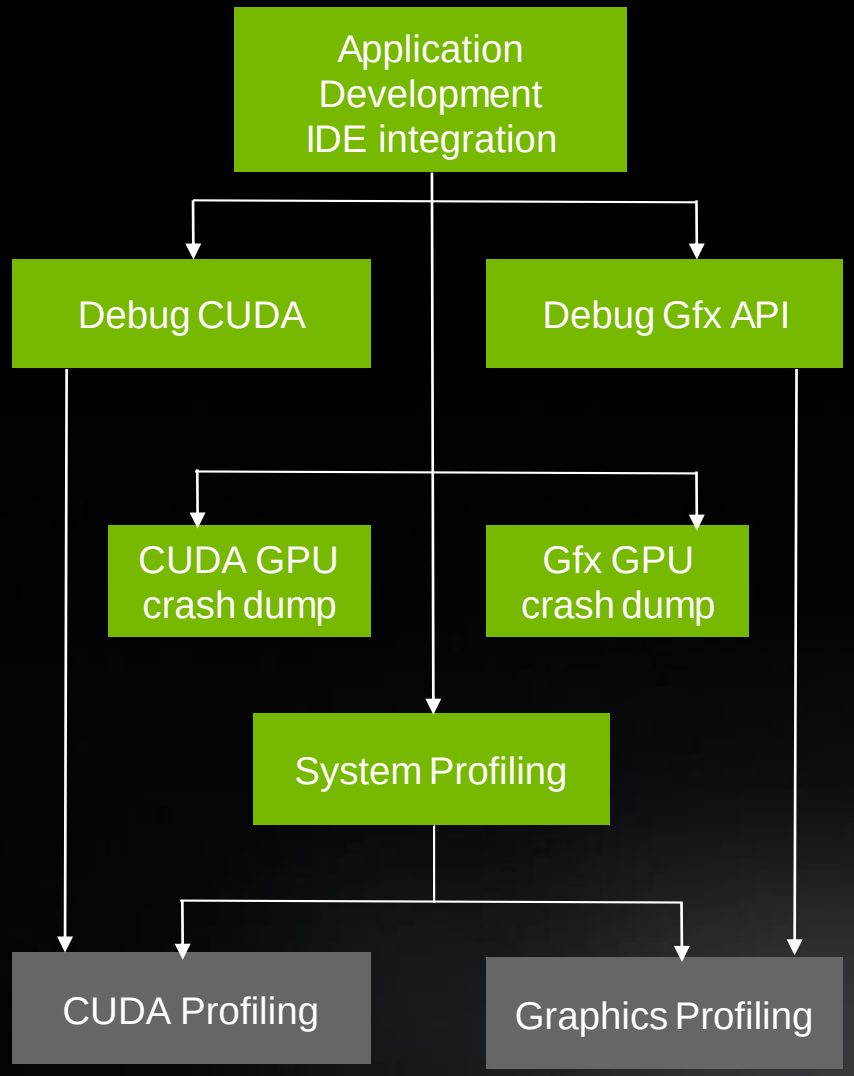
Nsight Eclipse Edition
Nsight Visual Studio Edition

Nsight Eclipse Edition
cuda-gdb
Nsight Visual Studio Edition
Nsight Compute
cuda-memcheck & Compute Sanitizer API
Nsight Graphics

---

cuda-gdb/Nsight Eclipse Edition
Nsight Visual Studio Edition
Nsight Aftermath

DEVELOPER TOOLS PORTFOLIO



Nsight Eclipse Edition
Nsight Visual Studio Edition

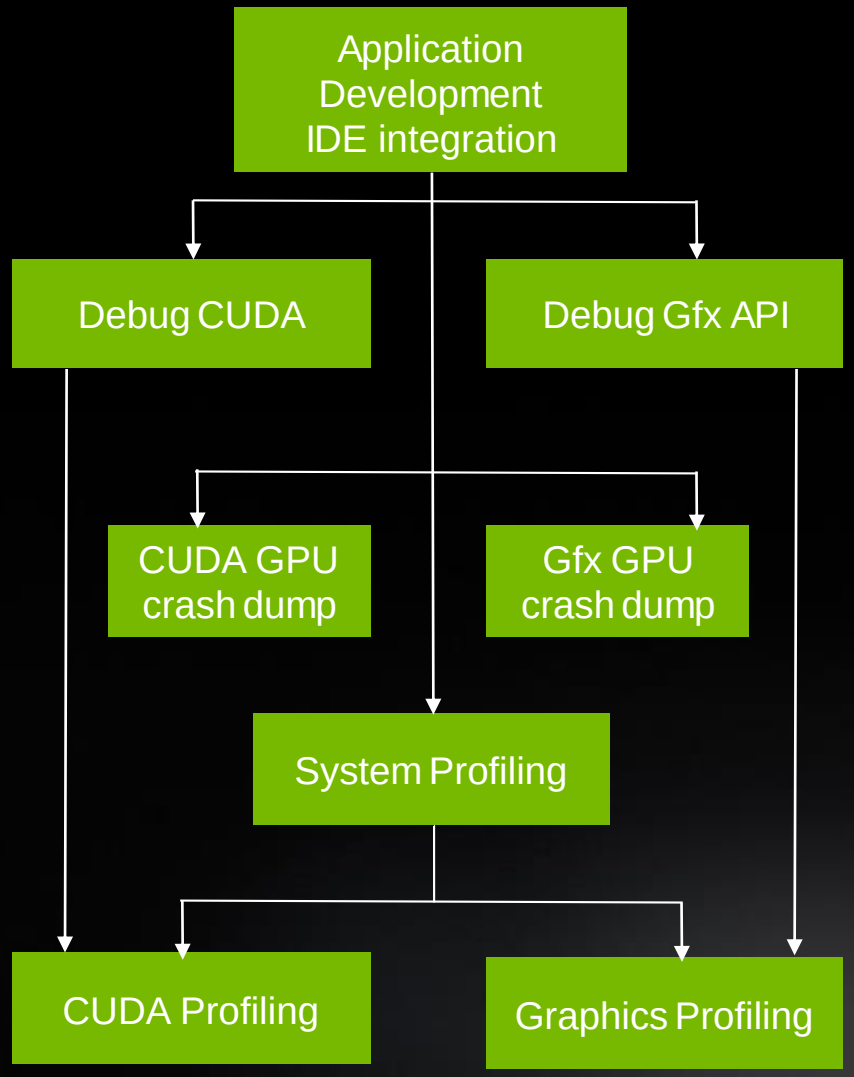
Nsight Eclipse Edition
cuda-gdb
Nsight Visual Studio Edition
Nsight Compute
cuda-memcheck & Compute Sanitizer API
Nsight Graphics

---

cuda-gdb/Nsight Eclipse Edition
Nsight Visual Studio Edition
Nsight Aftermath

Nsight Systems

DEVELOPER TOOLS PORTFOLIO



Nsight Eclipse Edition
Nsight Visual Studio Edition

Nsight Eclipse Edition
cuda-gdb
Nsight Visual Studio Edition
Nsight Compute
cuda-memcheck & Compute Sanitizer API
Nsight Graphics

