	Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current	1790 - gemm2dBlockTiling	(32, 32, 1)x(256, 1, 1)	656.22 ms	590,598,959	0 - NVIDIA GeForce RTX 3060 Laptop GPU	900.00 Mhz	[22613] 05_2d_block_tiling	

Summary | Details | Source | Context | Comments | Raw | Session

Compare | Tools | View | Export

GPU Speed Of Light Throughput

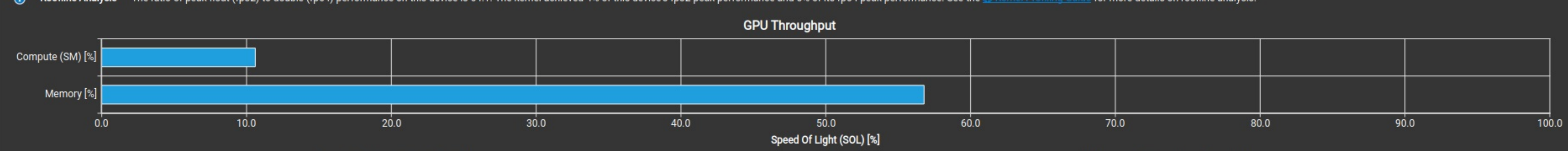
All

High-level overview of the throughput for compute and memory resources of the GPU. For each unit, the throughput reports the achieved percentage of utilization with respect to the theoretical maximum. Breakdowns show the throughput for each individual sub-metric of Compute and Memory to clearly identify the highest contributor. High-level overview of the utilization for compute and memory resources of the GPU presented as a roofline chart.

Compute (SM) Throughput [%]	10.60	Duration [ms]	656.22
Memory Throughput [%]	56.79	Elapsed Cycles [cycle]	590598959
L1/TEX Cache Throughput [%]	87.75	SM Active Cycles [cycle]	577253517.87
L2 Cache Throughput [%]	56.79	SM Frequency [Mhz]	900.00
DRAM Throughput [%]	15.63	DRAM Frequency [Ghz]	6.99

Latency Issue 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.

Roofline Analysis The ratio of peak float (fp32) to double (fp64) performance on this device is 64:1. The kernel achieved 4% of this device's fp32 peak performance and 0% of its fp64 peak performance. See the Kernel Profiling Guide for more details on roofline analysis.

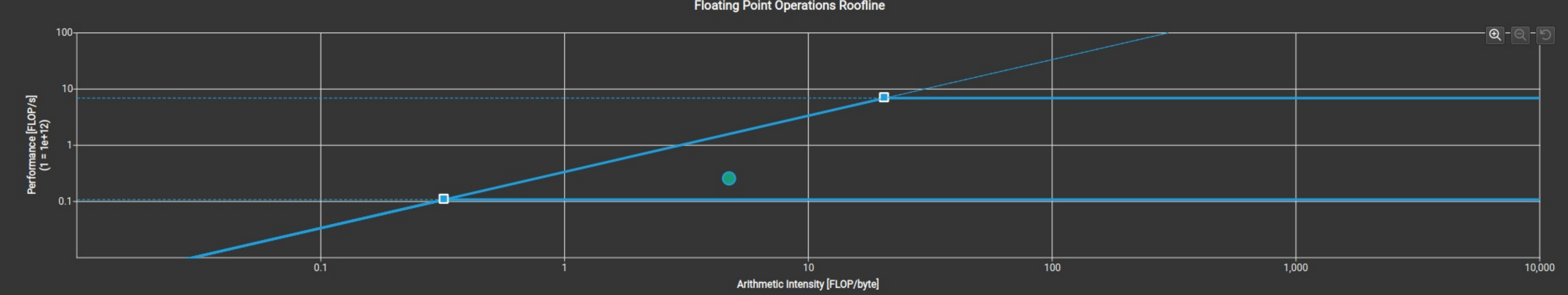


Compute Throughput Breakdown

SM: Inst Executed Pipe Lsu [%]	10.60
SM: Issue Active [%]	6.80
SM: Inst Executed [%]	6.80
SM: Pipe Fma Cycles Active [%]	3.91
SM: Mio Inst Issued [%]	3.69
SM: Pipe Alu Cycles Active [%]	3.05
SM: Mio Pq Read Cycles Active [%]	2.75
SM: Mio2rf Writeback Active [%]	2.38
SM: Pipe Fmaheavy Cycles Active [%]	1.98
SM: Mio Pq Write Cycles Active [%]	1.70
SM: Inst Executed Pipe Adu [%]	0.47
SM: Inst Executed Pipe Uniform [%]	0.04
SM: Inst Executed Pipe Cbu Pred On Any [%]	0.00
SM: Pipe Tensor Cycles Active [%]	0
SM: Pipe Fp64 Cycles Active [%]	0
SM: Memory Throughput Internal Activity [%]	0
IDC: Request Cycles Active [%]	0
SM: Instruction Throughput Internal Activity [%]	0
SM: Inst Executed Pipe Xu [%]	0
SM: Inst Executed Pipe Tex [%]	0
SM: Inst Executed Pipe Ipa [%]	0

