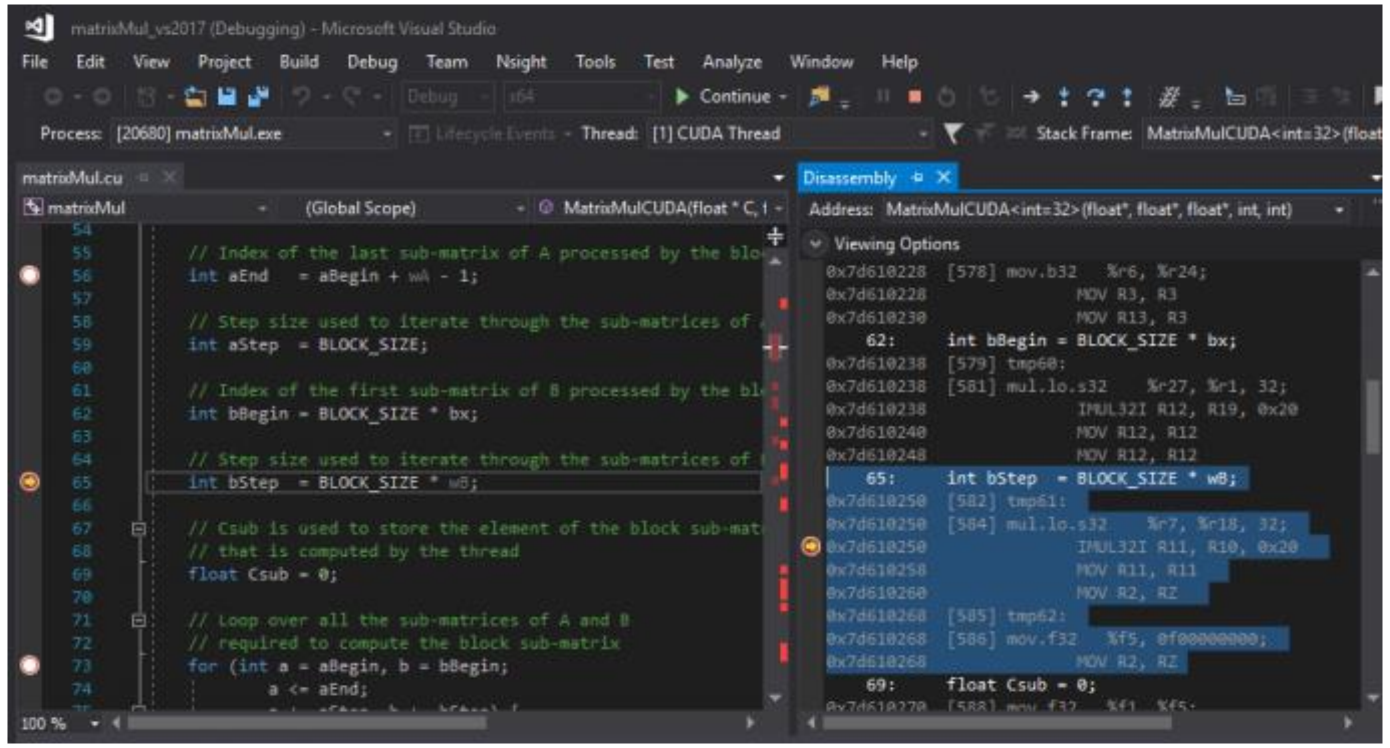
NSIGHT DEVELOPER TOOLS

DEVELOPER TOOLS

Debuggers: cuda-gdb, Nsight Eclipse Edition,
Nsight Visual Studio Edition, Nsight Visual Studio **Code** Edition

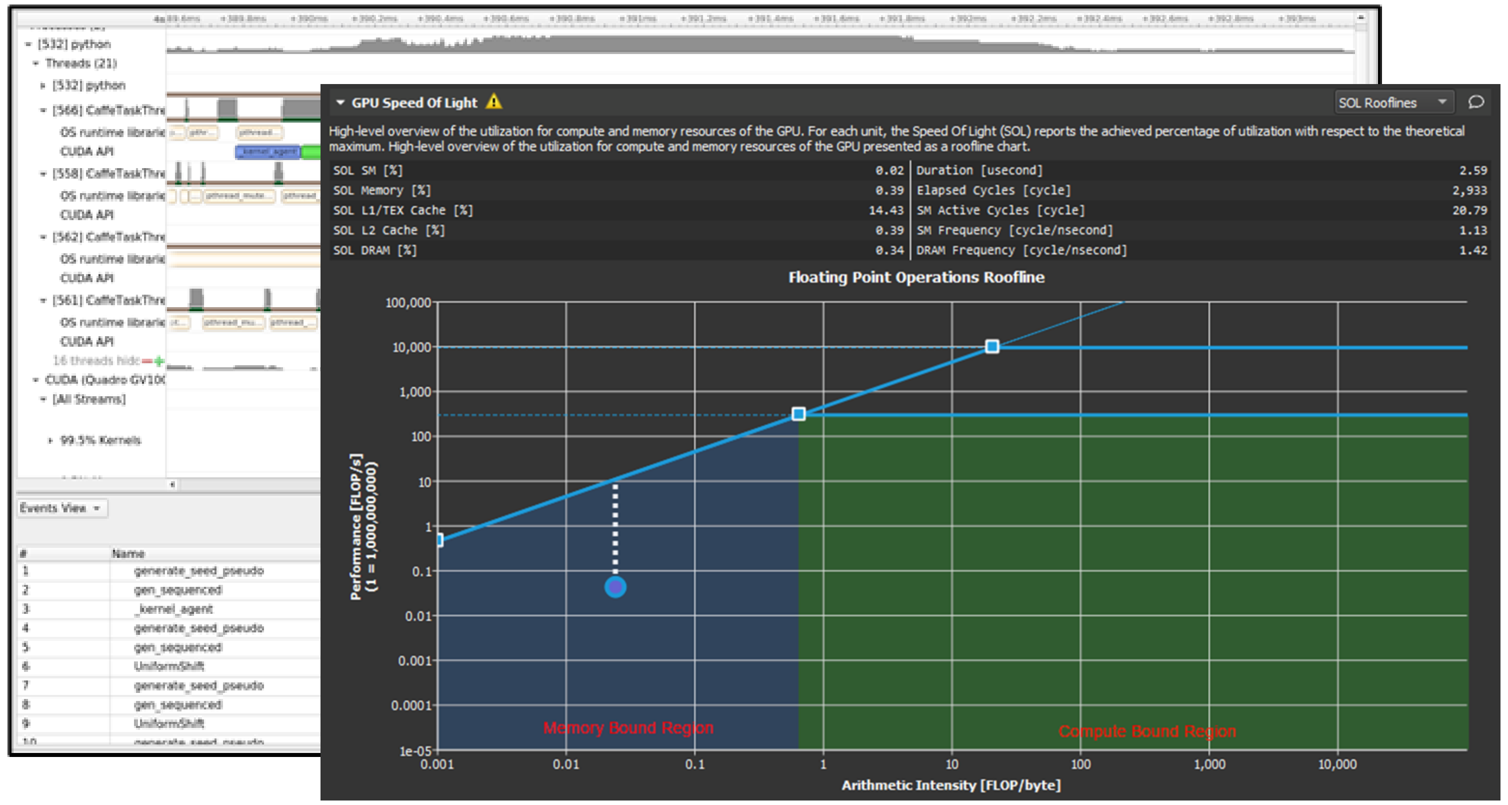


matrixMul_vs2017 (Debugging) - Microsoft Visual Studio
File Edit View Project Build Debug Team Nsight Tools Test Analyze Window Help
Continue
Process: [20680] matrixMul.exe
Lifecycle Events
Thread: [1] CUDA Thread
Stack Frame: MatrixMulCUDA<int=32>(float
matrixMul.cu
matrixMul
(Global Scope)
MatrixMulCUDA(float * C, f
Disassembly
Address: MatrixMulCUDA<int=32>(float*, float*, float*, int, int)
Viewing Options
100 %

```
// Index of the last sub-matrix of A processed by the blo
int aEnd   = aBegin + wA - 1;

// Step size used to iterate through the sub-matrices of
int aStep  = BLOCK_SIZE;

// Index of the first sub-matrix of B processed by the bl
int bBegin = BLOCK_SIZE * bx;

// Step size used to iterate through the sub-matrices of
int bStep  = BLOCK_SIZE * wB;

// Csub is used to store the element of the block sub-mat
// that is computed by the thread
float Csub = 0;

// Loop over all the sub-matrices of A and B
// required to compute the block sub-matrix
for (int a = aBegin, b = bBegin;
         a <= aEnd;
```

```
0x7d610228  [578] mov.b32   %r6, %r24;
0x7d610228                  MOV R3, R3
0x7d610230                  MOV R13, R3
     62:    int bBegin = BLOCK_SIZE * bx;
0x7d610238  [579] tmp60:
0x7d610238  [581] mul.lo.s32    %r27, %r1, 32;
0x7d610238                  IMUL32I R12, R19, 0x20
0x7d610240                  MOV R12, R12
0x7d610248                  MOV R12, R12
     65:    int bStep  = BLOCK_SIZE * wB;
0x7d610250  [582] tmp61:
0x7d610250  [584] mul.lo.s32    %r7, %r18, 32;
0x7d610250                  IMUL32I R11, R10, 0x20
0x7d610258                  MOV R11, R11
0x7d610260                  MOV R2, RZ
0x7d610268  [585] tmp62:
0x7d610268  [586] mov.f32   %f5, 0f00000000;
0x7d610268                  MOV R2, RZ
     69:    float Csub = 0;
0x7d610270  [588] mov.f32   %f1, %f5;
```

Profilers: Nsight Systems, Nsight Compute, CUPTI, NVIDIA Tools eXtension (NVTX)



#	Name
1	generate_seed_pseudo
2	gen_sequenced
3	_kernel_agent
4	generate_seed_pseudo
5	gen_sequenced
6	UniformShift
7	generate_seed_pseudo
8	gen_sequenced
9	UniformShift

