Current	Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
	562 - vectorizedVectorSum	(195313, 1, 1)x(256, 1, 1)	12.75 ms	18,494,396	0 - NVIDIA GeForce RTX 4060	1.45 Ghz	[1703] test_vectorSum	

Summary | Details | Source | Context | Comments | Raw | Session — Compare | Tools | View | Export

The report contains imported source files.

GPU Speed Of Light Throughput

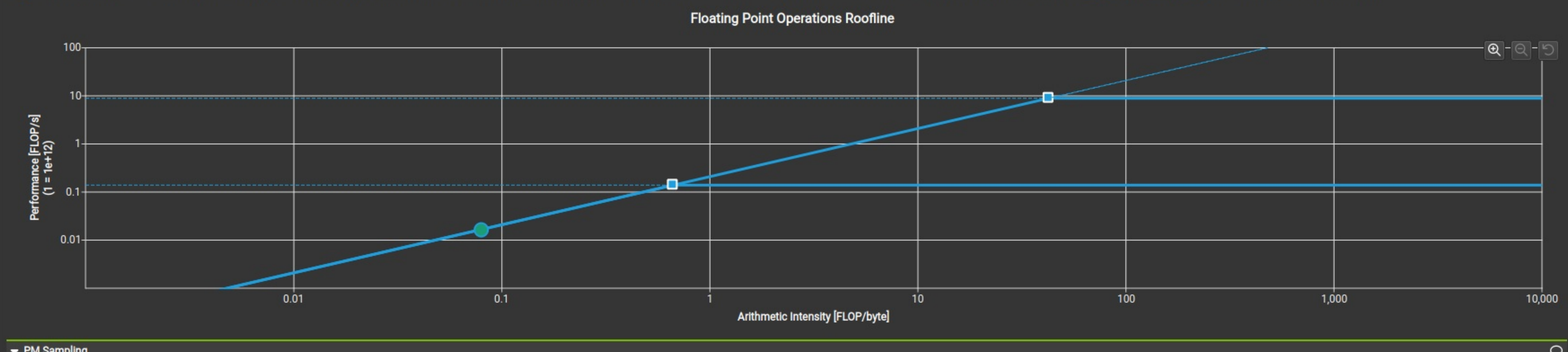
GPU Throughput Rooflines

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 [%]	2.82	Duration [ms]	12.75
Memory Throughput [%]	93.37	Elapsed Cycles [cycle]	18494396
L1/TEX Cache Throughput [%]	14.83	SM Active Cycles [cycle]	17934817.96
L2 Cache Throughput [%]	42.94	SM Frequency [Ghz]	1.45
DRAM Throughput [%]	93.37	DRAM Frequency [Ghz]	6.53

High Throughput The kernel is utilizing greater than 80.0% of the available compute or memory performance of the device. To further improve performance, work will likely need to be shifted from the most utilized to another unit. Start by analyzing DRAM in the Memory Workload Analysis section.

Roofline Analysis The ratio of peak float (fp32) to double (fp64) performance on this device is 64:1. The kernel achieved close to 0% 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.



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.