Memory Throughput Breakdown

L2: Xbar2lts Cycles Active [%]	56.79
L2: T Sectors [%]	47.59
L1: M L1tex2xbar Req Cycles Active [%]	43.88
L1: Data Pipe Lsu Wavefronts [%]	31.47
DRAM: Dram Sectors [%]	15.63
DRAM: Cycles Active [%]	15.55
L2: D Sectors [%]	13.85
GPU: Compute Memory Access Throughput Internal Activity [%]	12.29
L2: T Tag Requests [%]	12.28
L1: Lsuin Requests [%]	10.60
L1: Lsu Writeback Active [%]	5.84
L2: Lts2xbar Cycles Active [%]	2.81
L1: Data Bank Reads [%]	2.42
L1: M Xbar2l1tex Read Sectors [%]	2.15
L1: Data Bank Writes [%]	1.12
L2: D Sectors Fill Device [%]	0.55
L1: Texin Sm2tex Req Cycles Active [%]	0.03
L2: D Sectors Fill Sysmem [%]	0.00
L1: Tex Writeback Active [%]	0
L2: D Atomic Input Cycles Active [%]	0
L1: F Wavefronts [%]	0
L1: Data Pipe Tex Wavefronts [%]	0
GPU: Compute Memory Request Throughput Internal Activity [%]	0



PM Sampling

Timeline view of PM metrics sampled periodically over the workload duration. Data is collected across multiple passes. Use this section to understand how workload behavior changes over its runtime.

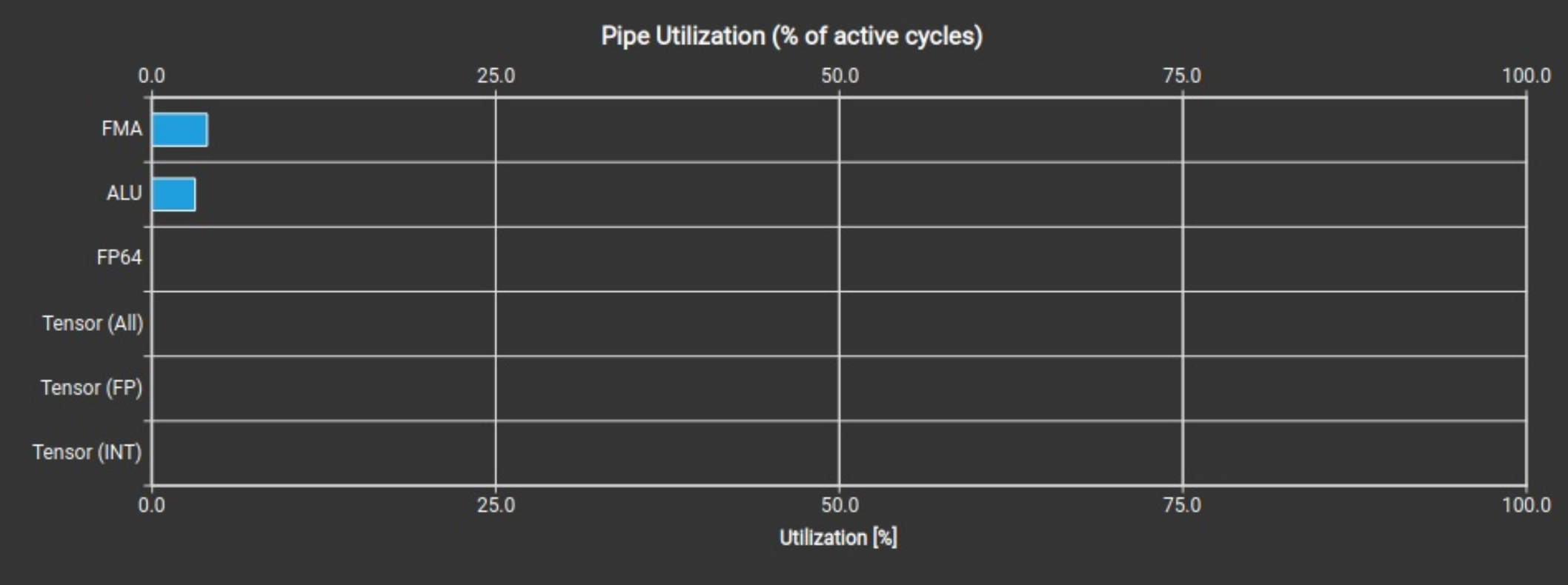
Maximum Sampling Interval [us]	128	# Pass Groups	2
Maximum Buffer Size [Mbytes]	18.69	Dropped Samples [sample]	9