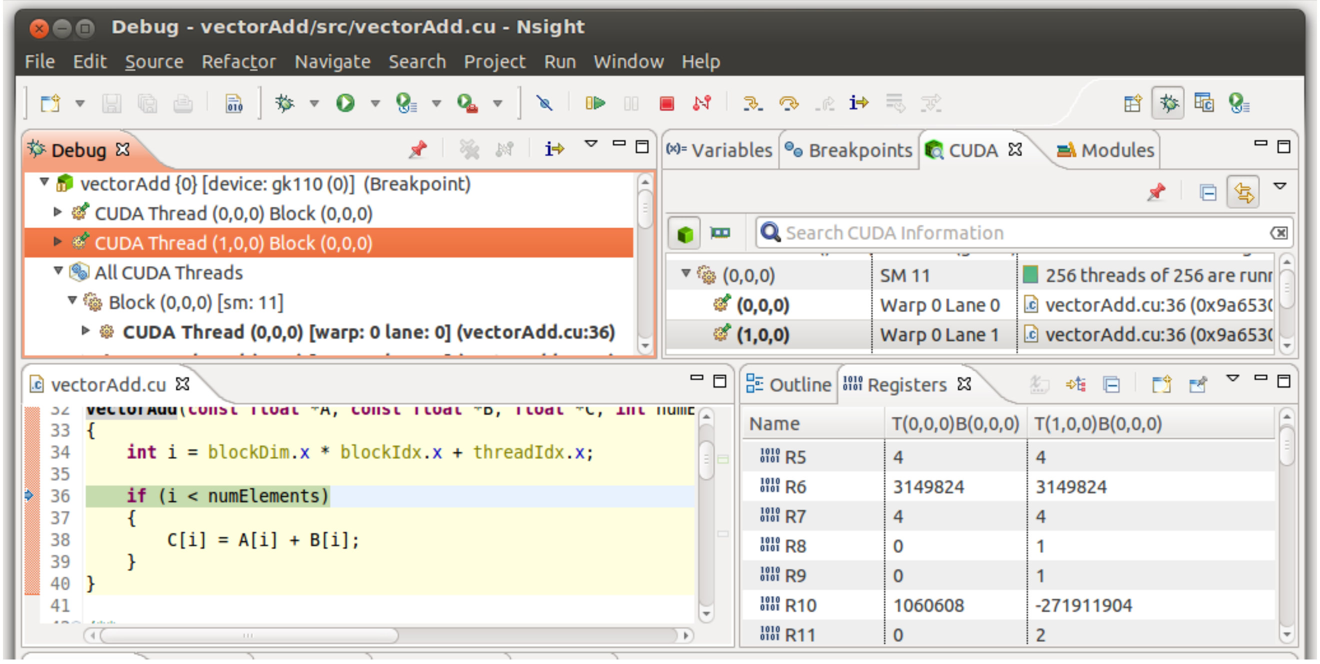
High-level overview of the utilization for compute and memory resources of the GPU. For each unit, the Speed Of Light (SOL) reports the achieved percentage of utilization with respect to the theoretical maximum. High-level overview of the utilization for compute and memory resources of the GPU presented as a roofline chart.

SOL SM [%]	0.02	Duration [usecond]	2.59
SOL Memory [%]	0.39	Elapsed Cycles [cycle]	2,933
SOL L1/TEX Cache [%]	14.43	SM Active Cycles [cycle]	20.79
SOL L2 Cache [%]	0.39	SM Frequency [cycle/nsecond]	1.13
SOL DRAM [%]	0.34	DRAM Frequency [cycle/nsecond]	1.42

Correctness Checker: Compute Sanitizer

```
$ compute-sanitizer --leak-check full memcheck_demo
========= COMPUTE-SANITIZER
Mallocing memory
Running unaligned_kernel
Ran unaligned_kernel: no error
Sync: no error
Running out_of_bounds_kernel
Ran out_of_bounds_kernel: no error
Sync: no error
========= Invalid __global__ write of size 4 bytes
=========     at 0x60 in memcheck_demo.cu:6:unaligned_kernel(void)
=========     by thread (0,0,0) in block (0,0,0)
=========     Address 0x400100001 is misaligned
```

IDE integrations: Nsight Eclipse Edition
Nsight Visual Studio Edition
Nsight Visual Studio **Code** Edition



Debug - vectorAdd/src/vectorAdd.cu - Nsight
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Variables
Breakpoints
CUDA
Modules
Search CUDA Information
vectorAdd.cu
Outline
Registers

- vectorAdd {0} [device: gk110 (0)] (Breakpoint)
  - CUDA Thread (0,0,0) Block (0,0,0)
  - CUDA Thread (1,0,0) Block (0,0,0)
  - All CUDA Threads
    - Block (0,0,0) [sm: 11]
      - CUDA Thread (0,0,0) [warp: 0 lane: 0] (vectorAdd.cu:36)

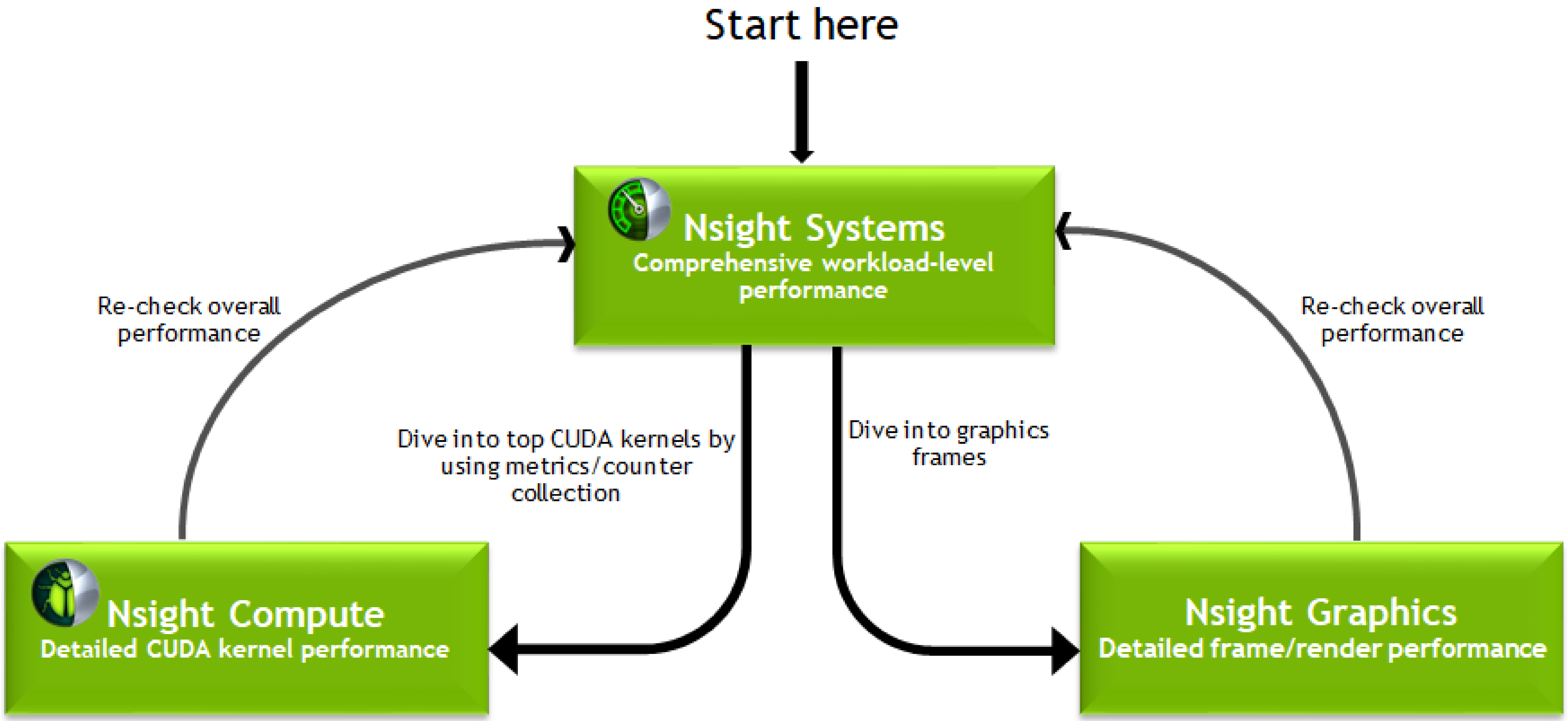
(0,0,0)	SM 11	256 threads of 256 are runr
(0,0,0)	Warp 0 Lane 0	vectorAdd.cu:36 (0x9a653(
(1,0,0)	Warp 0 Lane 1	vectorAdd.cu:36 (0x9a653(

```
vectorAdd(const float *A, const float *B, float *C, int nume
{
    int i = blockDim.x * blockIdx.x + threadIdx.x;

    if (i < numElements)
    {
        C[i] = A[i] + B[i];
    }
}
```

Name	T(0,0,0)B(0,0,0)	T(1,0,0)B(0,0,0)
R5	4	4
R6	3149824	3149824
R7	4	4
R8	0	1
R9	0	1
R10	1060608	-271911904
R11	0	2

NSIGHT PROFILING TOOLS WORKFLOW

ITERATIVE OPTIMIZATION FOR COMPUTE AND GRAPHICS



Start here

Nsight Systems
Comprehensive workload-level performance

Re-check overall performance

Dive into top CUDA kernels by using metrics/counter collection

Dive into graphics frames

Re-check overall performance

Nsight Compute
Detailed CUDA kernel performance

Nsight Graphics
Detailed frame/render performance

NSIGHT SYSTEMS

SYSTEM PROFILER

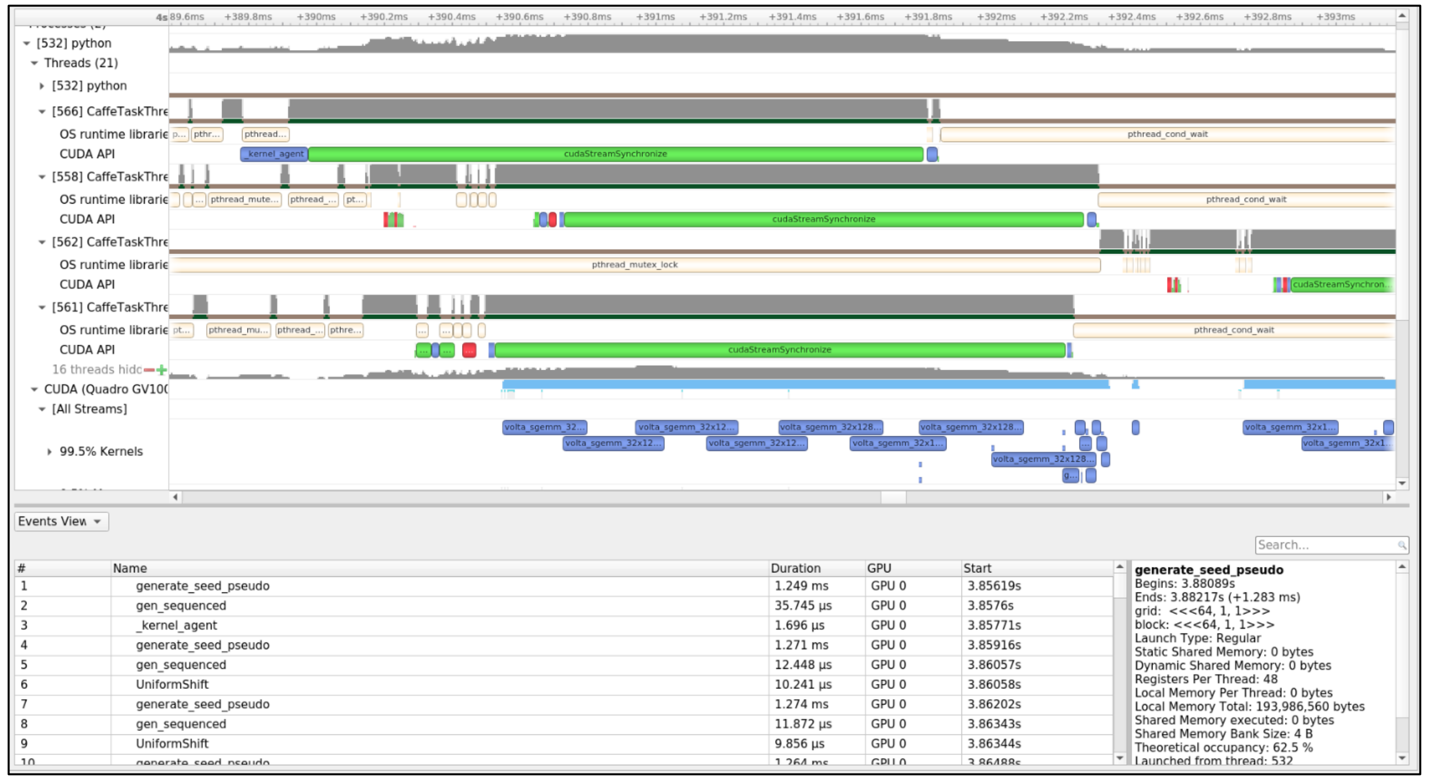
Key Features:

- 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)

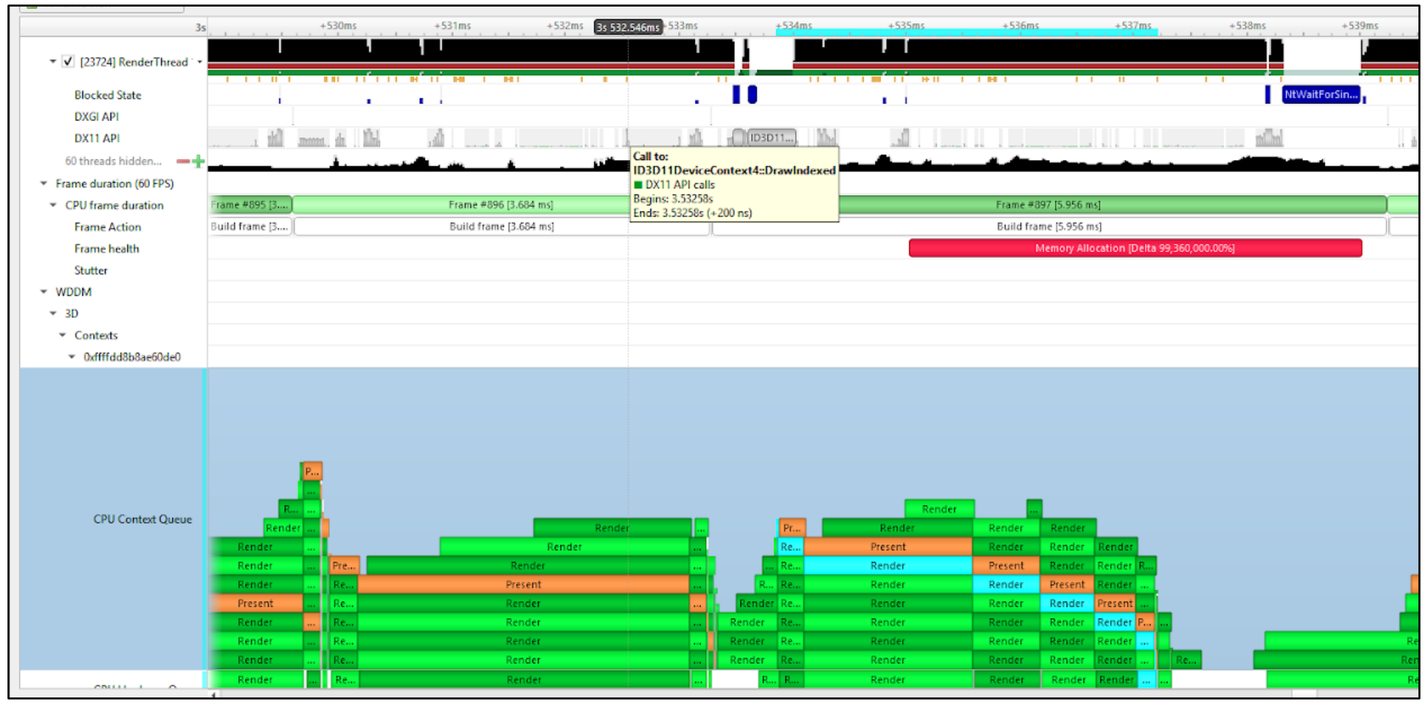
GPUs: Pascal+

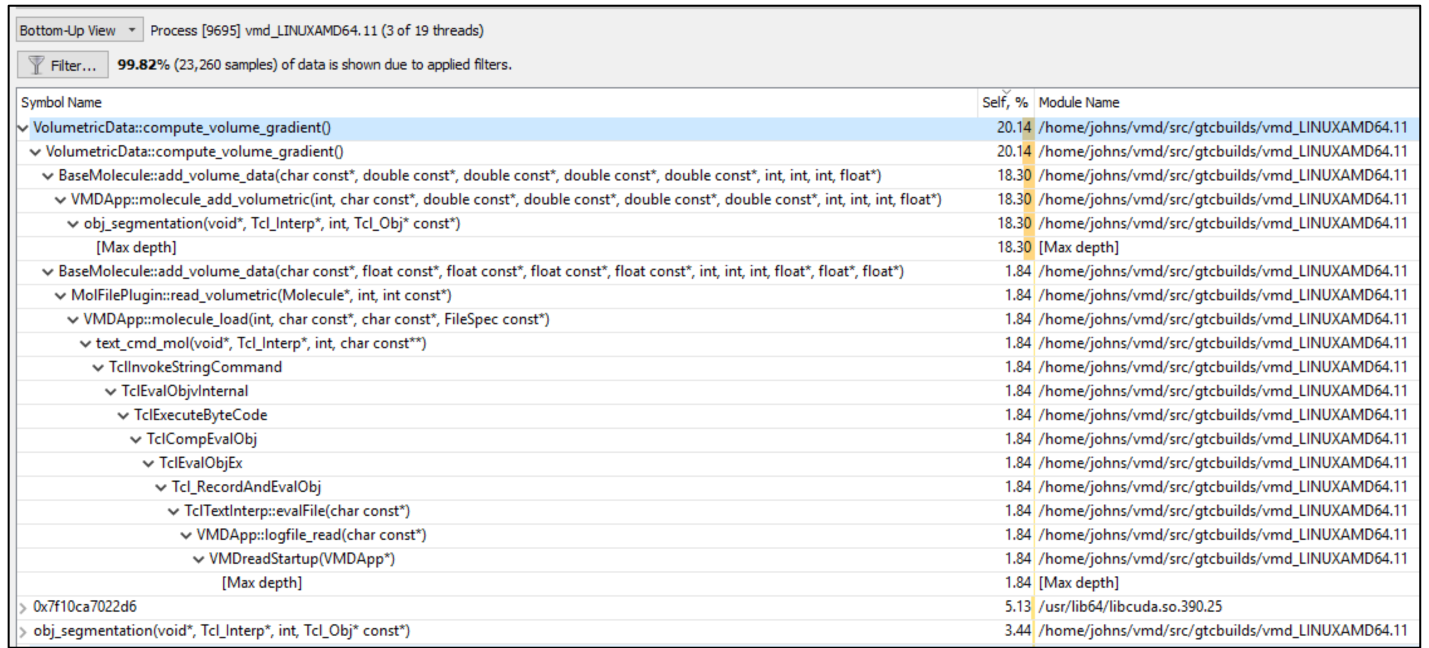
Docs/product: https://developer.nvidia.com/nsight-systems



#	Name	Duration	GPU	Start
1	generate_seed_pseudo	1.249 ms	GPU 0	3.85619s
2	gen_sequenced	35.745 µs	GPU 0	3.8576s
3	_kernel_agent	1.696 µs	GPU 0	3.85771s
4	generate_seed_pseudo	1.271 ms	GPU 0	3.85916s
5	gen_sequenced	12.448 µs	GPU 0	3.86057s
6	UniformShift	10.241 µs	GPU 0	3.86058s
7	generate_seed_pseudo	1.274 ms	GPU 0	3.86202s
8	gen_sequenced	11.872 µs	GPU 0	3.86343s
9	UniformShift	9.856 µs	GPU 0	3.86344s
10	generate_seed_pseudo	1.264 ms	GPU 0	3.86488s

generate_seed_pseudo
Begins: 3.88089s
Ends: 3.88217s (+1.283 ms)
grid: <<<64, 1, 1>>>
block: <<<64, 1, 1>>>
Launch Type: Regular
Static Shared Memory: 0 bytes
Dynamic Shared Memory: 0 bytes
Registers Per Thread: 48
Local Memory Per Thread: 0 bytes
Local Memory Total: 193,986,560 bytes
Shared Memory executed: 0 bytes
Shared Memory Bank Size: 4 B
Theoretical occupancy: 62.5 %
Launched from thread: 532

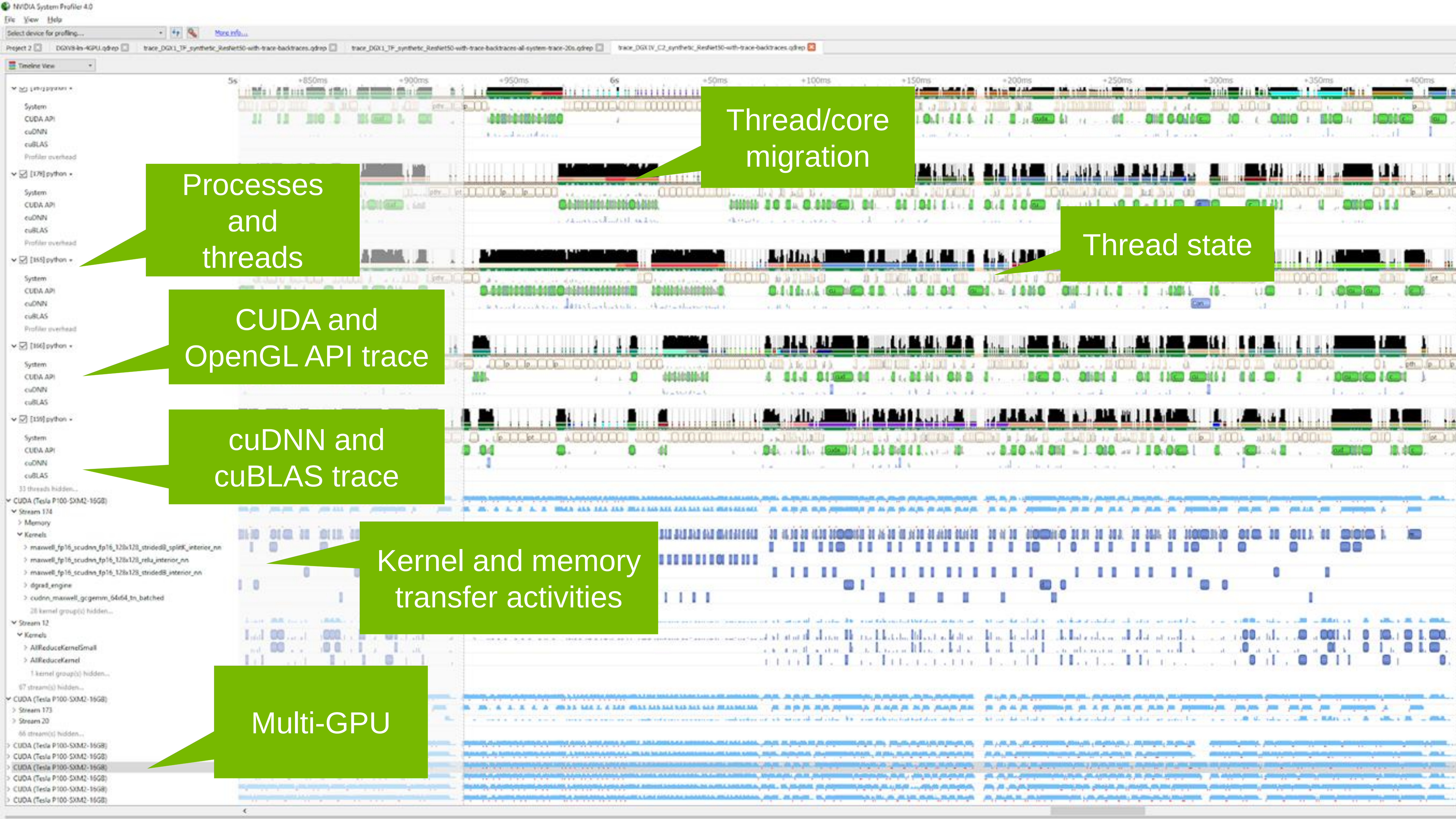




Bottom-Up View ▾ Process [9695] vmd_LINUXAMD64.11 (3 of 19 threads)