---

cuda-gdb/Nsight Eclipse Edition
Nsight Visual Studio Edition
Nsight Aftermath

Nsight Systems

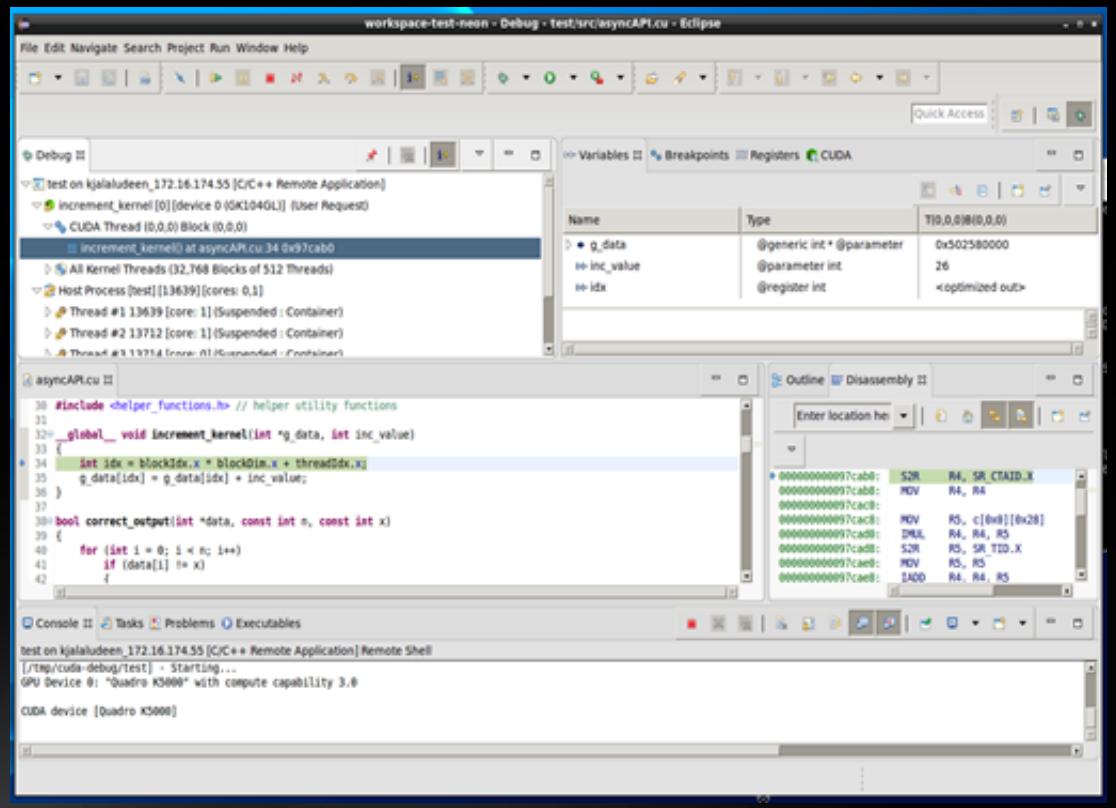
Nsight Compute

Nsight Graphics

NVIDIA® NSIGHT™ ECLIPSE EDITION

Plug-in

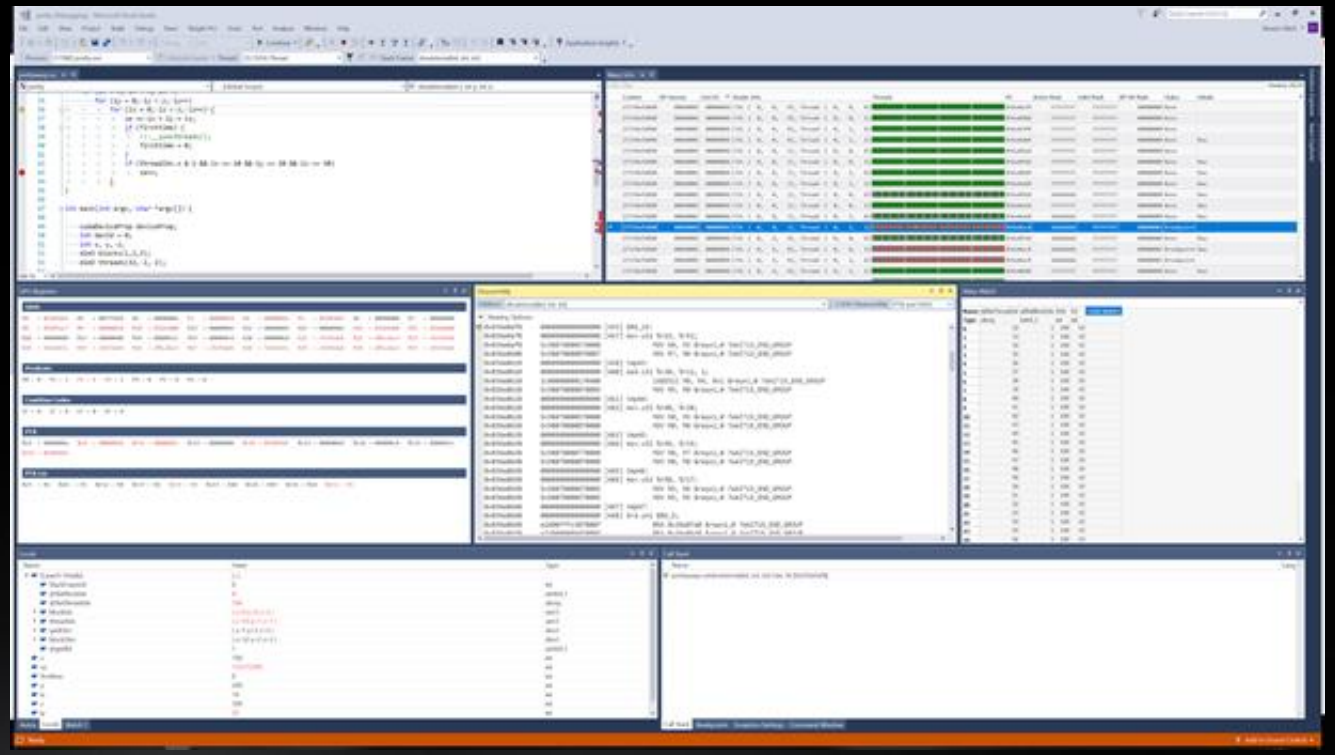
- Plug-in to Eclipse
  Eclipse 4.7, 4.8 and 4.9 support
- Edit, build, debug CUDA-C applications
- CUDA aware source code editor – syntax highlighting, code completion and inline help
- Debugger - Seamless and simultaneous debugging of CPU and GPU code
- NVCC build integration to cross compile for various target platforms
- Docker support



- Documentation: https://docs.nvidia.com/cuda/nsight-eclipse-plugins-guide

NSIGHT VISUAL STUDIO EDITION

Plug-in



- Visual Studio 2015, 2017 and 2019 support
- Native CUDA C/C++ GPU Debugging
- Source-correlated assembly debugging (SASS / PTX / SASS+PTX)
- Data breakpoints for CUDA C/C++ code
- Expressions in Locals, Watch and Conditionals
- CUDA info view
- Warp Watch

Documentation: https://developer.nvidia.com/nsight-visual-studio-edition

CUDA-GDB

Overview

Provides the debugging features of Nsight Eclipse Edition

Command line source and assembly (SASS) level debugger

Simultaneous CPU and GPU debugging

Inspect and modify memory, register, variable state

Control program execution

Runtime GPU error detection

```
(cuda-gdb) info cuda threads breakpoint all
  BlockIdx ThreadIdx         Virtual PC Dev SM Wp Ln        Filename  Line
Kernel 0
   (1,0,0)   (0,0,0) 0x0000000000948e58   0 11  0  0 infoCommands.cu   12
   (1,0,0)   (1,0,0) 0x0000000000948e58   0 11  0  1 infoCommands.cu   12
   (1,0,0)   (2,0,0) 0x0000000000948e58   0 11  0  2 infoCommands.cu   12
   (1,0,0)   (3,0,0) 0x0000000000948e58   0 11  0  3 infoCommands.cu   12
   (1,0,0)   (4,0,0) 0x0000000000948e58   0 11  0  4 infoCommands.cu   12
   (1,0,0)   (5,0,0) 0x0000000000948e58   0 11  0  5 infoCommands.cu   12
(cuda-gdb) info cuda threads breakpoint 2 lane 1
  BlockIdx ThreadIdx         Virtual PC Dev SM Wp Ln        Filename  Line
Kernel 0
   (1,0,0)   (1,0,0) 0x0000000000948e58   0 11  0  1 infoCommands.cu   12
```

Support for multiple GPUs, multiple contexts, multiple kernels, Thread focus

Core dump support

Documentation : http://docs.nvidia.com/cuda/cuda-gdb

CUDA-GDB

New Features

Support for ARM (Server Base System Architecture) *preview*

Performance improvements

- Module load time (~30% faster)

Quality improvements

- Improved handling of `--lineinfo` debug information (OptiX)
- Improved display of uniform registers (Turing)

CUDA-MEMCHECK

Functional correctness checking tool suite

Multiple tools

- memcheck : reports out of bounds/misaligned memory access errors
- racecheck : identifies races on `__shared__` memory
- initcheck : usage of uninitialized global memory
- synccheck : identify invalid usage of `__syncthreads()` and `__syncwarp()`

Documentation: http://docs.nvidia.com/cuda/cuda-memcheck/index.html

SANITIZER

API

Provides finer control than cuda-memcheck through APIs to analyze memory patterns

APIs are grouped in two:

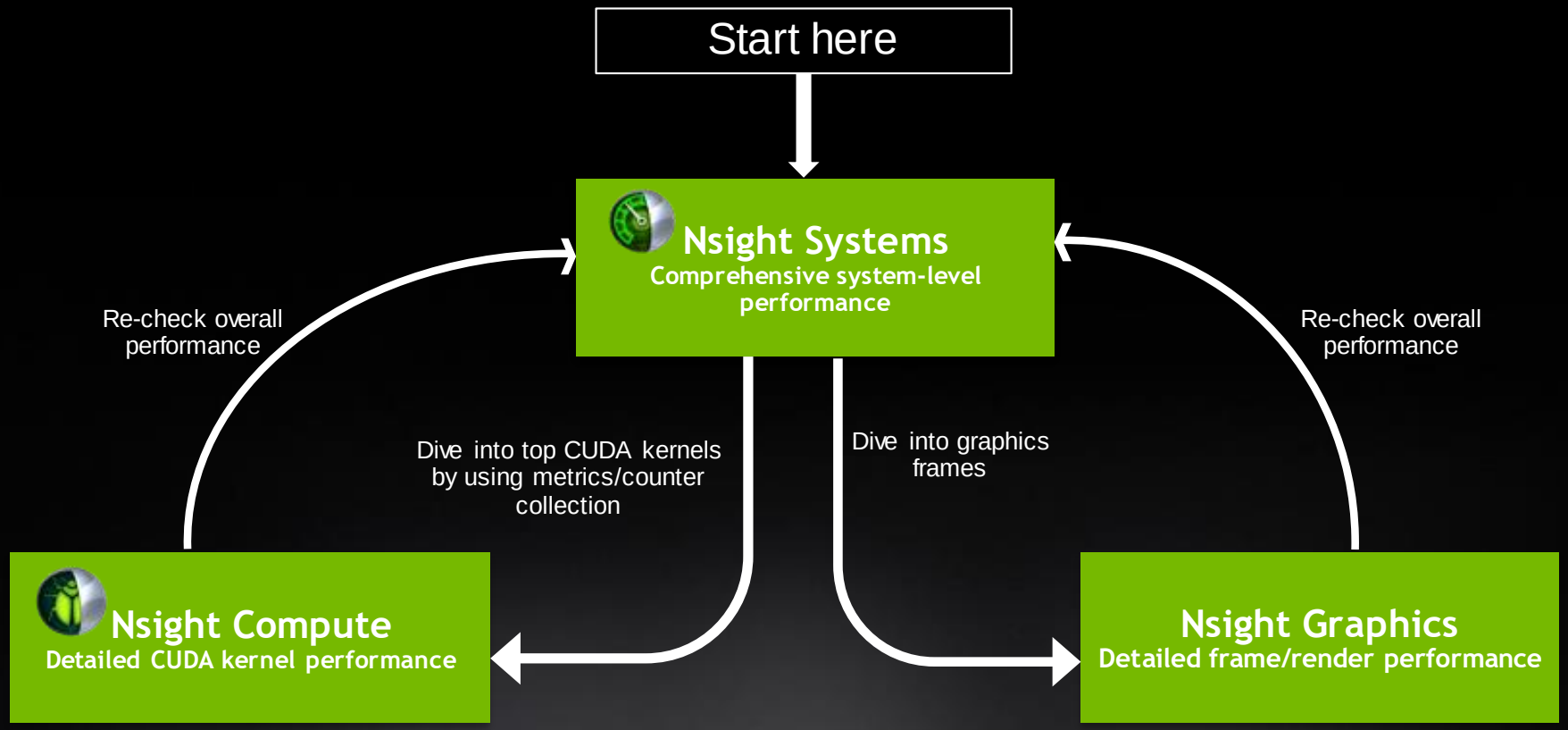
- Callback API – CUDA events such as memory allocations/kernel
- Patching API – inserts patches for specific memory instructions

Documentation: https://docs.nvidia.com/cuda/compute-sanitizer/index.html

Samples: https://github.com/NVIDIA/compute-sanitizer-samples

NSIGHT SUITE OF PROFILERS

NSIGHT TOOLS WORKFLOW



NSIGHT SYSTEMS

Overview

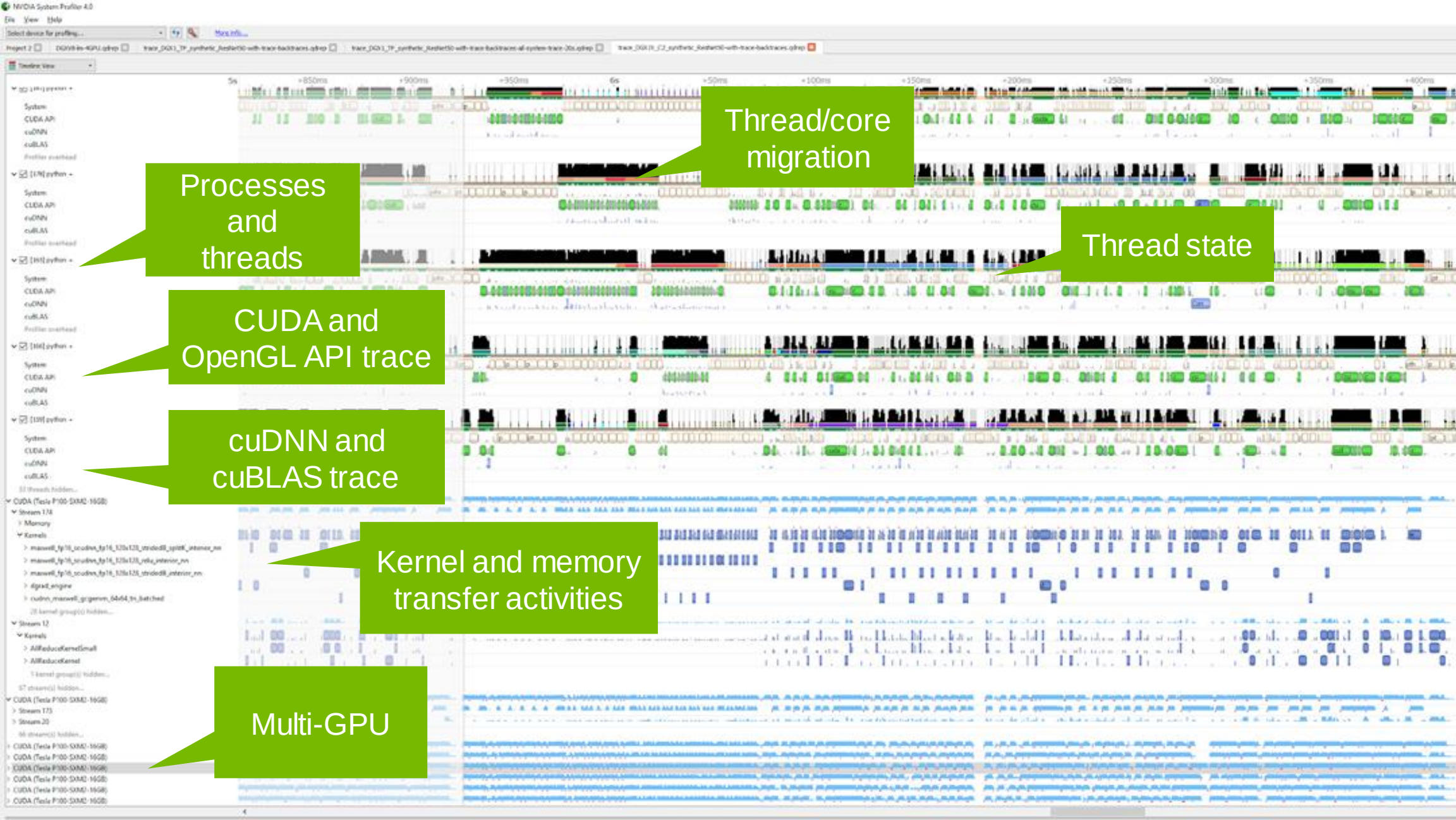
System-wide application algorithm tuning
- Multi-process tree support