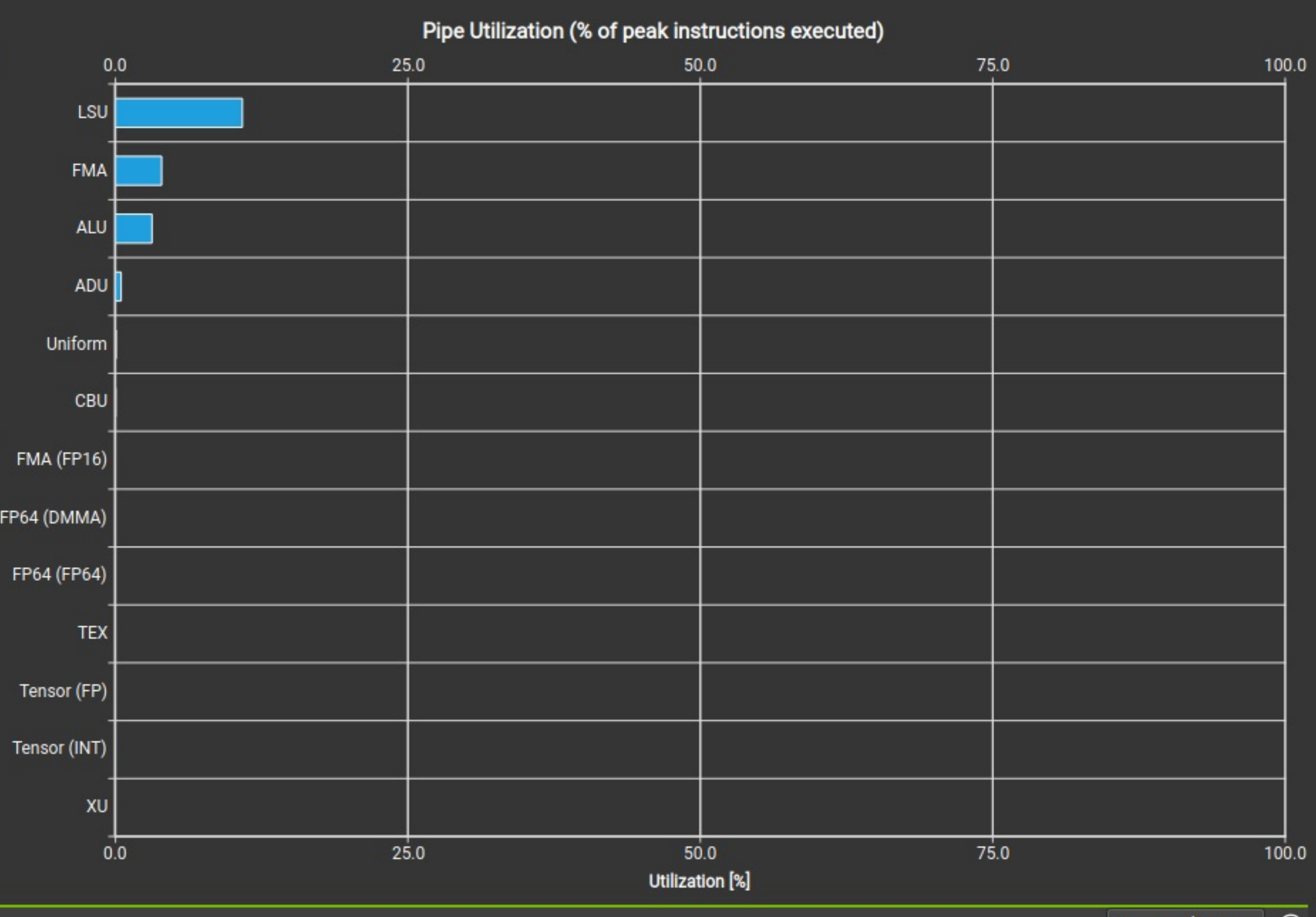
Compute Workload Analysis

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]	0.27	SM Busy [%]	6.96
Executed Ipc Active [inst/cycle]	0.28	Issue Slots Busy [%]	6.96
Issued Ipc Active [inst/cycle]	0.28		

Low Utilization
Est. Local Speedup: 96.00% All compute pipelines are under-utilized. Either this workload is very small or it doesn't issue enough warps per scheduler. Check the Launch Statistics and Scheduler Statistics sections for further details.





Memory Workload Analysis

Memory Chart

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/s]	52.20	Mem Busy [%]	47.59
L1/TEX Hit Rate [%]	95.46	Max Bandwidth [%]	56.79
L2 Hit Rate [%]	98.84	Mem Pipes Busy [%]	10.60
L2 Compression Success Rate [%]	0	L2 Compression Ratio	0

L1TEX Global Load Access Pattern
Est. Speedup: 75.60% The memory access pattern for global loads from L1TEX might not be optimal. On average, only 4.4 of the 32 bytes transmitted per sector are utilized by each thread. This could possibly be caused by a stride between threads. Check the Source Counters section for uncoalesced global loads.