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


NVIDIA System Profiler 4.0
File View Help
Select device for profiling...
More info...
Project 2
DGX8-lin-4GPU.qdrep
trace_DGX1_TF_synthetic_ResNet50-with-trace-backtraces.qdrep
trace_DGX1_TF_synthetic_ResNet50-with-trace-backtraces-all-system-trace-20s.qdrep
trace_DGX1V_C2_synthetic_ResNet50-with-trace-backtraces.qdrep
Timeline View
5s
+850ms
+900ms
+950ms
6s
+50ms
+100ms
+150ms
+200ms
+250ms
+300ms
+350ms
+400ms
System
CUDA API
cuDNN
cuBLAS
Profiler overhead
[178] python
[165] python
[166] python
[159] python
33 threads hidden...
CUDA (Tesla P100-SXM2-16GB)
Stream 174
Memory
Kernels
maxwell_fp16_scudnn_fp16_128x128_stridedB_splitK_interior_nn
maxwell_fp16_scudnn_fp16_128x128_relu_interior_nn
maxwell_fp16_scudnn_fp16_128x128_stridedB_interior_nn
dgrad_engine
cudnn_maxwell_gcgemm_64x64_tn_batched
28 kernel group(s) hidden...
Stream 12
Kernels
AllReduceKernelSmall
AllReduceKernel
1 kernel group(s) hidden...
67 stream(s) hidden...
CUDA (Tesla P100-SXM2-16GB)
Stream 173
Stream 20
66 stream(s) hidden...
CUDA (Tesla P100-SXM2-16GB)
CUDA (Tesla P100-SXM2-16GB)
CUDA (Tesla P100-SXM2-16GB)
CUDA (Tesla P100-SXM2-16GB)
CUDA (Tesla P100-SXM2-16GB)
CUDA (Tesla P100-SXM2-16GB)

Thread/core migration

Processes and threads

Thread state

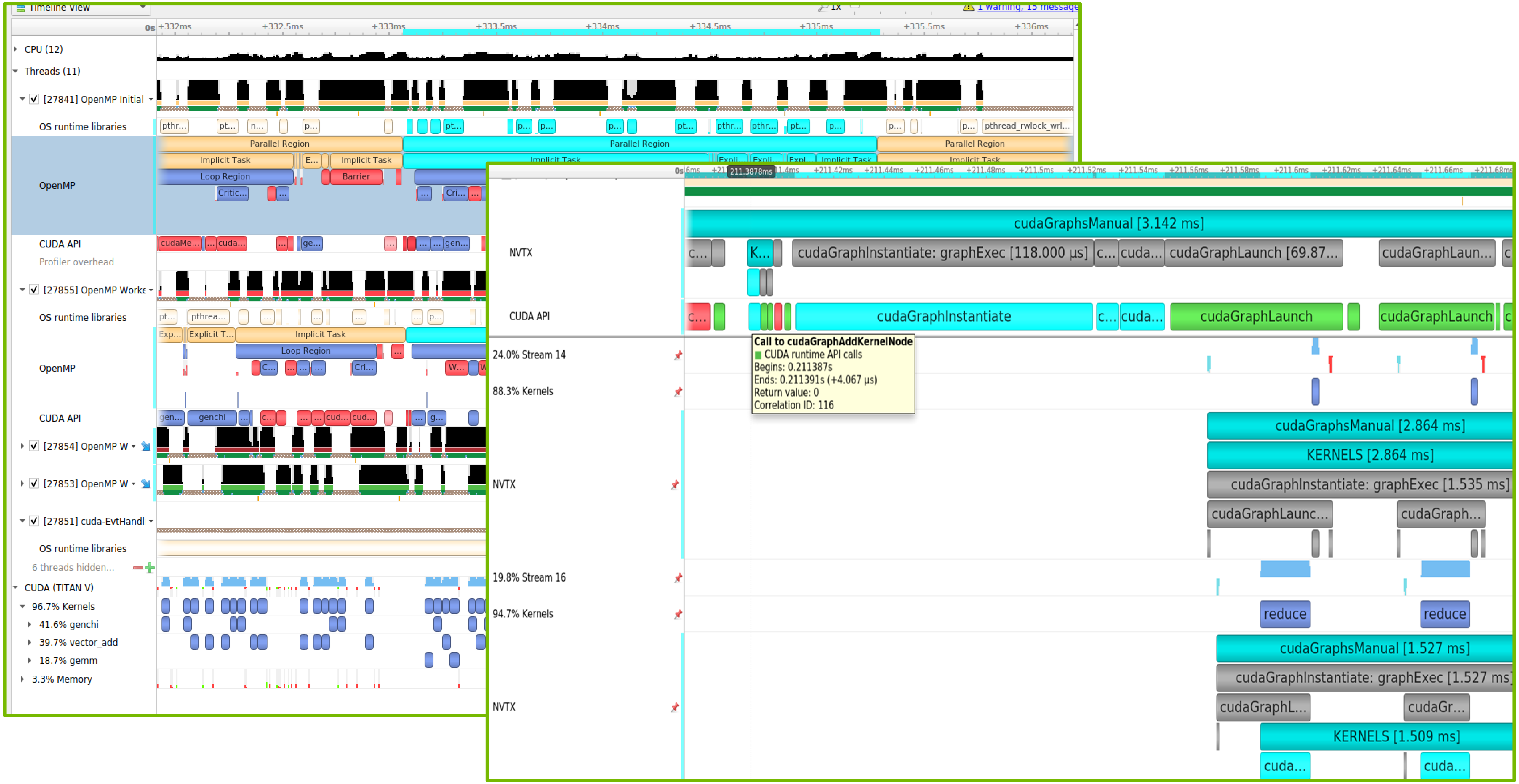
CUDA and OpenGL API trace

cuDNN and cuBLAS trace

Kernel and memory transfer activities

Multi-GPU

ZOOM/FILTER TO EXACT AREAS OF INTEREST



Call to cudaGraphAddKernelNode
🟩 CUDA runtime API calls
Begins: 0.211387s
Ends: 0.211391s (+4.067 μs)
Return value: 0
Correlation ID: 116

COLLECT A PROFILE WITH NSIGHT SYSTEMS

```
$ nsys profile –o report --stats=true ./myapp.exe
```

Generated file: `report.qdrep` (or `report.nsys-rep);` open for viewing in the Nsight Systems UI

When using MPI, I recommend using nsys after mpirun/srun/jsrun/etc.:

```
$ mpirun –n 4 nsys profile ./myapp.exe
```

NSIGHT COMPUTE

KERNEL PROFILING TOOL

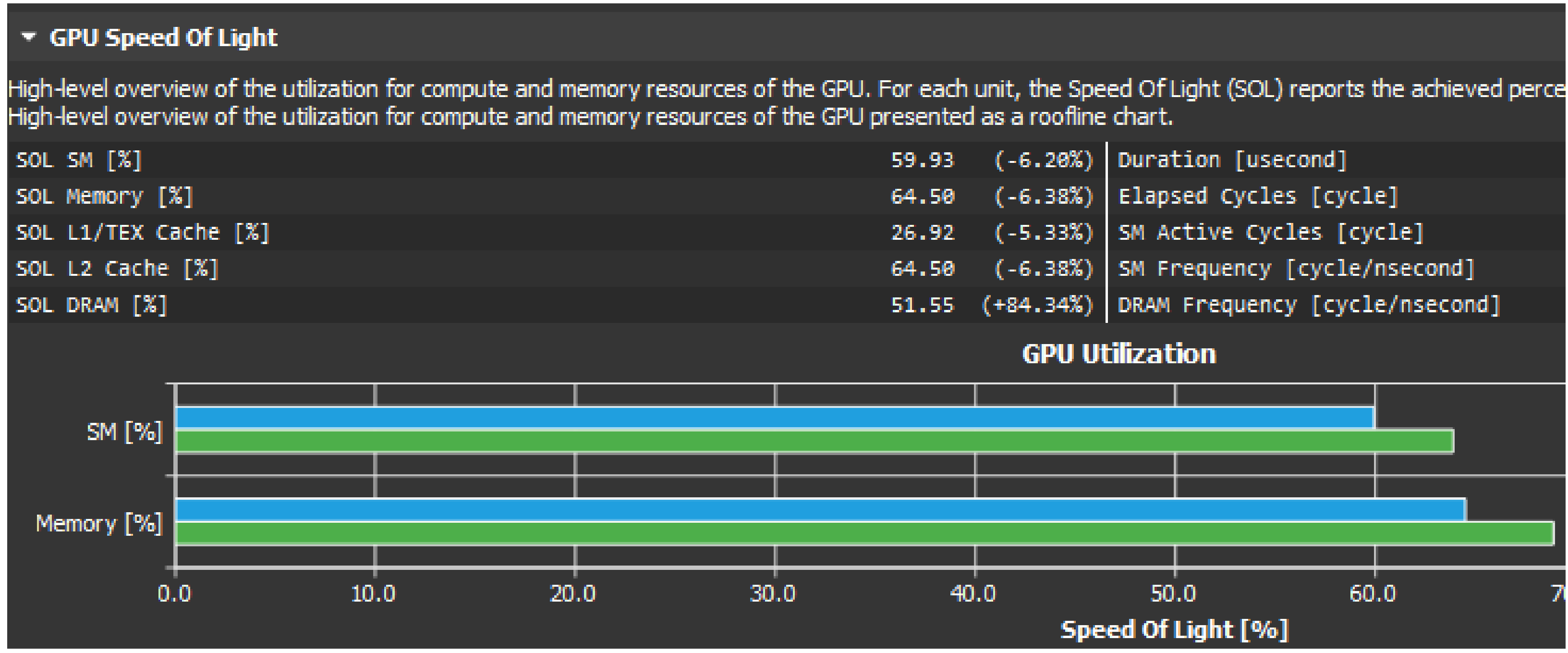
Key Features:

- 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, Turing, Ampere GPUs

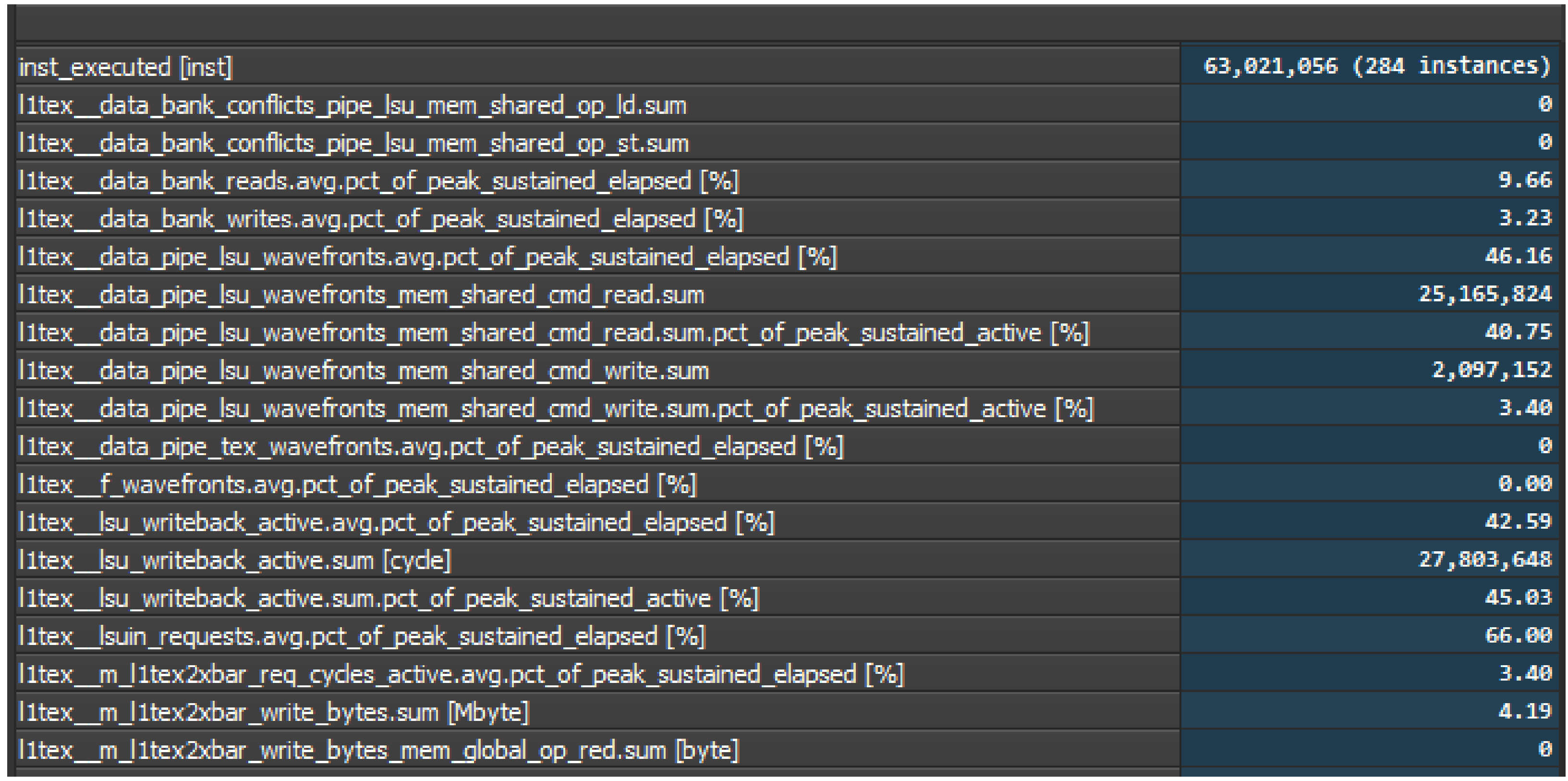
Docs/product: https://developer.nvidia.com/nsight-compute



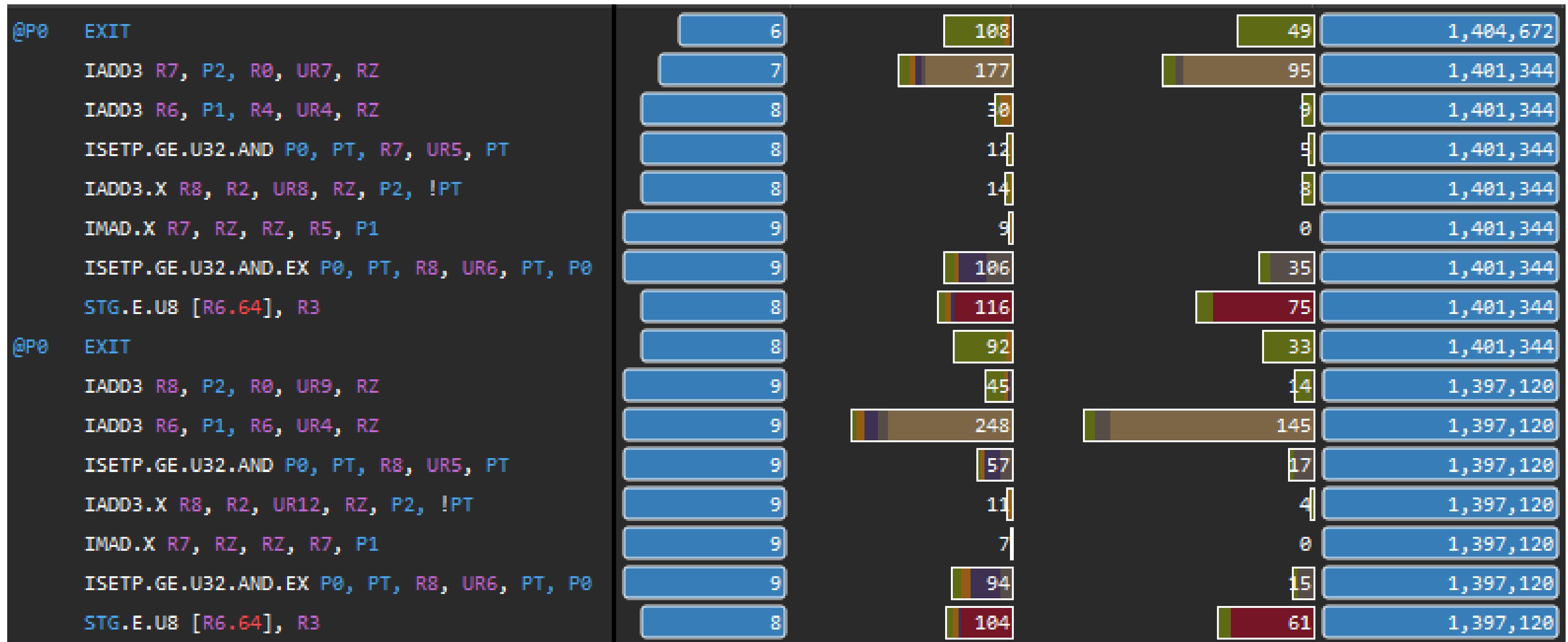
▾ GPU Speed Of Light

High-level overview of the utilization for compute and memory resources of the GPU. For each unit, the Speed Of Light (SOL) reports the achieved perce
High-level overview of the utilization for compute and memory resources of the GPU presented as a roofline chart.

```
SOL SM [%]                59.93   (-6.20%)  Duration [usecond]
SOL Memory [%]            64.50   (-6.38%)  Elapsed Cycles [cycle]
SOL L1/TEX Cache [%]      26.92   (-5.33%)  SM Active Cycles [cycle]
SOL L2 Cache [%]          64.50   (-6.38%)  SM Frequency [cycle/nsecond]
SOL DRAM [%]              51.55  (+84.34%)  DRAM Frequency [cycle/nsecond]
```



```
inst_executed [inst]                                                          63,021,056 (284 instances)
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_ld.sum                                               0
l1tex__data_bank_conflicts_pipe_lsu_mem_shared_op_st.sum                                               0
l1tex__data_bank_reads.avg.pct_of_peak_sustained_elapsed [%]                                        9.66
l1tex__data_bank_writes.avg.pct_of_peak_sustained_elapsed [%]                                       3.23
l1tex__data_pipe_lsu_wavefronts.avg.pct_of_peak_sustained_elapsed [%]                              46.16
l1tex__data_pipe_lsu_wavefronts_mem_shared_cmd_read.sum                                       25,165,824
l1tex__data_pipe_lsu_wavefronts_mem_shared_cmd_read.sum.pct_of_peak_sustained_active [%]           40.75
l1tex__data_pipe_lsu_wavefronts_mem_shared_cmd_write.sum                                       2,097,152
l1tex__data_pipe_lsu_wavefronts_mem_shared_cmd_write.sum.pct_of_peak_sustained_active [%]           3.40
l1tex__data_pipe_tex_wavefronts.avg.pct_of_peak_sustained_elapsed [%]                                  0
l1tex__f_wavefronts.avg.pct_of_peak_sustained_elapsed [%]                                           0.00
l1tex__lsu_writeback_active.avg.pct_of_peak_sustained_elapsed [%]                                  42.59
l1tex__lsu_writeback_active.sum [cycle]                                                       27,803,648
l1tex__lsu_writeback_active.sum.pct_of_peak_sustained_active [%]                                   45.03
l1tex__lsuin_requests.avg.pct_of_peak_sustained_elapsed [%]                                        66.00
l1tex__m_l1tex2xbar_req_cycles_active.avg.pct_of_peak_sustained_elapsed [%]                         3.40
l1tex__m_l1tex2xbar_write_bytes.sum [Mbyte]                                                         4.19
l1tex__m_l1tex2xbar_write_bytes_mem_global_op_red.sum [byte]                                           0
```