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

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

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


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

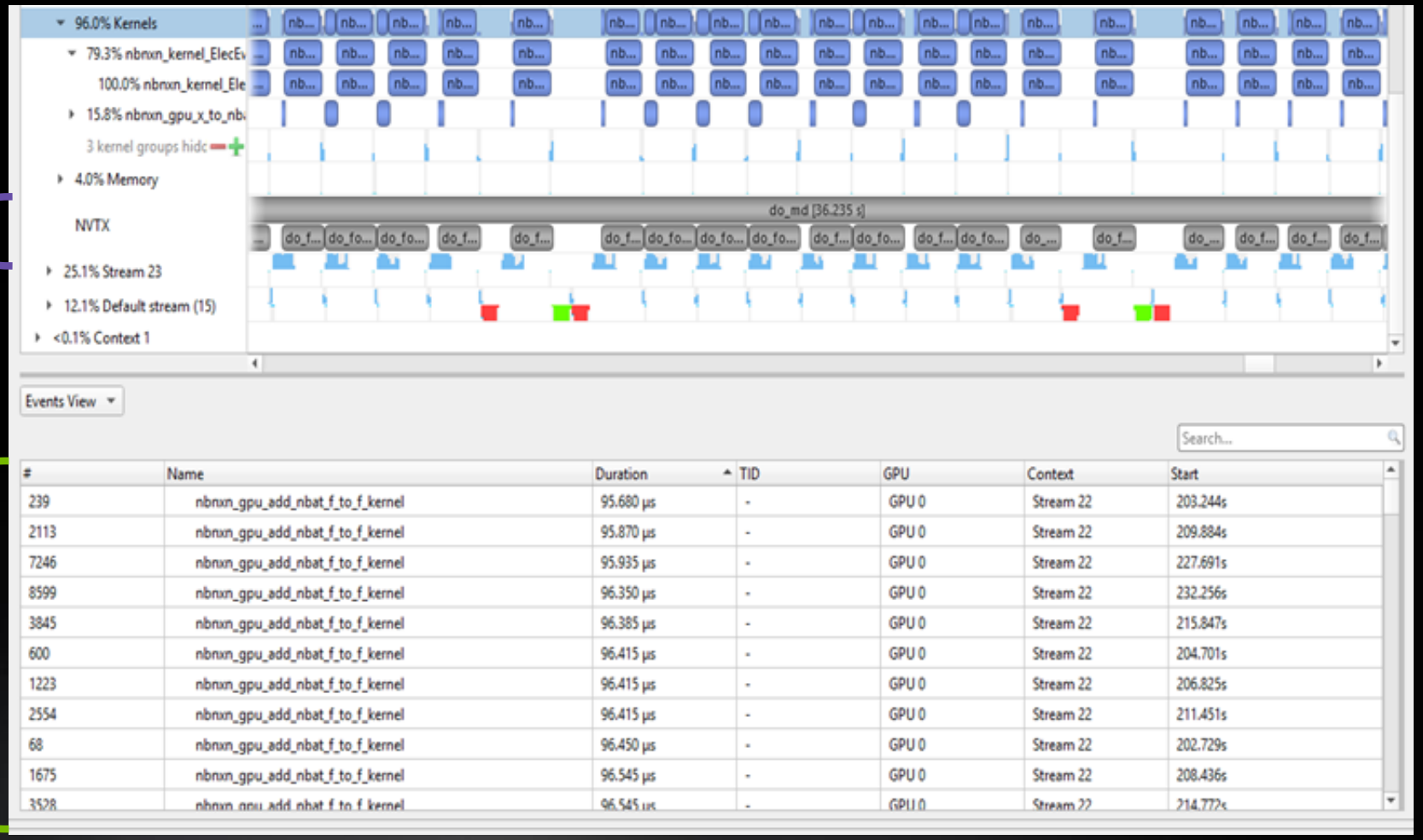
NSIGHT SYSTEMS

New Features

- ARM (SBSA) support
- ftrace collection

- NVTX correlated to GPU

- Event table



#	Name	Duration	TID	GPU	Context	Start
239	nbnxn_gpu_add_nbat_f_to_f_kernel	95.680 µs	-	GPU 0	Stream 22	203.244s
2113	nbnxn_gpu_add_nbat_f_to_f_kernel	95.870 µs	-	GPU 0	Stream 22	209.884s
7246	nbnxn_gpu_add_nbat_f_to_f_kernel	95.935 µs	-	GPU 0	Stream 22	227.691s
8599	nbnxn_gpu_add_nbat_f_to_f_kernel	96.350 µs	-	GPU 0	Stream 22	232.256s
3845	nbnxn_gpu_add_nbat_f_to_f_kernel	96.385 µs	-	GPU 0	Stream 22	215.847s
600	nbnxn_gpu_add_nbat_f_to_f_kernel	96.415 µs	-	GPU 0	Stream 22	204.701s
1223	nbnxn_gpu_add_nbat_f_to_f_kernel	96.415 µs	-	GPU 0	Stream 22	206.825s
2554	nbnxn_gpu_add_nbat_f_to_f_kernel	96.415 µs	-	GPU 0	Stream 22	211.451s
68	nbnxn_gpu_add_nbat_f_to_f_kernel	96.450 µs	-	GPU 0	Stream 22	202.729s
1675	nbnxn_gpu_add_nbat_f_to_f_kernel	96.545 µs	-	GPU 0	Stream 22	208.436s
3528	nbnxn_gpu_add_nbat_f_to_f_kernel	96.545 µs	-	GPU 0	Stream 22	214.772s

NSIGHT SYSTEMS

MPI API Trace

Project 8 | laplace2d_0.qdrep

Timeline View

1x

13 messages

+755ms +756ms +757ms +758ms +759ms +760ms +761ms +762ms

CPU (12)

Threads (10)

[3394] MPI Rank 0

MPI — MPI_Sendrecv [52.485 ms] | MPI_Sendrecv [1.341 ...

CUDA API — cuMemHostRegister | c... | culpcOpenMe...

Profiler overhead

9 threads hidden...

CUDA (MPI Rank 0)

88.2% Stream 15

96.9% Kernels

3.1% Memory

10.2% Stream 16

100.0% Kernels

100.0% main_123_gpu

100.0% main_123_gpu

1.6% Stream 17

100.0% Memory

NVTX (MPI)

<0.1% Stream 20

100.0% Memory

NVTX (MPI)

4,800

Collect MPI trace

Select the MPI implementation used by the target application to trace a default set of synchronous MPI calls. If the application uses a different MPI implementation, see the documentation for additional setup required to trace MPI. Note that NVTX tracing will also be enabled on selecting MPI tracing.

- (•) OpenMPI
- ( ) MPICH and its derivatives

Collect NVTX trace

NSIGHT SYSTEMS

New Trace Collections

NVIDIA Nsight Systems 2019.4.0

File View Tools Help

Project Explorer

Project 1

Project 1

workstation-All-Series-Invalid-entry-len

Target is ready More info...

Collect OS runtime libraries trace

Skip if shorter than 1.000 microseconds

Userspace tracing of Operating System runtime libraries that provide interfaces to communicate with the hardware or control resources such as threads and processes. This traces the system call wrappers and thread synchronization interfaces exposed by the C runtime and POSIX Threads (pthread) libraries.

Collect OpenGL trace

Collect CUDA trace

Collect backtraces for API calls longer than 1.000 microseconds

Flush data periodically 10.00 seconds

Skip some API calls

Collect cuDNN trace

Collect cuBLAS trace

Collect OpenACC trace

Collect FTrace events

Choose FTrace events...

No events selected

Collect MPI trace

Collect NVTX trace

Collect Vulkan trace

Install debug symbol packages on the target to get more symbols resolved in system libraries.
See Ubuntu wiki for more info.

Start

Start profiling manually

Start profiling after 10.0 seconds

Limit

Choose functions

Search criteria:

Check all Uncheck all

Search...

Events

alarmtimer
block
cgroup
clk
cma
compaction
cpuhp
dma_fence
drm
exceptions
ext4
fib
fib6
filelock
filemap
fjes
fs
gpio
hda
hda_controller
hda_intel
huge_memory
i2c
iommu
irq
irq_vectors
jbd2
kmem
kvm
kvmmmu