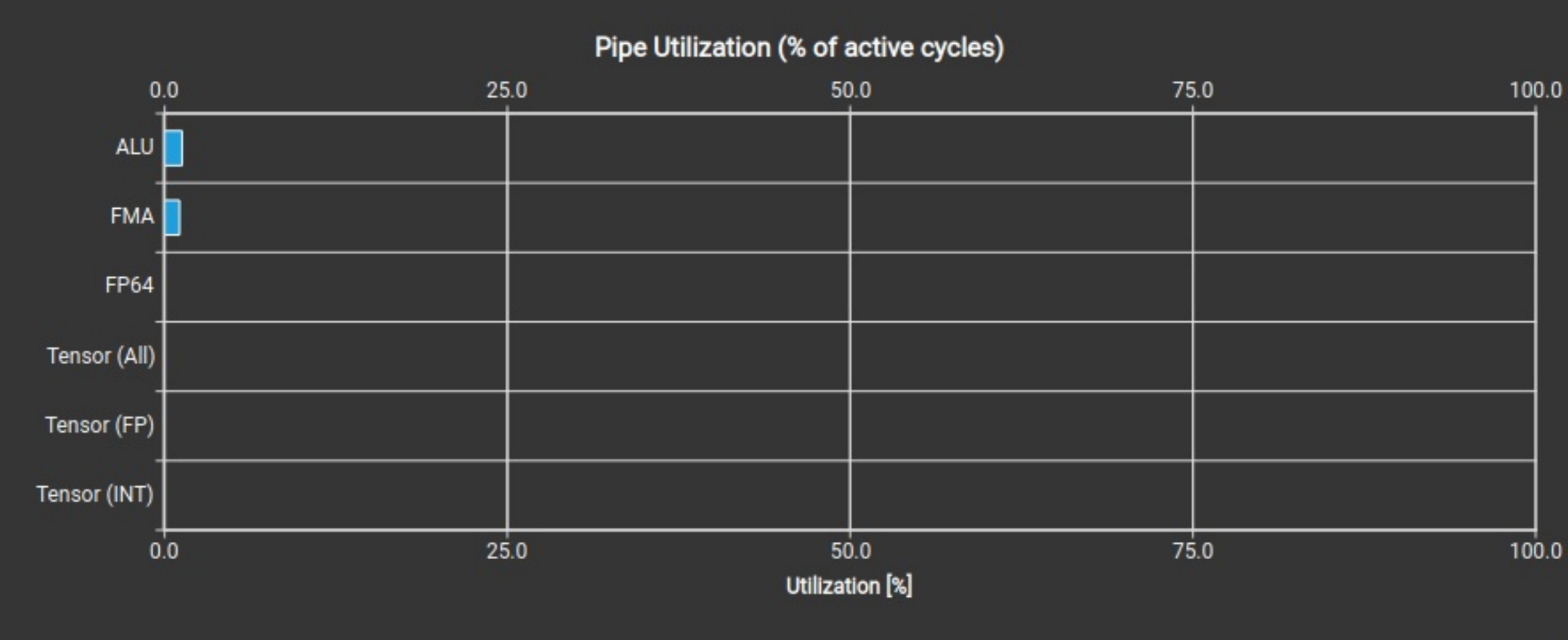
Maximum Sampling Interval [us]	8	# Pass Groups	2
Maximum Buffer Size [Mbytes]	10.94	Dropped Samples [sample]	66