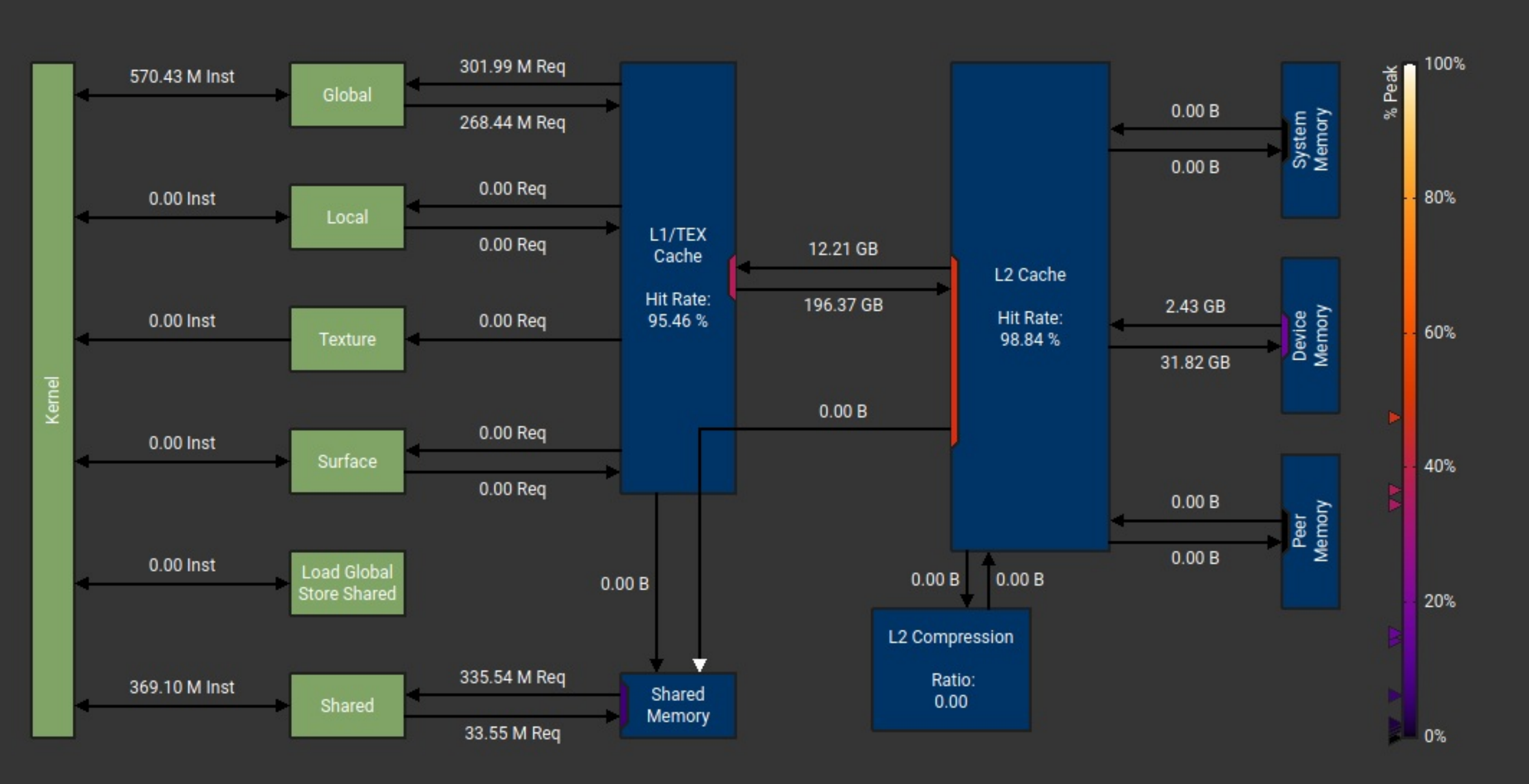
L1TEX Global Store Access Pattern
Est. Speedup: 76.78% The memory access pattern for global stores to L1TEX might not be optimal. On average, only 4.0 of the 32 bytes transmitted per sector are utilized by each thread. This could possibly be caused by a stride between threads. Check the Source Counters section for uncoalesced global stores.

Shared Load Bank Conflicts
Est. Speedup: 43.88% The memory access pattern for shared loads might not be optimal and causes on average a 3.2 - way bank conflict across all 335544320 shared load requests.This results in 536870912 bank conflicts, which represent 50.00% of the overall 1073741824 wavefronts for shared loads. Check the Source Counters section for uncoalesced shared loads.

Shared Store Bank Conflicts
Est. Speedup: 12.46% The memory access pattern for shared stores might not be optimal and causes on average a 1.2 - way bank conflict across all 33554432 shared store requests.This results in 5544275 bank conflicts, which represent 14.20% of the overall 39041207 wavefronts for shared stores. Check the Source Counters section for uncoalesced shared stores.

Memory Chart

Show As: Transfer Size

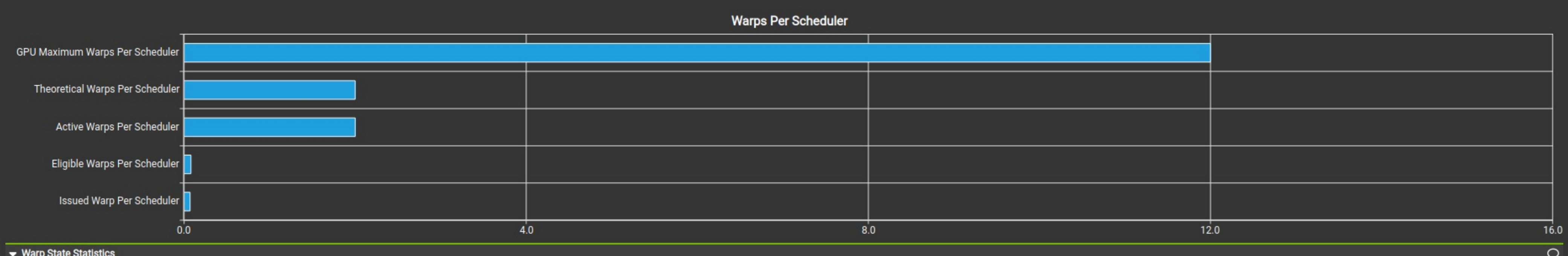


Scheduler Statistics

Summary of the activity of the schedulers issuing instructions. Each scheduler maintains a pool of warps that it can issue instructions for. The upper bound of warps in the pool (Theoretical Warps) is limited by the launch configuration. On every cycle each scheduler checks the state of the allocated warps in the pool (Active Warps). Active warps that are not stalled (Eligible Warps) are ready to issue their next instruction. From the set of eligible warps the scheduler selects a single warp from which to issue one or more instructions (Issued Warp). On cycles with no eligible warps, the issue slot is skipped and no instruction is issued. Having many skipped issue slots indicates poor latency hiding.