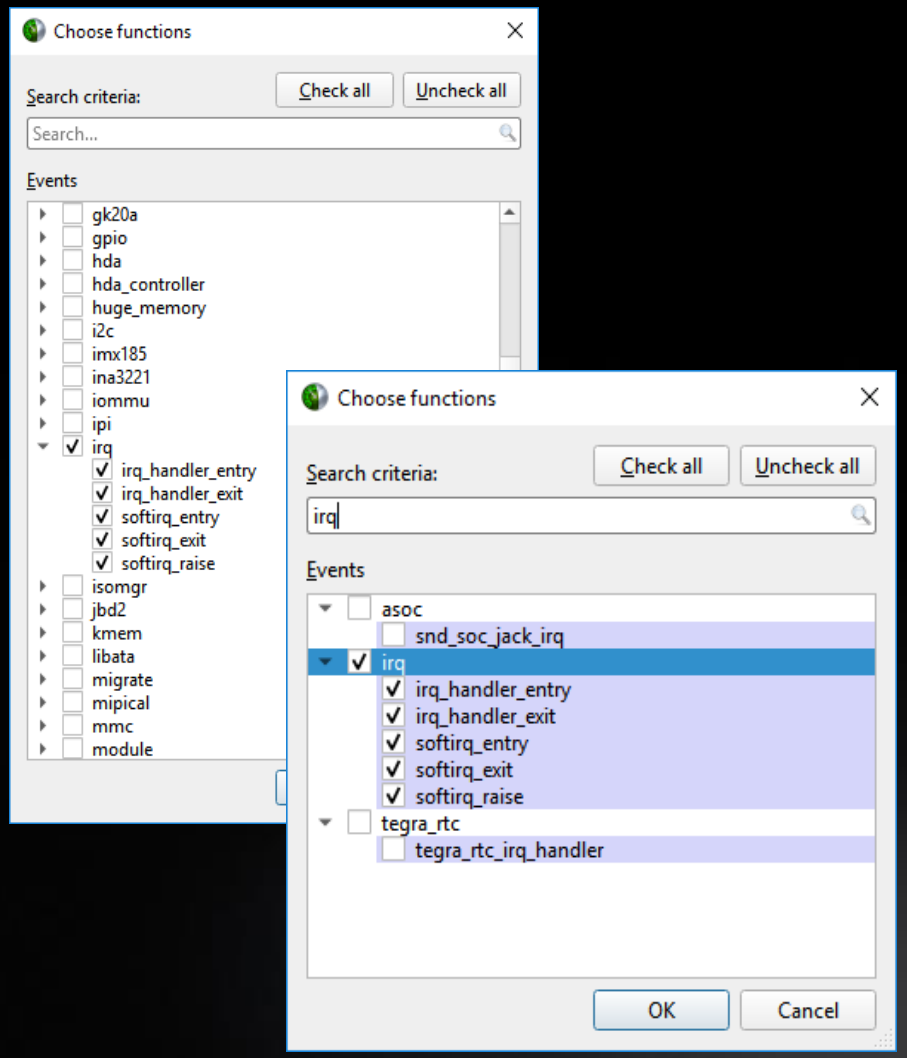
OK Cancel

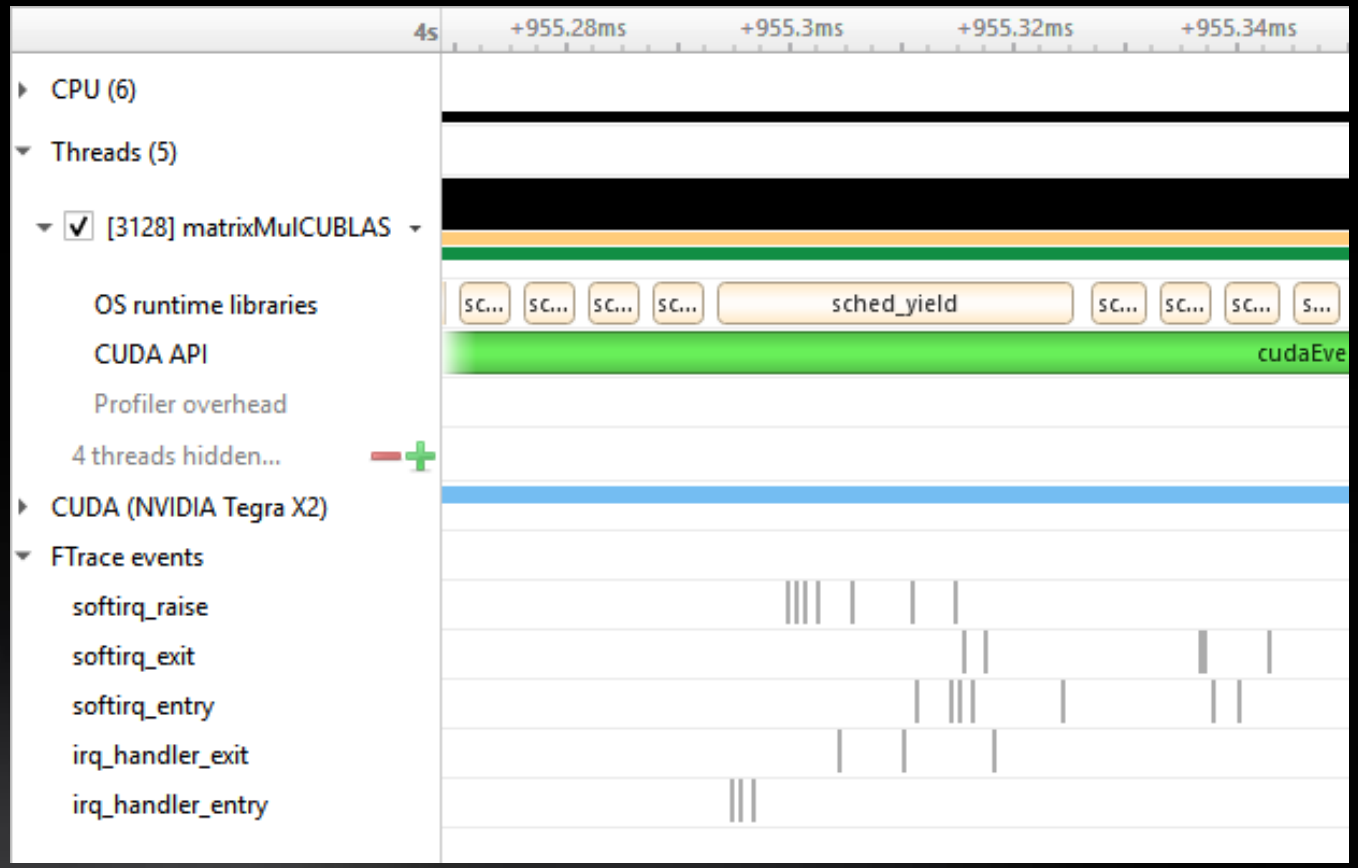
CUDA Kernel Backtraces

FTrace events

NSIGHT SYSTEMS

FTrace





NSIGHT COMPUTE

Next-Gen Kernel Profiling Tool

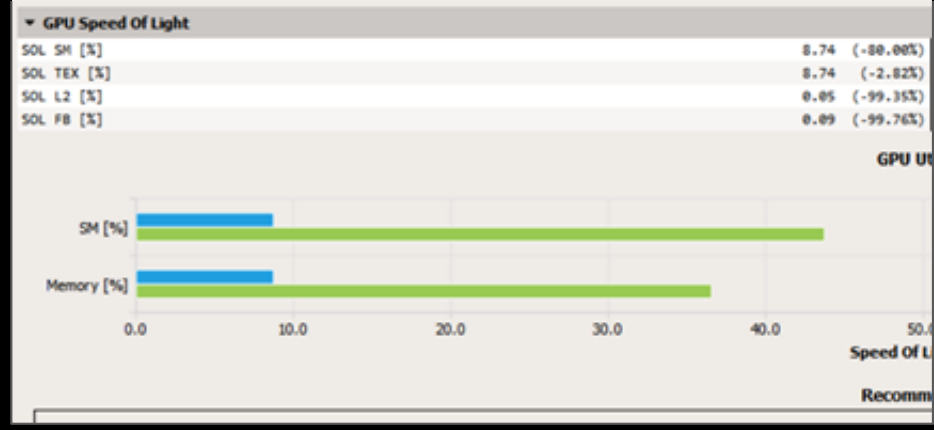
Key Features:

- Interactive CUDA API debugging and kernel profiling
- Fast Data Collection
- Improved Workflow (Diff'ing Results)
- Fully Customizable (Programmable UI/Rules)
- Command Line, Standalone, IDE Integration

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

GPUs: Pascal, Volta, Turing

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



inst_executed [inst]	16,528.00; 16,528.00; …	13,476.00; 13,476.00; …
l1tex__sol_pct [%]	14.33	n/a
launch__block_size	128.00	128.00
launch__function_pcs	47,611,587,968.00	12,273,728.00
launch__grid_size	4,132.00	3,369.00
launch__occupancy_limit_blocks [block]	32.00	32.00
launch__occupancy_limit_registers [register]	21.00	21.00
launch__occupancy_limit_shared_mem [bytes]	384.00	384.00
launch__occupancy_limit_warps [warps]	16.00	16.00
launch__occupancy_per_block_size	3,638.00	3,638.00
launch__occupancy_per_register_count	5,792.00	5,792.00
launch__occupancy_per_shared_mem_size	2,260.00	2,260.00
launch__registers_per_thread [register/thread]	17.00	17.00
launch__shared_mem_config_size [bytes]	49,152.00	49,152.00
launch__shared_mem_per_block_dynamic [bytes/block]	0.00	0.00
launch__shared_mem_per_block_static [bytes/block]	20.00	20.00
launch__thread_count [thread]	528,896.00	431,232.00
launch__waves_per_multiprocessor	3.23	42.11
ltc__sol_pct [%]	6.93	7.18
memory_access_size_type [bytes]	2.00; 32.00; 32.00; 32…	2.00; 32.00; 32.00; 32…

Source	Live Registers	Sampling Data (All)	Sampling Data (No Issue)
@!PT SHFL.IDX PT, RZ, RZ, RZ, RZ;	0	223	0
MOV R1, c[0x0][0x28];	1	13	44
S2R R0, SR_CTAID.X;	2	143	75
S2R R2, SR_TID.X;	3	0	38
IMAD R0, R0, c[0x0][0x0], R2;	3	599	94
ISETP.GE.AND P0, PT, R0, c[0x0][0x170]	2	125	26
@P0 EXIT;	2	259	86
MOV R2, R0;	3	386	29
@!PT SHFL.IDX PT, RZ, RZ, RZ, RZ;	2	0	0
MOV R4, 0x4;	3	0	0
IMAD.WIDE R4, R2, R4, c[0x0][0x160];	4	0	0
LDG.E.SYS R3, [R4];	3	0	0
BSSY B0, 0xb00976780;	3	0	0
SHF.R.S32.HI R0, RZ, 0x1f, R2;	4	0	0

NSIGHT COMPUTE

Profile Report – Details Page

All Data on Single Page

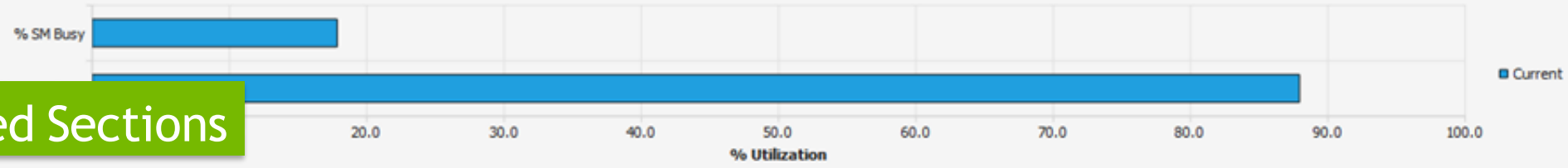
GPU Speed Of Light

% SOL SM	17.84	Duration (Nanoseconds)	709,056.00
% SOL TEX	17.84	Elapsed Cycles	1,761,844.00
% SOL L2	15.08	SM Frequency (Hz)	1,242,387,061.11
% SOL FB	87.94	Memory Frequency (Hz)	2,499,503,565.30

Recommendations

Bottleneck Simple GPU bottleneck detection. Apply

GPU Utilization



Focused Sections

Compute Workload Analysis

Executed Ipc Elapsed	0.71	% SM Busy	17.84
Executed Ipc Active	0.72	% Issue Slots Busy	11.94
Issued Ipc Active	0.72		