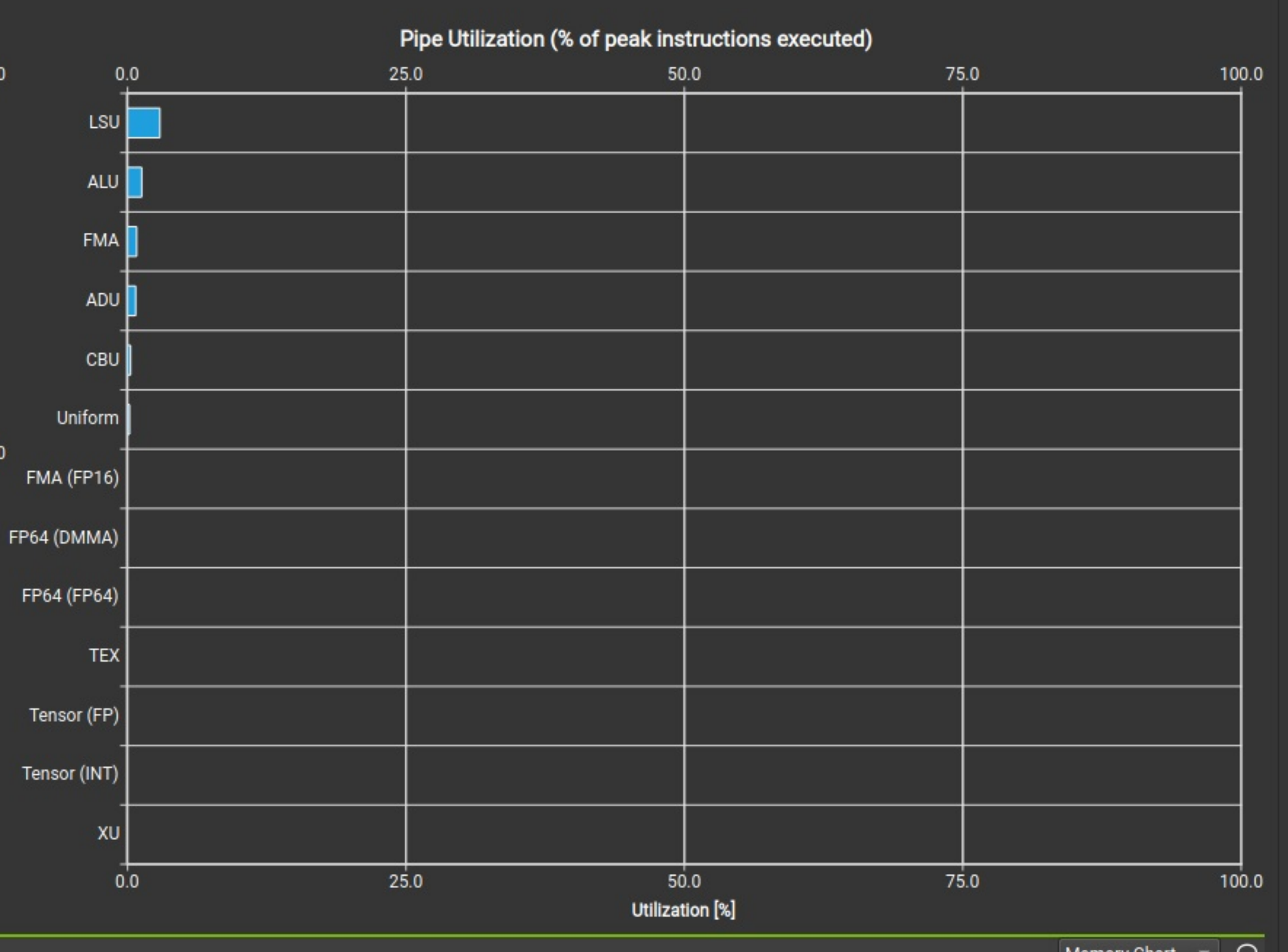
Compute Workload Analysis

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]	0.09	SM Busy [%]	2.36
Executed Ipc Active [inst/cycle]	0.09	Issue Slots Busy [%]	2.36
Issued Ipc Active [inst/cycle]	0.09		

Low Utilization
Est. Local Speedup: 98.73% 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.