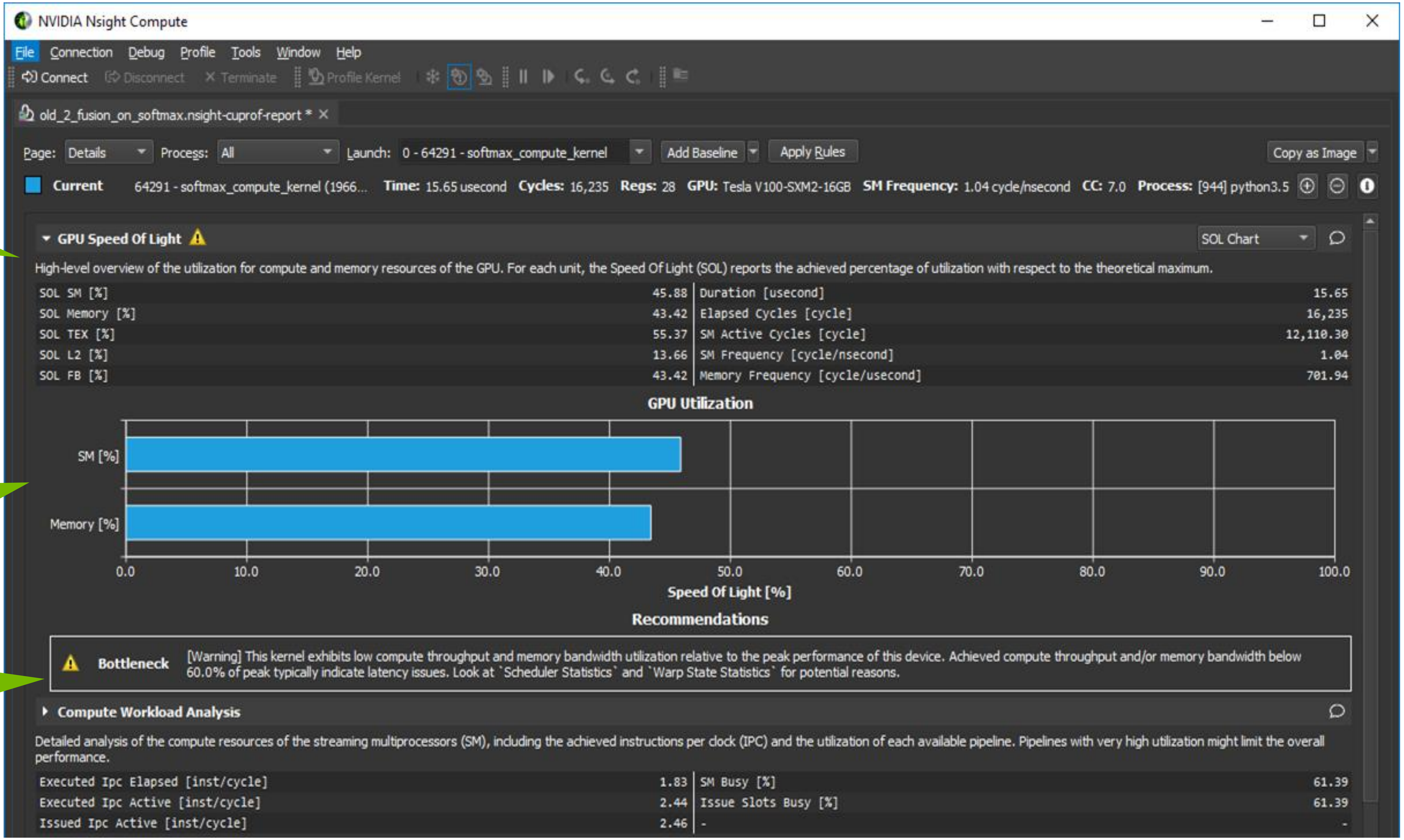


```
@P0   EXIT                                       6    108     49   1,404,672
      IADD3 R7, P2, R0, UR7, RZ                  7    177     95   1,401,344
      IADD3 R6, P1, R4, UR4, RZ                  8     30      9   1,401,344
      ISETP.GE.U32.AND P0, PT, R7, UR5, PT       8     12      5   1,401,344
      IADD3.X R8, R2, UR8, RZ, P2, !PT           8     14      8   1,401,344
      IMAD.X R7, RZ, RZ, R5, P1                  9      9      0   1,401,344
      ISETP.GE.U32.AND.EX P0, PT, R8, UR6, PT, P0  9  106     35   1,401,344
      STG.E.U8 [R6.64], R3                       8    116     75   1,401,344
@P0   EXIT                                       8     92     33   1,401,344
      IADD3 R8, P2, R0, UR9, RZ                  9     45     14   1,397,120
      IADD3 R6, P1, R6, UR4, RZ                  9    248    145   1,397,120
      ISETP.GE.U32.AND P0, PT, R8, UR5, PT       9     57     17   1,397,120
      IADD3.X R8, R2, UR12, RZ, P2, !PT          9     11      4   1,397,120
      IMAD.X R7, RZ, RZ, R7, P1                  9      7      0   1,397,120
      ISETP.GE.U32.AND.EX P0, PT, R8, UR6, PT, P0  9   94     15   1,397,120
      STG.E.U8 [R6.64], R3                       8    104     61   1,397,120
```

Targeted metric sections

Customizable data collection and presentation

Built-in expertise for Guided Analysis and optimization



Page: Details Process: All Launch: 0 - 64291 - softmax_compute_kernel

Current 64291 - softmax_compute_kernel (1966... **Time:** 15.65 usecond **Cycles:** 16,235 **Regs:** 28 **GPU:** Tesla V100-SXM2-16GB **SM Frequency:** 1.04 cycle/nsecond **CC:** 7.0 **Process:** [944] python3.5

GPU Speed Of Light ⚠

High-level overview of the utilization for compute and memory resources of the GPU. For each unit, the Speed Of Light (SOL) reports the achieved percentage of utilization with respect to the theoretical maximum.

SOL SM [%]	45.88	Duration [usecond]	15.65
SOL Memory [%]	43.42	Elapsed Cycles [cycle]	16,235
SOL TEX [%]	55.37	SM Active Cycles [cycle]	12,110.30
SOL L2 [%]	13.66	SM Frequency [cycle/nsecond]	1.04
SOL FB [%]	43.42	Memory Frequency [cycle/usecond]	701.94

Recommendations

⚠ **Bottleneck** [Warning] This kernel exhibits low compute throughput and memory bandwidth utilization relative to the peak performance of this device. Achieved compute throughput and/or memory bandwidth below 60.0% of peak typically indicate latency issues. Look at `Scheduler Statistics` and `Warp State Statistics` for potential reasons.

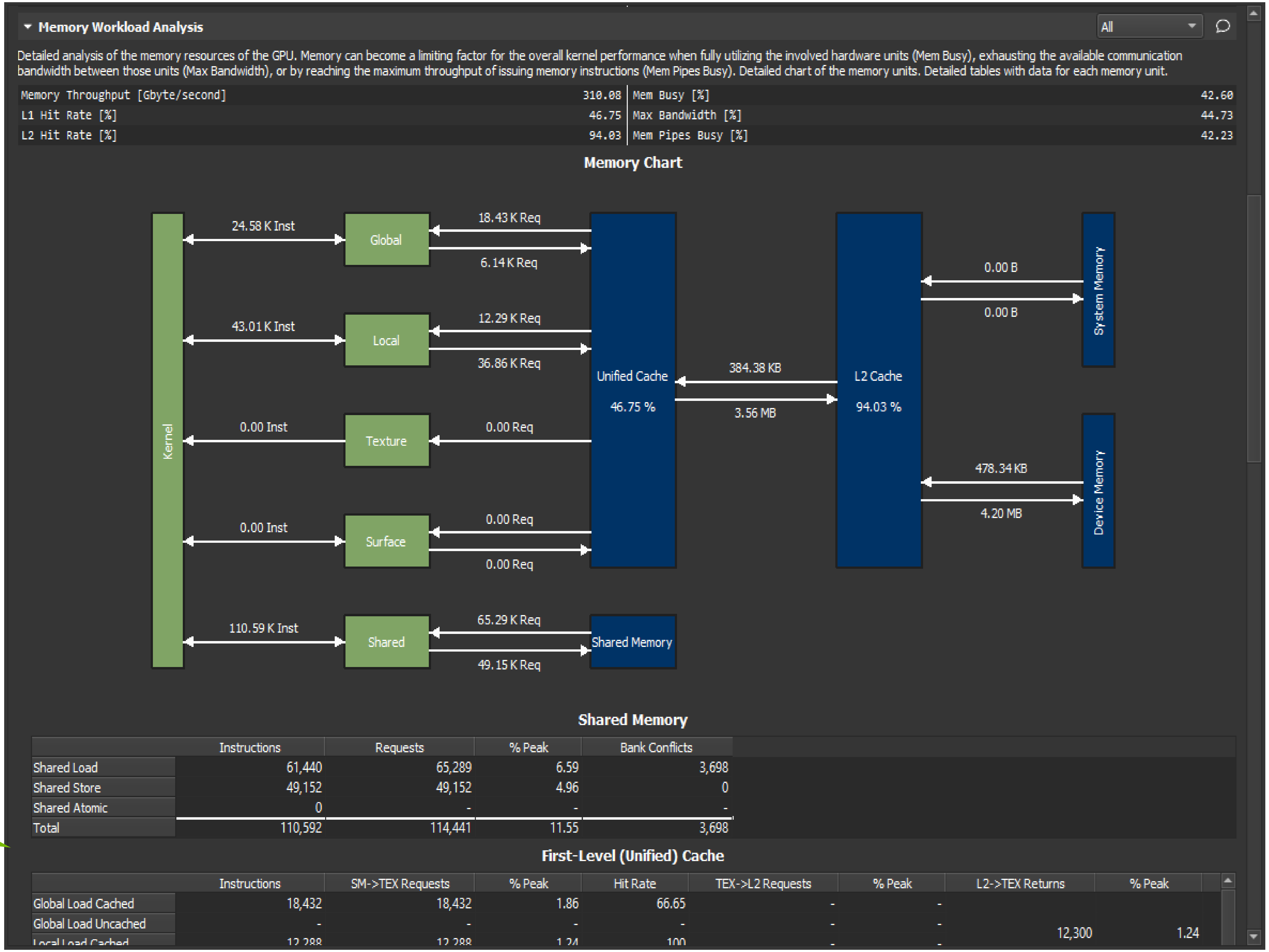
Compute Workload Analysis

Detailed analysis of the compute resources of the streaming multiprocessors (SM), including the achieved instructions per clock (IPC) and the utilization of each available pipeline. Pipelines with very high utilization might limit the overall performance.