Memory Workload Analysis

Memory Throughput (bytes/s)	70,335,950,898.10	% Mem Busy	87.94
% L1 Hit Rate	0.00	% Max Bandwidth	87.94
% L2 Hit Rate	33.34	% Mem Pipes Busy	17.90

Scheduler Statistics

Active Warps Per Scheduler	13.20	Instructions Per Active Issue Slot	1.04
Eligible Warps Per Scheduler	0.25	% No Eligible	82.73
Issued Warps Per Scheduler	0.18	% One or More Elig	17

Warp State Statistics

Cycles Per Issued Instruction	72.86	Avg. Active Thread	10
Cycles Per Issue Slot	76.02	Avg. Not Predicate	37
Cycles Per Executed Instruction	72.87		

Ordered from Top-Level to Low-Level

NVIDIA

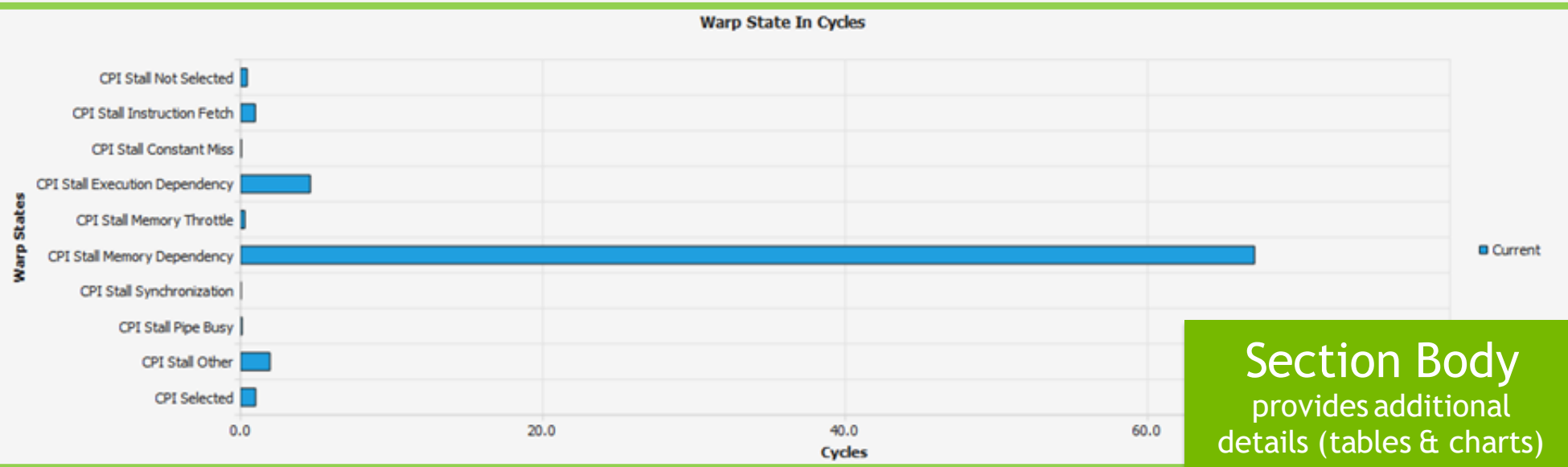
NSIGHT COMPUTE

Section Example

Section Header
provides overview & context for other sections

Warp State Statistics

Cycles Per Issued Instruction	72.86	Avg. Active Threads Per Warp	32.00
Cycles Per Issue Slot	76.02	Avg. Not Predicated Off Threads Per Warp	30.67
Cycles Per Executed Instruction	72.87		



Section Body
provides additional details (tables & charts)

Section Config
completely data driven
add/modify/change sections

NSIGHT COMPUTE

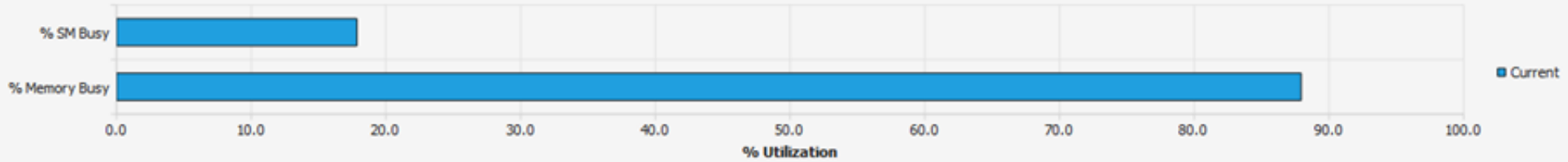
Unguided Analysis / Rules System

Analysis Rules
recommendations from nvvp and more

17.84	Duration (Nanoseconds)	709,056.00
17.84	Elapsed Cycles	1,761,844.00
15.08	SM Frequency (Hz)	1,242,387,061.11
87.94	Memory Frequency (Hz)	2,499,503,565.30

Recommendations

Bottleneck [Warning] Memory is more heavily utilized than Compute: Look at `Memory Workload Analysis` report section to see where the memory system bottleneck is. Check memory replay (coalescing) metrics to make sure you're efficiently utilizing the bytes transferred. Also consider whether it is possible to do more work per memory access (kernel fusion) or whether there are values you can recompute.



Rules Config
completely data driven
add/modify/change rules

NSIGHT COMPUTE

Diff'ing kernel runs

Baseline
from any previous profile report
(different kernel, gpu, …)

Metric delta
current values and changes
from baseline

Chart difference
current values and
baseline values

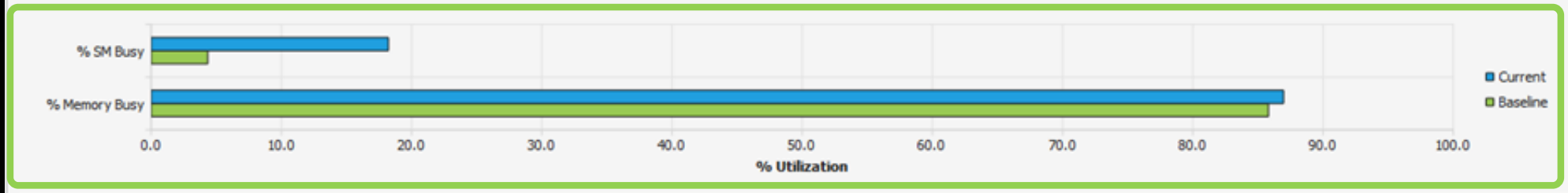
GPU Speed Of

SOL SM	18.23	(+318.69%)	Duration (Nanoseconds)	190,208.00	(-76.03%)
SOL TEX	18.23	(+39.56%)	Elapsed Cycles	470,600.00	(-76.12%)
SOL L2	15.42	(+4.81%)	SM Frequency (Hz)	1,237,066,790.04	(-0.37%)
SOL FB	86.97	(+1.36%)	Memory Frequency (Hz)	2,499,327,052.49	(-0.02%)

Recommendations

Bottleneck Simple GPU bottleneck detection. Apply

GPU Utilization



	0.61	(+318.69%)	% SM Busy	18.23	(+318.69%)
	0.61	(+321.96%)	% Issue Slots Busy	10.25	(+321.96%)
	0.61	(+321.96%)			
	69,557,705,248.99	(+1.33%)	% Mem Busy	86.97	(+1.36%)
% L1 Hit Rate	100.00	(+0.00%)	% Max Bandwidth	86.97	(+1.36%)
% L2 Hit Rate	33.34	(+0.02%)	% Mem Pipes Busy	18.43	(+321.96%)

Scheduler Statistics

Active Warps Per Scheduler	13.83	(+1.73%)	Instructions Per Active Issue Slot	1.11	(+0.00%)
Eligible Warps Per Scheduler	0.17	(+383.47%)	% No Eligible	86.06	(-11.00%)
Issued Warps Per Scheduler	0.15	(+322.90%)	% One or More Eligible	13.92	(+322.90%)

Warp State Statistics

Cycles Per Issued Instruction	89.24	(-75.99%)	Avg. Active Threads Per Warp	32.00	(+0.00%)
Cycles Per Issue Slot	99.14	(-75.99%)	Avg. Not Predicated Off Threads Per Warp	30.40	(+0.00%)