Active Warps Per Scheduler [warp]	2.00	No Eligible [%]	93.04
Eligible Warps Per Scheduler [warp]	0.08	One or More Eligible [%]	6.96
Issued Warp Per Scheduler	0.07		

Issue Slot Utilization
Est. Local Speedup: 43.21% Every scheduler is capable of issuing one instruction per cycle, but for this kernel each scheduler only issues an instruction every 14.4 cycles. This might leave hardware resources underutilized and may lead to less optimal performance. Out of the maximum of 12 warps per scheduler, this kernel allocates an average of 2.00 active warps per scheduler, but only an average of 0.08 warps were eligible per cycle. Eligible warps are the subset of active warps that are ready to issue their next instruction. Every cycle with no eligible warp results in no instruction being issued and the issue slot remains unused. To increase the number of eligible warps, avoid possible load imbalances due to highly different execution durations per warp. Reducing stalls indicated on the Warp State Statistics and Source Counters sections can help, too.



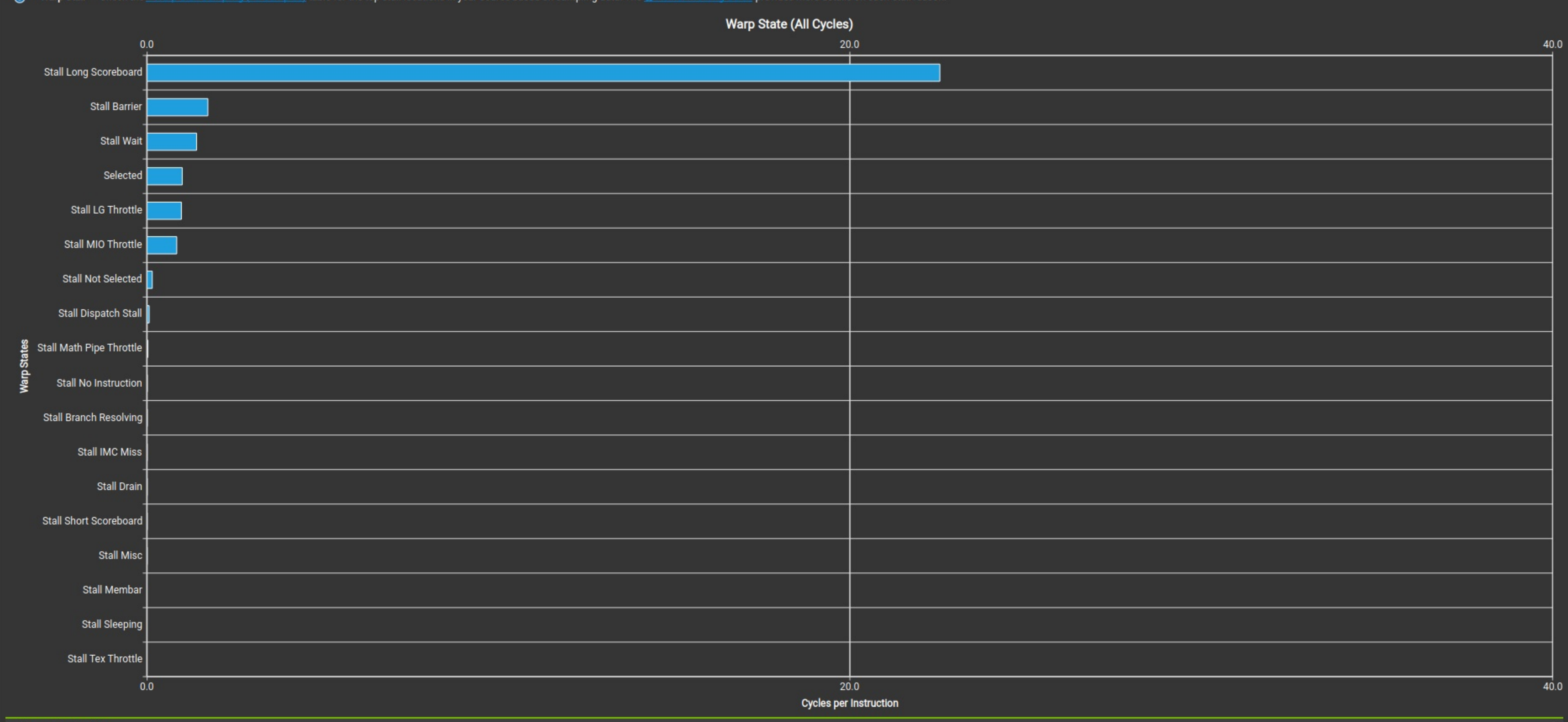
Warp State Statistics

Analysis of the states in which all warps spent cycles during the kernel execution. The warp states describe a warp's readiness or inability to issue its next instruction. The warp cycles per instruction define the latency between two consecutive instructions. The higher the value, the more warp parallelism is required to hide this latency. For each warp state, the chart shows the average number of cycles spent in that state per issued instruction. Stalls are not always impacting the overall performance nor are they completely avoidable. Only focus on stall reasons if the schedulers fail to issue every cycle. When executing a kernel with mixed library and user code, these metrics show the combined values.