Executed Ipc Elapsed [inst/cycle]	1.83	SM Busy [%]	61.39
Executed Ipc Active [inst/cycle]	2.44	Issue Slots Busy [%]	61.39
Issued Ipc Active [inst/cycle]	2.46	-	-

Visual memory analysis chart

Metrics for peak performance ratios



▾ Memory Workload Analysis

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]	310.08	Mem Busy [%]	42.60
L1 Hit Rate [%]	46.75	Max Bandwidth [%]	44.73
L2 Hit Rate [%]	94.03	Mem Pipes Busy [%]	42.23

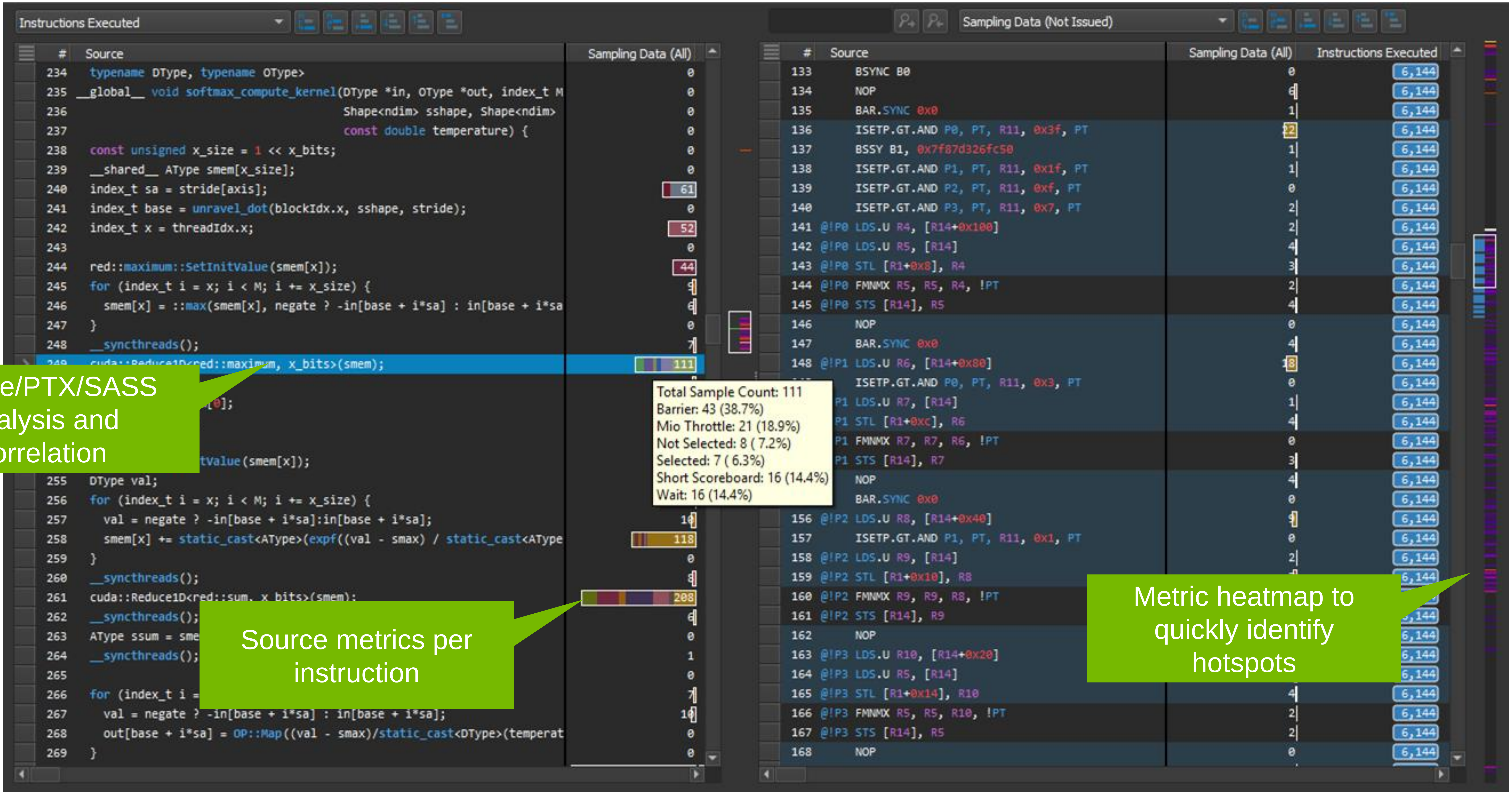
Memory Chart

Shared Memory

	Instructions	Requests	% Peak	Bank Conflicts
Shared Load	61,440	65,289	6.59	3,698
Shared Store	49,152	49,152	4.96	0
Shared Atomic	0	-	-	-
Total	110,592	114,441	11.55	3,698

First-Level (Unified) Cache

	Instructions	SM->TEX Requests	% Peak	Hit Rate	TEX->L2 Requests	% Peak	L2->TEX Returns	% Peak
Global Load Cached	18,432	18,432	1.86	66.65	-	-		
Global Load Uncached	-	-	-	-	-	-	12,300	1.24
Local Load Cached	12,288	12,288	1.24	100	-	-		



```
#    Source                                                              Sampling Data (All)
234   typename DType, typename OType>                                      0
235 __global__ void softmax_compute_kernel(DType *in, OType *out, index_t M  0
236                                         Shape<ndim> sshape, Shape<ndim>  0
237                                         const double temperature) {      0
238   const unsigned x_size = 1 << x_bits;                                 0
239   __shared__ AType smem[x_size];                                       0
240   index_t sa = stride[axis];                                          61
241   index_t base = unravel_dot(blockIdx.x, sshape, stride);              0
242   index_t x = threadIdx.x;                                            52
243                                                                        0
244   red::maximum::SetInitValue(smem[x]);                                44
245   for (index_t i = x; i < M; i += x_size) {                            5
246     smem[x] = ::max(smem[x], negate ? -in[base + i*sa] : in[base + i*sa  6
247   }                                                                    0
248   __syncthreads();                                                     7
249   cuda::Reduce1D<red::maximum, x_bits>(smem);                        111
                 [0];
                 tValue(smem[x]);
255   DType val;
256   for (index_t i = x; i < M; i += x_size) {
257     val = negate ? -in[base + i*sa]:in[base + i*sa];                  10
258     smem[x] += static_cast<AType>(expf((val - smax) / static_cast<AType  118
259   }                                                                    0
260   __syncthreads();                                                     8
261   cuda::Reduce1D<red::sum, x_bits>(smem);                            208
262   __syncthreads();                                                     6
263   AType ssum = sme                                                     0
264   __syncthreads();                                                     1
265                                                                        0
266   for (index_t i =                                                     7
267     val = negate ? -in[base + i*sa] : in[base + i*sa];                10
268     out[base + i*sa] = OP::Map((val - smax)/static_cast<DType>(temperat  0
269   }                                                                    0
```

```
Total Sample Count: 111
Barrier: 43 (38.7%)
Mio Throttle: 21 (18.9%)
Not Selected: 8 ( 7.2%)
Selected: 7 ( 6.3%)
Short Scoreboard: 16 (14.4%)
Wait: 16 (14.4%)
```

```
#    Source                                  Sampling Data (All)  Instructions Executed
133      BSYNC B0                                          0      6,144
134      NOP                                               6      6,144
135      BAR.SYNC 0x0                                      1      6,144
136      ISETP.GT.AND P0, PT, R11, 0x3f, PT               22      6,144
137      BSSY B1, 0x7f87d326fc50                           1      6,144
138      ISETP.GT.AND P1, PT, R11, 0x1f, PT                1      6,144
139      ISETP.GT.AND P2, PT, R11, 0xf, PT                 0      6,144
140      ISETP.GT.AND P3, PT, R11, 0x7, PT                 2      6,144
141 @!P0 LDS.U R4, [R14+0x100]                             2      6,144
142 @!P0 LDS.U R5, [R14]                                   4      6,144
143 @!P0 STL [R1+0x8], R4                                  3      6,144
144 @!P0 FMNMX R5, R5, R4, !PT                             2      6,144
145 @!P0 STS [R14], R5                                     4      6,144
146      NOP                                               0      6,144
147      BAR.SYNC 0x0                                      4      6,144
148 @!P1 LDS.U R6, [R14+0x80]                             18      6,144
         ISETP.GT.AND P0, PT, R11, 0x3, PT                 0      6,144
      P1 LDS.U R7, [R14]                                   1      6,144
      P1 STL [R1+0xc], R6                                  4      6,144
      P1 FMNMX R7, R7, R6, !PT                             0      6,144
      P1 STS [R14], R7                                     3      6,144
         NOP                                               4      6,144
         BAR.SYNC 0x0                                      0      6,144
156 @!P2 LDS.U R8, [R14+0x40]                              9      6,144
157      ISETP.GT.AND P1, PT, R11, 0x1, PT                 0      6,144
158 @!P2 LDS.U R9, [R14]                                   2      6,144
159 @!P2 STL [R1+0x10], R8                                        6,144
160 @!P2 FMNMX R9, R9, R8, !PT
161 @!P2 STS [R14], R9                                              ,144
162      NOP                                                       6,144
163 @!P3 LDS.U R10, [R14+0x20]                                     6,144
164 @!P3 LDS.U R5, [R14]                                           6,144
165 @!P3 STL [R1+0x14], R10                                4      6,144
166 @!P3 FMNMX R5, R5, R10, !PT                            2      6,144
167 @!P3 STS [R14], R5                                     2      6,144
168      NOP                                               0      6,144
```

Source/PTX/SASS analysis and correlation

Source metrics per instruction

Metric heatmap to quickly identify hotspots

KERNEL PROFILES WITH NSIGHT COMPUTE

```
$ ncu –k mykernel –o report ./myapp.exe
```

Generated file: `report.ncu-rep;` open for viewing in the Nsight Compute UI

(Without the –k option, Nsight Compute will profile everything and take a long time!)

PROFILING RESOURCES

Nsight Systems, Nsight Compute product pages

NVIDIA Developer Blog: NVTX

NVIDIA Developer Blog: Transitioning from nvprof to nsys

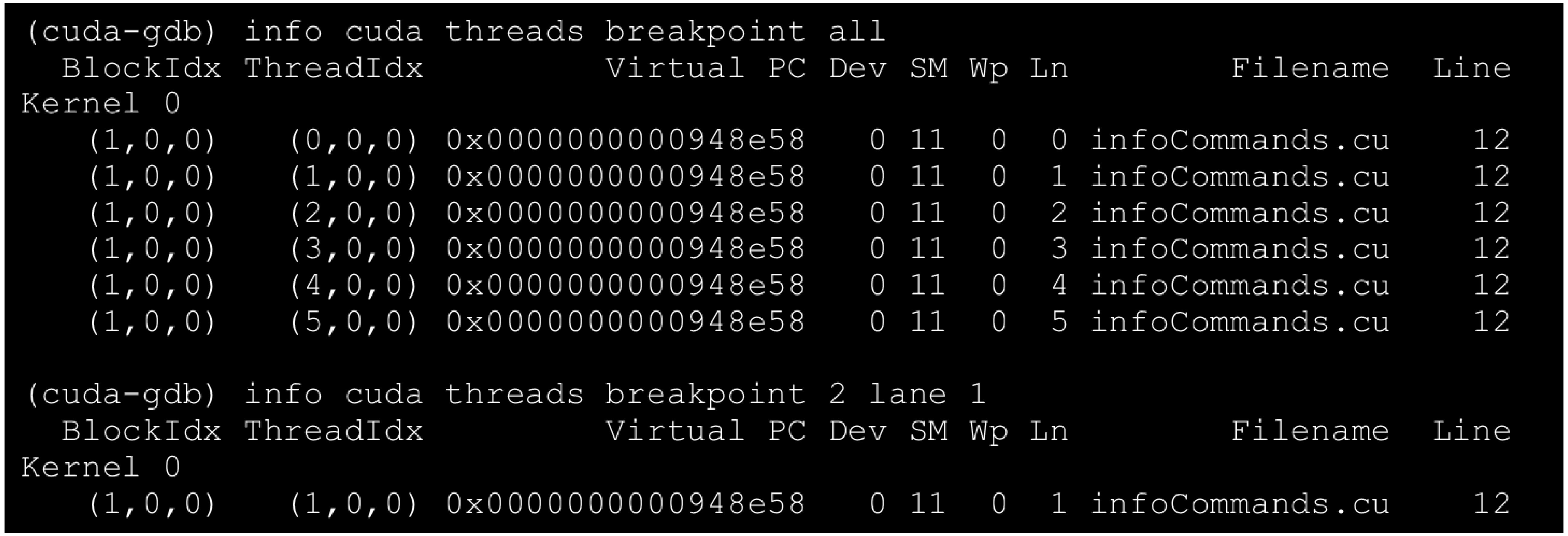
NVIDIA Developer Blog: Using Nsight Compute to Inspect Your Kernels