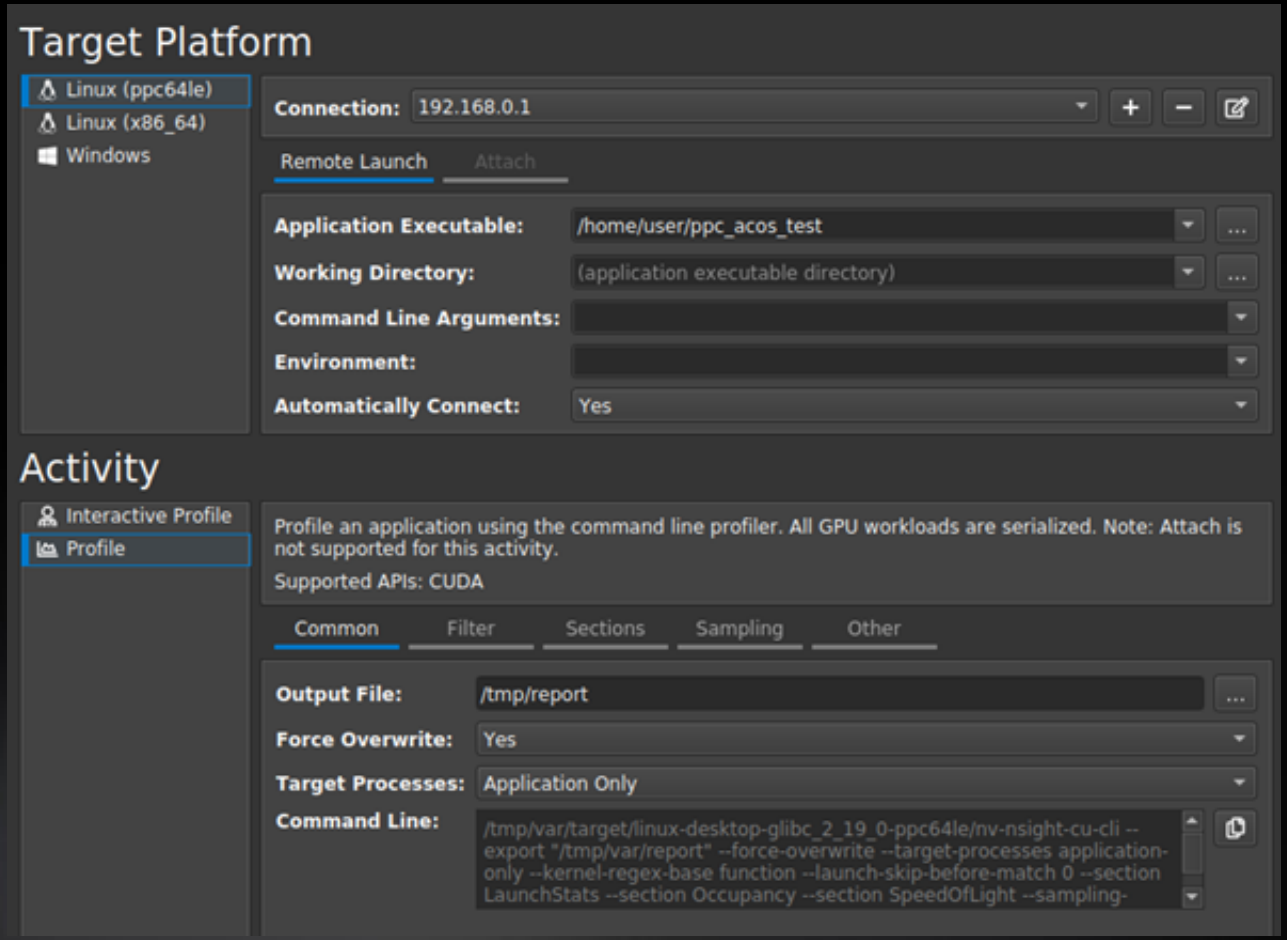
NSIGHT COMPUTE

New Features

Support for PowerPC target architecture

Support for ARM (SBSA) *preview* target architecture

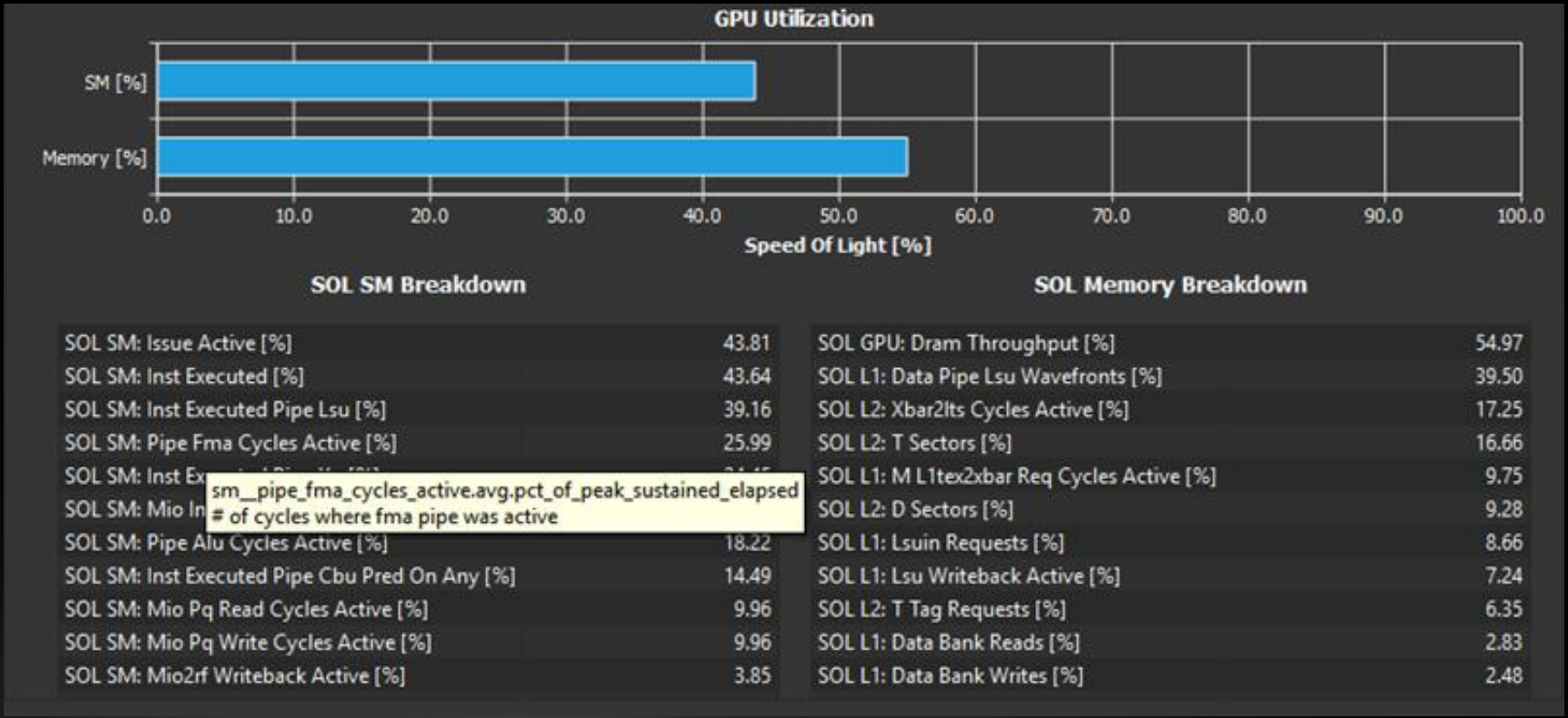
Profile activity command line is shown



NSIGHT COMPUTE

New Features

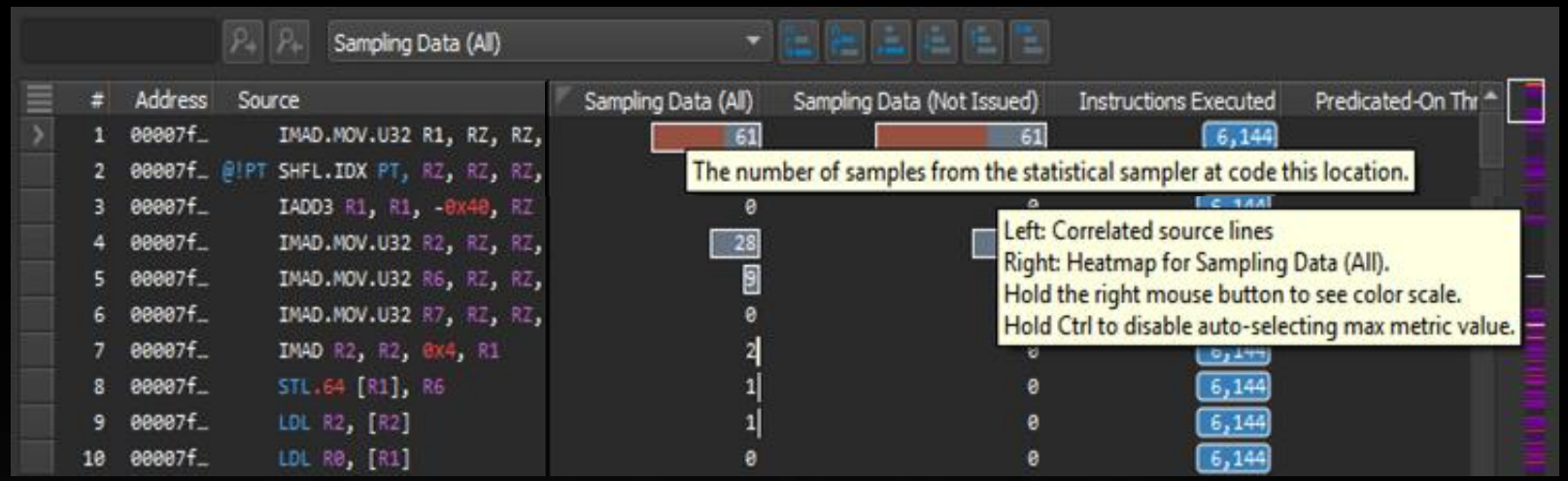
Detailed breakdown of high-level SOL metrics