Warp Cycles Per Issued Instruction [cycle]	28.75	Avg. Active Threads Per Warp	32
Warp Cycles Per Executed Instruction [cycle]	28.75	Avg. Not Predicated Off Threads Per Warp	31.97

Long Scoreboard Stalls
Est. Speedup: 43.21% On average, each warp of this kernel spends 22.6 cycles being stalled waiting for a scoreboard dependency on a L1TEX (local, global, surface, texture) operation. Find the instruction producing the data being waited upon to identify the culprit. To reduce the number of cycles waiting on L1TEX data accesses verify the memory access patterns are optimal for the target architecture, attempt to increase cache hit rates by increasing data locality (coalescing), or by changing the cache configuration. Consider moving frequently used data to shared memory. This stall type represents about 78.5% of the total average of 28.7 cycles between issuing two instructions.

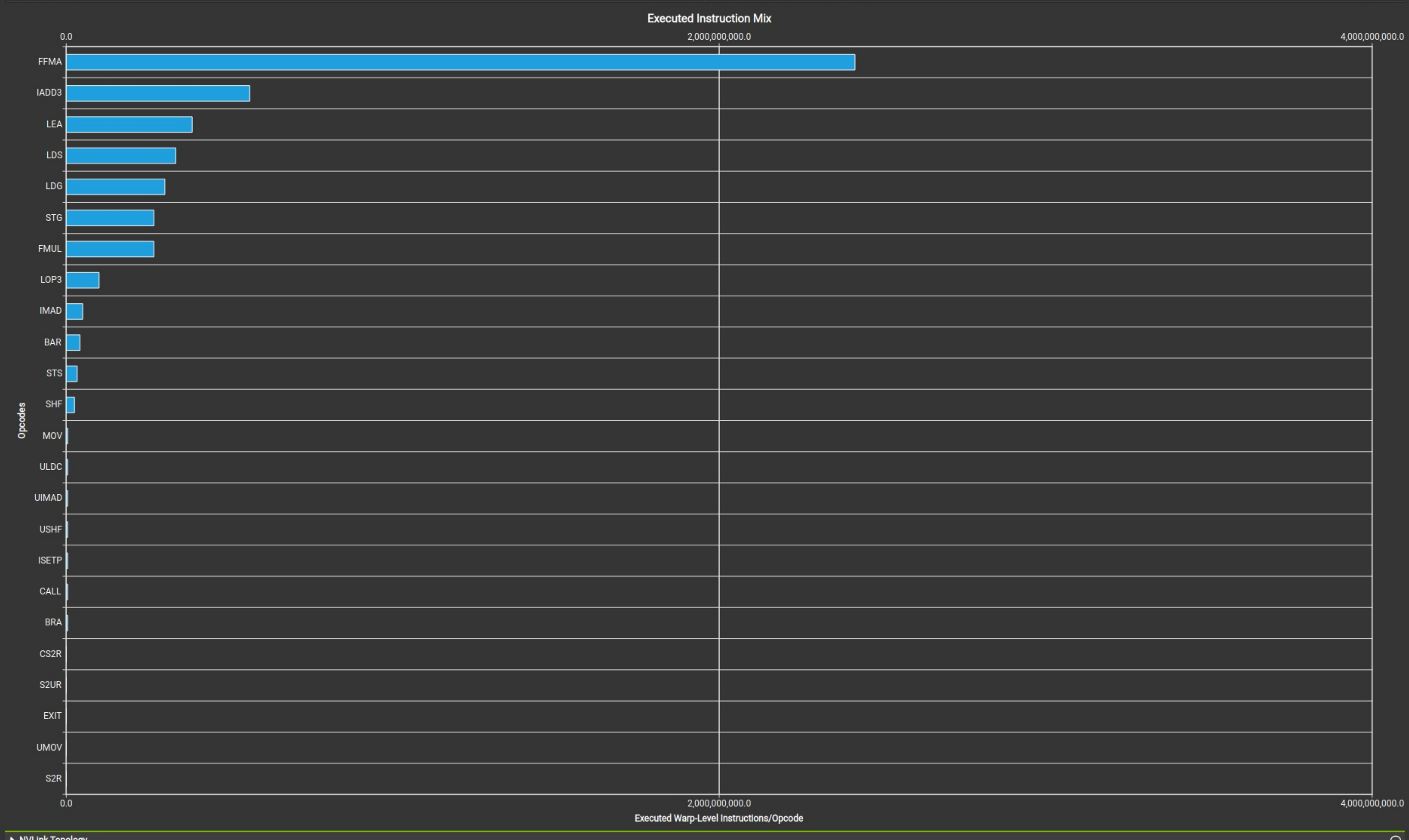
Warp Stall Check the Warp Stall Sampling (All Samples) table for the top stall locations in your source based on sampling data. The Kernel Profiling Guide provides more details on each stall reason.