OLCF Training, March 2020: Nsight Compute, Nsight Systems

CUDA GDB

COMMAND LINE AND IDE BACKEND DEBUGGER

- Unified CPU and CUDA Debugging
- CUDA-C/PTX/SASS support
- Built on GDB and uses many of the same CLI commands

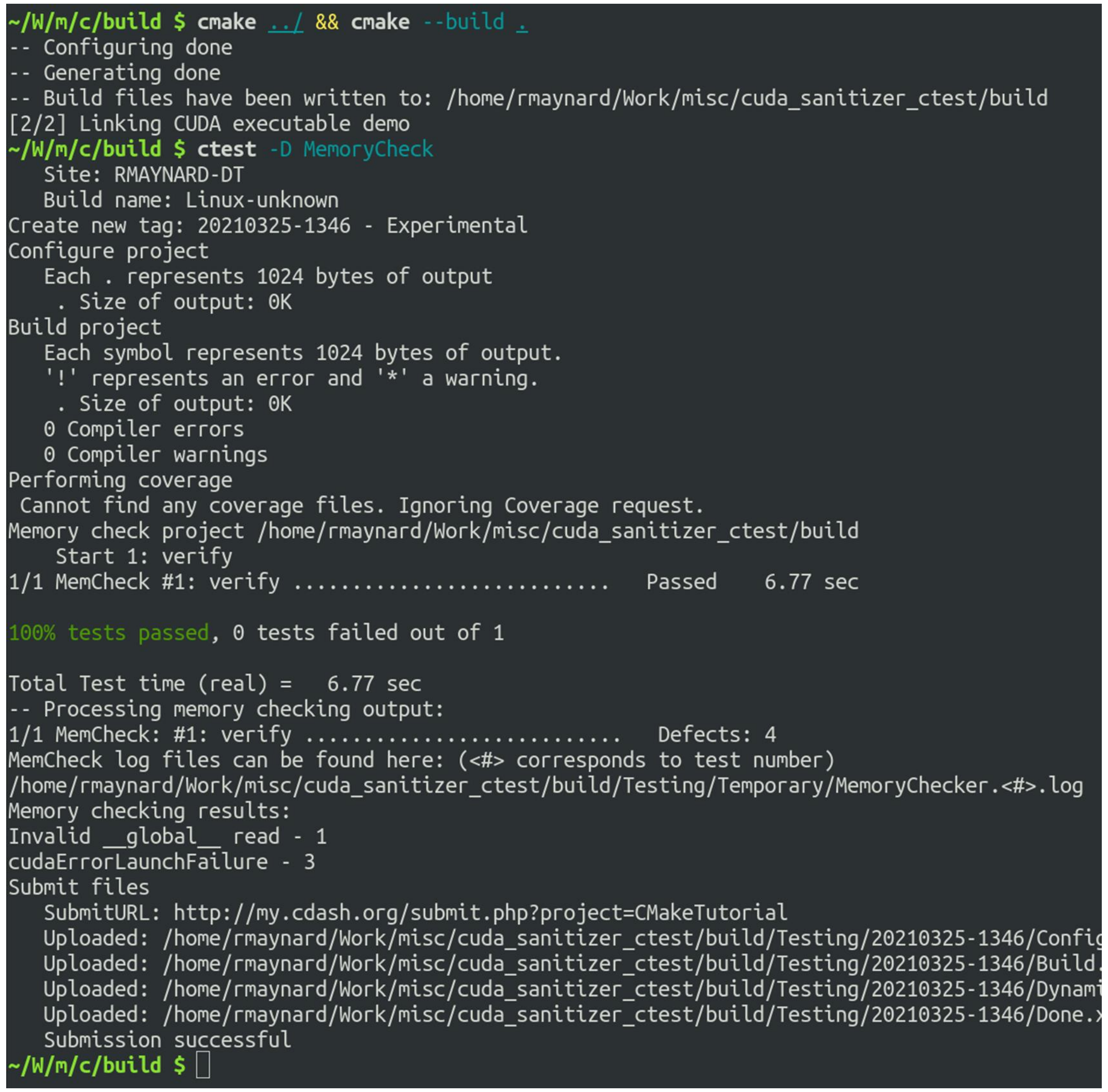


```
(cuda-gdb) info cuda threads breakpoint all
  BlockIdx ThreadIdx         Virtual PC Dev SM Wp Ln        Filename  Line
Kernel 0
   (1,0,0)   (0,0,0) 0x0000000000948e58   0 11  0  0 infoCommands.cu    12
   (1,0,0)   (1,0,0) 0x0000000000948e58   0 11  0  1 infoCommands.cu    12
   (1,0,0)   (2,0,0)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

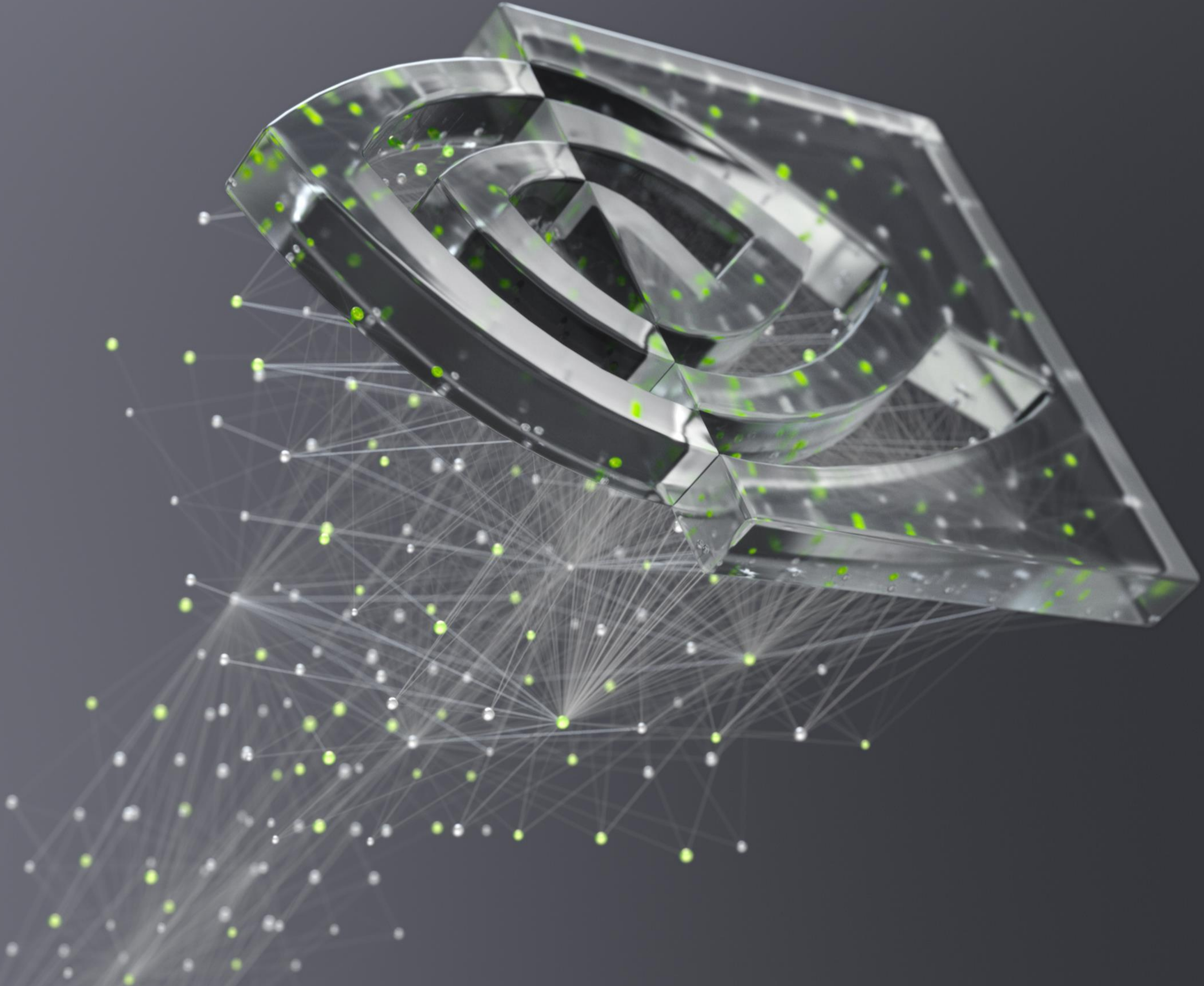
COMPUTE SANITIZER

AUTOMATICALLY SCAN FOR BUGS AND MEMORY ISSUES

- Compute Sanitizer checks correctness issues via sub-tools:
- *Memcheck* – The memory access error and leak detection tool.
- *Racecheck* – The shared memory data access hazard detection tool.
- *Initcheck* – The uninitialized device global memory access detection tool.
- *Synccheck* – The thread synchronization hazard detection tool.



```
~/W/m/c/build $ cmake ../ && cmake --build .
-- Configuring done
-- Generating done
-- Build files have been written to: /home/rmaynard/Work/misc/cuda_sanitizer_ctest/build
[2/2] Linking CUDA executable demo
~/W/m/c/build $ ctest -D MemoryCheck
   Site: RMAYNARD-DT
   Build name: Linux-unknown
Create new tag: 20210325-1346 - Experimental
Configure project
   Each . represents 1024 bytes of output
    . Size of output: 0K
Build project
   Each symbol represents 1024 bytes of output.
   '!' represents an error and '*' a warning.
    . Size of output: 0K
   0 Compiler errors
   0 Compiler warnings
Performing coverage
 Cannot find any coverage files. Ignoring Coverage request.
Memory check project /home/rmaynard/Work/misc/cuda_sanitizer_ctest/build
    Start 1: verify
1/1 MemCheck #1: verify ..........................   Passed    6.77 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) =   6.77 sec
-- Processing memory checking output:
1/1 MemCheck: #1: verify ..........................   Defects: 4
MemCheck log files can be found here: (<#> corresponds to test number)
/home/rmaynard/Work/misc/cuda_sanitizer_ctest/build/Testing/Temporary/MemoryChecker.<#>.log
Memory checking results:
Invalid __global__ read - 1
cudaErrorLaunchFailure - 3
Submit files
   SubmitURL: http://my.cdash.org/submit.php?project=CMakeTutorial
   Uploaded: /home/rmaynard/Work/misc/cuda_sanitizer_ctest/build/Testing/20210325-1346/Config
   Uploaded: /home/rmaynard/Work/misc/cuda_sanitizer_ctest/build/Testing/20210325-1346/Build.
   Uploaded: /home/rmaynard/Work/misc/cuda_sanitizer_ctest/build/Testing/20210325-1346/Dynami
   Uploaded: /home/rmaynard/Work/misc/cuda_sanitizer_ctest/build/Testing/20210325-1346/Done.x
   Submission successful
~/W/m/c/build $
```



NVIDIA