NSIGHT COMPUTE

New Features

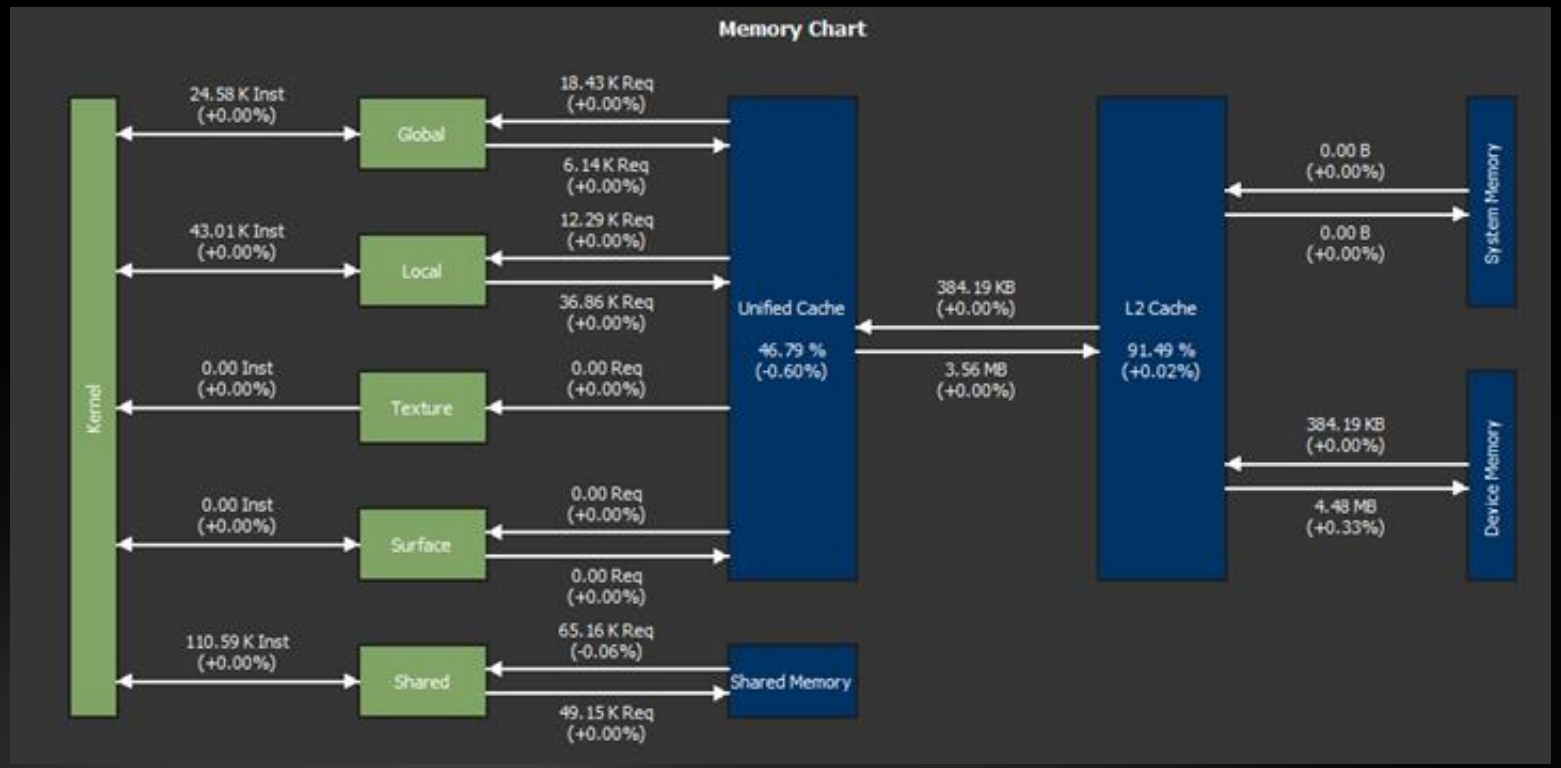
Improved help for Source page and Details page charts



NSIGHT COMPUTE

New Features

The Memory Workload Analysis chart now supports baselines



NSIGHT COMPUTE

New Features

Command line interface has new profiling and output controls

```
C:\Program Files\NVIDIA Corporation\Nsight Compute 2019.5.0>nv-nsight-cu-cli.bat --list-chips
gp102, gp104, gp106, gp107, gp108, gv100, gv11b, tu102, tu104, tu106, tu116, tu117
```

```
 --cache-control arg (=all)          Control the behavior of the GPU caches during profiling. Allowed values:
                                       all
                                       none
```

```
C:\Program Files\NVIDIA Corporation\Nsight Compute 2019.5.0\target\windows-desktop-win7-x64>nv-nsight-cu-cli.exe --kernel-base demangled
[314] python3.5@127.0.0.1
  void mxnet::op::mxnet_op::softmax_compute_kernel<7, mxnet::op::mxnet_op::softmax_fwd, false, float, 2, mshadow::half::half_t, mshadow:
::Shape<2>, mshadow::Shape<2>, double), 2019-Aug-07 07:25:25, Context 1, Stream 80
```

SUMMARY

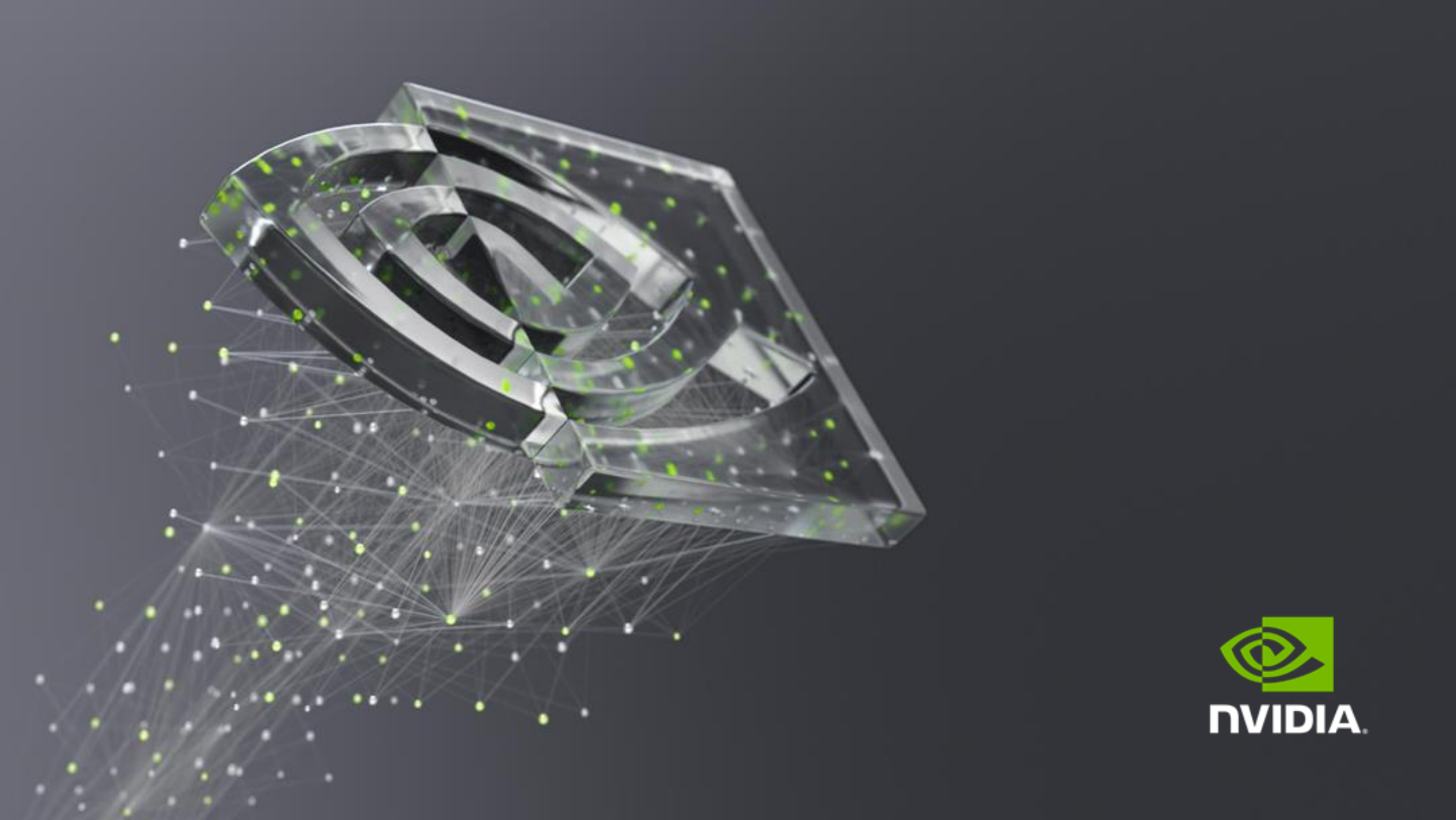
Full stack of Developer Tools available across platforms and APIs

Overall user consistent experience across many platforms

Compute & Graphics API support

Lower overhead (performance and footprint)

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


NVIDIA