Instruction Statistics

Statistics of the executed low-level assembly instructions (SASS). The instruction mix provides insight into the types and frequency of the executed instructions. A narrow mix of instruction types implies a dependency on few instruction pipelines, while others remain unused. Using multiple pipelines allows hiding latencies and enables parallel execution. Note that 'Instructions/Opcode' and 'Executed Instructions' are measured differently and can diverge if cycles are spent in system calls.

Executed Instructions [inst]	4819763200	Avg. Executed Instructions Per Scheduler [inst]	40164693.33
Issued Instructions [inst]	4819773239	Avg. Issued Instructions Per Scheduler [inst]	40164776.99



NVLink Topology

NVLink Topology diagram shows logical NVLink connections with transmit/receive throughput.

NVLink Tables

Detailed tables with properties for each NVLink.

NUMA Affinity

Non-uniform memory access (NUMA) affinities based on compute and memory distances for all GPUs.

Launch Statistics

Summary of the configuration used to launch the kernel. The launch configuration defines the size of the kernel grid, the division of the grid into blocks, and the GPU resources needed to execute the kernel. Choosing an efficient launch configuration maximizes device utilization.

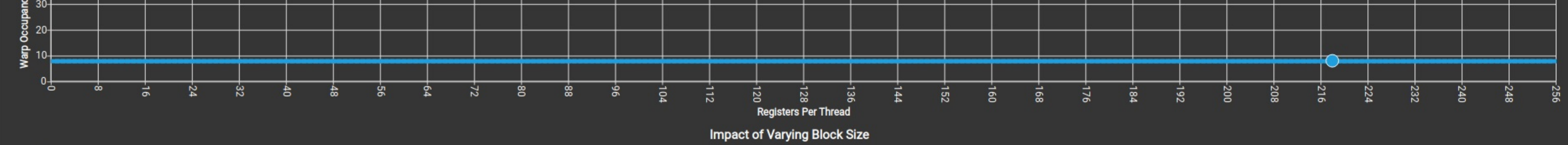
Grid Size	1024	Function Cache Configuration	CachePreferNone
Registers Per Thread [register/thread]	218	Static Shared Memory Per Block [Kbyte/block]	8.19
Block Size	256	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	262144	Driver Shared Memory Per Block [Kbyte/block]	1.02
Waves Per SM	34.13	Shared Memory Configuration Size [Kbyte]	16.38
Uses Green Context	0	# SMs [SM]	30

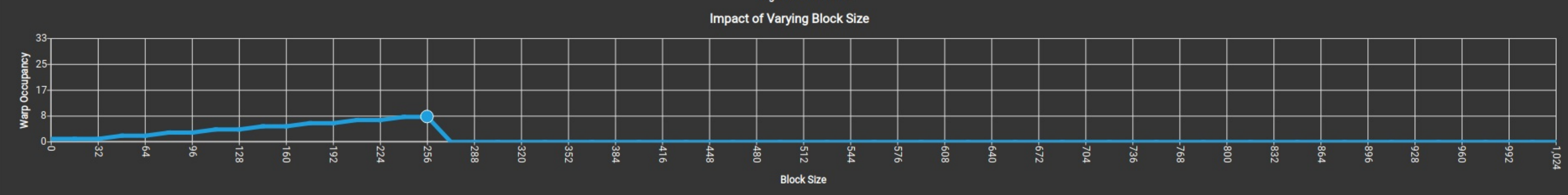
Occupancy

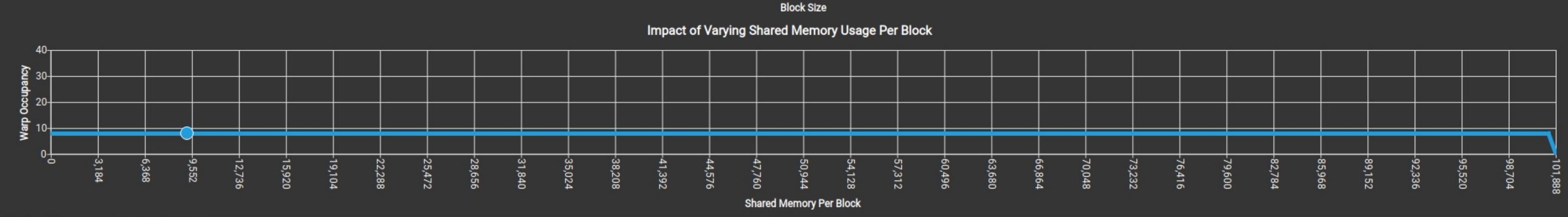
Occupancy is the ratio of the number of active warps per multiprocessor to the maximum number of possible active warps. Another way to view occupancy is the percentage of the hardware's ability to process warps that is actively in use. Higher occupancy does not always result in higher performance, however, low occupancy always reduces the ability to hide latencies, resulting in overall performance degradation. Large discrepancies between the theoretical and the achieved occupancy during execution typically indicates highly imbalanced workloads.

Theoretical Occupancy [%]	16.67	Block Limit Registers [block]	1
Theoretical Active Warps per SM [warp]	8	Block Limit Shared Mem [block]	1
Achieved Occupancy [%]	16.66	Block Limit Warps [block]	6
Achieved Active Warps Per SM [warp]	8.00	Block Limit SM [block]	16

Theoretical Occupancy
Est. Speedup: 43.21% The 2.00 theoretical warps per scheduler this kernel can issue according to its occupancy are below the hardware maximum of 12. This kernel's theoretical occupancy (16.7%) is limited by the number of required registers. This kernel's theoretical occupancy (16.7%) is limited by the required amount of shared memory.







GPU and Memory Workload Distribution

Analysis of workload distribution in active cycles of SM, SMP, SMSP, L1 & L2 caches, and DRAM

Average SM Active Cycles [cycle]	577253517.87	Average L1 Active Cycles [cycle]	577253517.87
Average L2 Active Cycles [cycle]	549549980.17	Average SMSP Active Cycles [cycle]	577108338.29
Average DRAM Active Cycles [cycle]	713652565.33	Total SM Elapsed Cycles [cycle]	17719378630
Total L1 Elapsed Cycles [cycle]	17719378630	Total L2 Elapsed Cycles [cycle]	13701895704
Total SMSP Elapsed Cycles [cycle]	70877514520	Total DRAM Elapsed Cycles [cycle]	27537668096

Workload Distribution

	Average	Min	Max	Sum
SM Active Cycles	577253517.87	574445160	591126704	17317605536
SMSP Active Cycles	577108338.29	574567371	590864554	69253000595
L1 Active Cycles	577253517.87	574445160	591126704	17317605536
L2 Active Cycles	549549980.17	548630405	552685089	13189199524
DRAM Active Cycles	713652565.33	708433824	718224688	4281915392

Source Counters

Source metrics, including branch efficiency and sampled warp stall reasons. Warp Stall Sampling metrics are periodically sampled over the kernel runtime. They indicate when warps were stalled and couldn't be scheduled. See the documentation for a description of all stall reasons. Only focus on stalls if the schedulers fail to issue every cycle.

Branch Instructions [inst]	8396800	Branch Efficiency [%]	100
Branch Instructions Ratio [%]	0.00	Avg. Divergent Branches	0

Uncoalesced Global Accesses
Est. Speedup: 83.57% This kernel has uncoalesced global accesses resulting in a total of 15032385536 excessive sectors (87% of the total 17314086912 sectors). Check the L2 Theoretical Sectors Global Excessive table for the primary source locations. The CUDA Programming Guide has additional information on reducing uncoalesced device memory accesses.

L2 Theoretical Sectors Global Excessive

Location	Value	Value (%)
05_2d_block_tiling.cu:91 (0x500d90d70 in gemm2dBlockTiling)	117,440,512	1
05_2d_block_tiling.cu:93 (0x500d90c60 in gemm2dBlockTiling)	117,440,512	1
05_2d_block_tiling.cu:91 (0x500d90c50 in gemm2dBlockTiling)	117,440,512	1
05_2d_block_tiling.cu:93 (0x500d90b50 in gemm2dBlockTiling)	117,440,512	1
05_2d_block_tiling.cu:91 (0x500d90b40 in gemm2dBlockTiling)	117,440,512	1

Uncoalesced Shared Accesses
Est. Speedup: 47.39% This kernel has uncoalesced shared accesses resulting in a total of 536870912 excessive wavefronts (48% of the total 1107296256 wavefronts). Check the L1 Wavefronts Shared Excessive table for the primary source locations. The CUDA Best Practices Guide has an example on optimizing shared memory accesses.

L1 Wavefronts Shared Excessive

Location	Value	Value (%)
05_2d_block_tiling.cu:77 (0x500d8cfe0 in gemm2dBlockTiling)	16,777,216	3
05_2d_block_tiling.cu:77 (0x500d8cfd0 in gemm2dBlockTiling)	16,777,216	3
05_2d_block_tiling.cu:77 (0x500d8cf30 in gemm2dBlockTiling)	16,777,216	3
05_2d_block_tiling.cu:77 (0x500d8cf20 in gemm2dBlockTiling)	16,777,216	3
05_2d_block_tiling.cu:77 (0x500d8ce80 in gemm2dBlockTiling)	16,777,216	3

Warp Stall Sampling (All Samples)

Location	Value	Value (%)
05_2d_block_tiling.cu:50 (0x500d90d90 in gemm2d...	137,454	4
05_2d_block_tiling.cu:93 (0x500d8dc80 in gemm2dB...	61,800	2
05_2d_block_tiling.cu:93 (0x500d8dd90 in gemm2d...	61,724	2
05_2d_block_tiling.cu:93 (0x500d8e040 in gemm2dB...	60,696	2
05_2d_block_tiling.cu:93 (0x500d8f110 in gemm2dB...	60,456	2

Most Instructions Executed

Location	Value	Value (%)
05_2d_block_tiling.cu:50 (0x500d90d90 in gemm2d...	4,194,304	0
05_2d_block_tiling.cu:96 (0x500d90d80 in gemm2d...	4,194,304	0
05_2d_block_tiling.cu:91 (0x500d90d70 in gemm2d...	4,194,304	0
05_2d_block_tiling.cu:93 (0x500d90d60 in gemm2d...	4,194,304	0
05_2d_block_tiling.cu:93 (0x500d90d50 in gemm2d...	4,194,304	0

Follow the *rules outputs* to get guidance on how to navigate through the report and quickly discover performance bottlenecks in this kernel.
You could also disable individual sections to focus on selected performance aspects and make profiling faster.