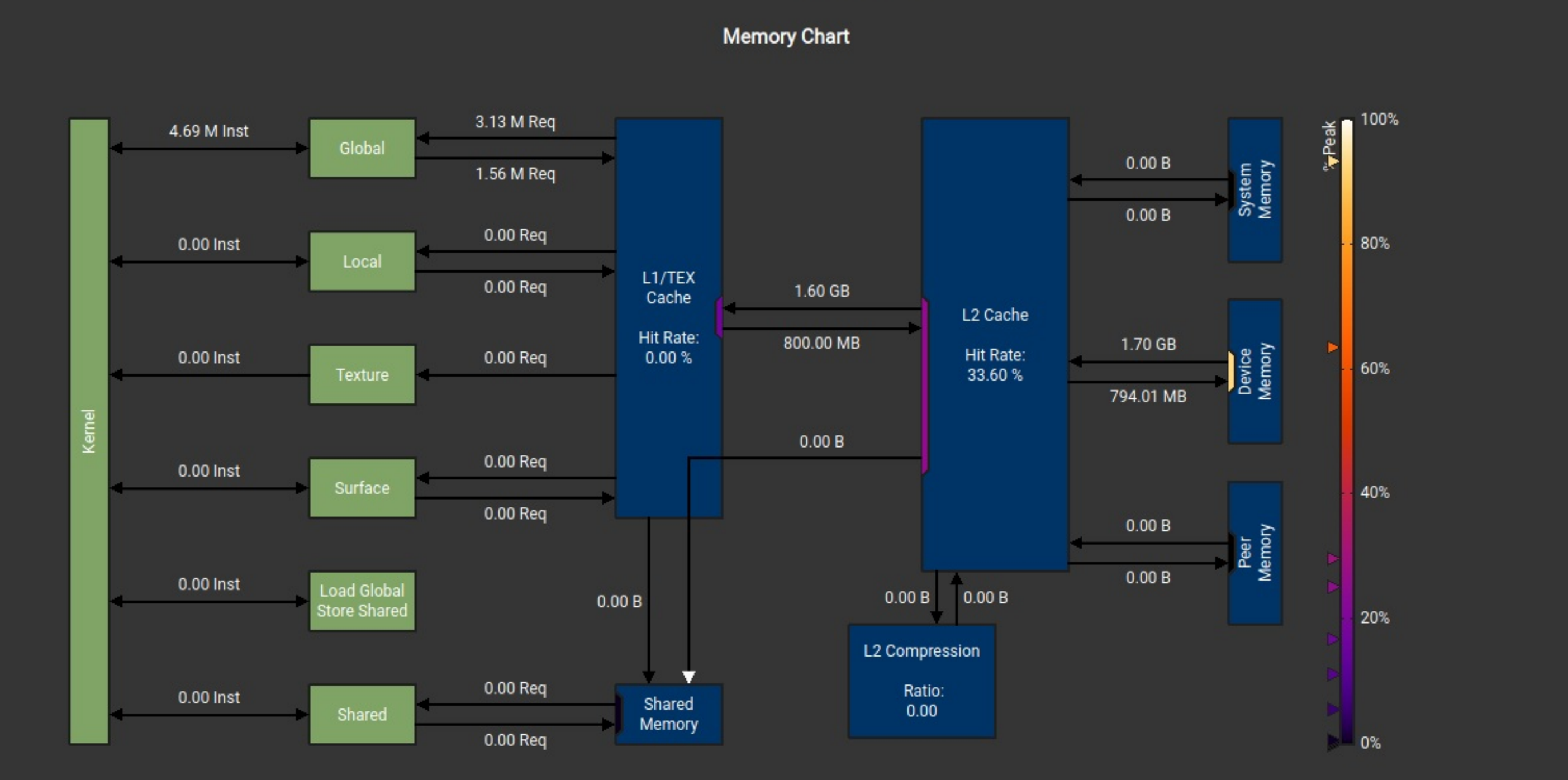
Memory Workload Analysis

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]	195.22	Mem Busy [%]	42.94
L1/TEX Hit Rate [%]	0	Max Bandwidth [%]	93.37
L2 Hit Rate [%]	33.60	Mem Pipes Busy [%]	2.82
L2 Compression Success Rate [%]	0	L2 Compression Ratio	0

Show As: Transfer Size

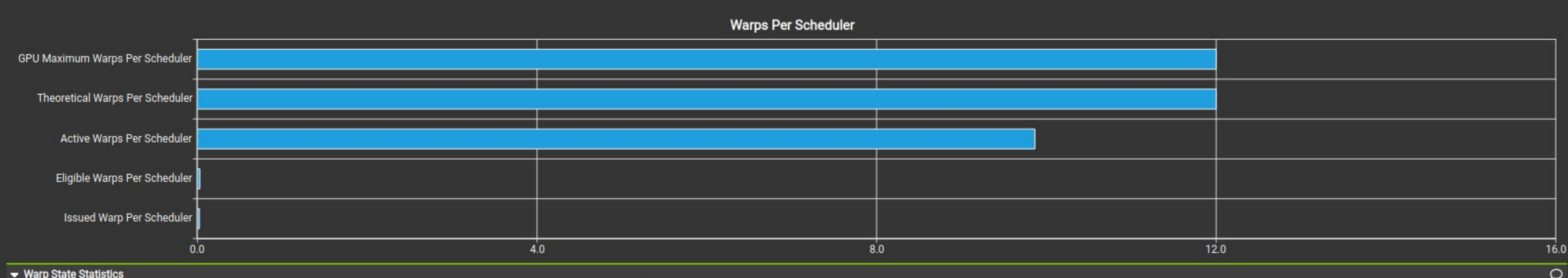


Scheduler Statistics

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]	9.86	No Eligible [%]	97.72
Eligible Warps Per Scheduler [warp]	0.03	One or More Eligible [%]	2.28
Issued Warp Per Scheduler	0.02		

Issue Slot Utilization
Est. Local Speedup: 6.63% Every scheduler is capable of issuing one instruction per cycle, but for this kernel each scheduler only issues an instruction every 44.0 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 9.86 active warps per scheduler, but only an average of 0.03 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.



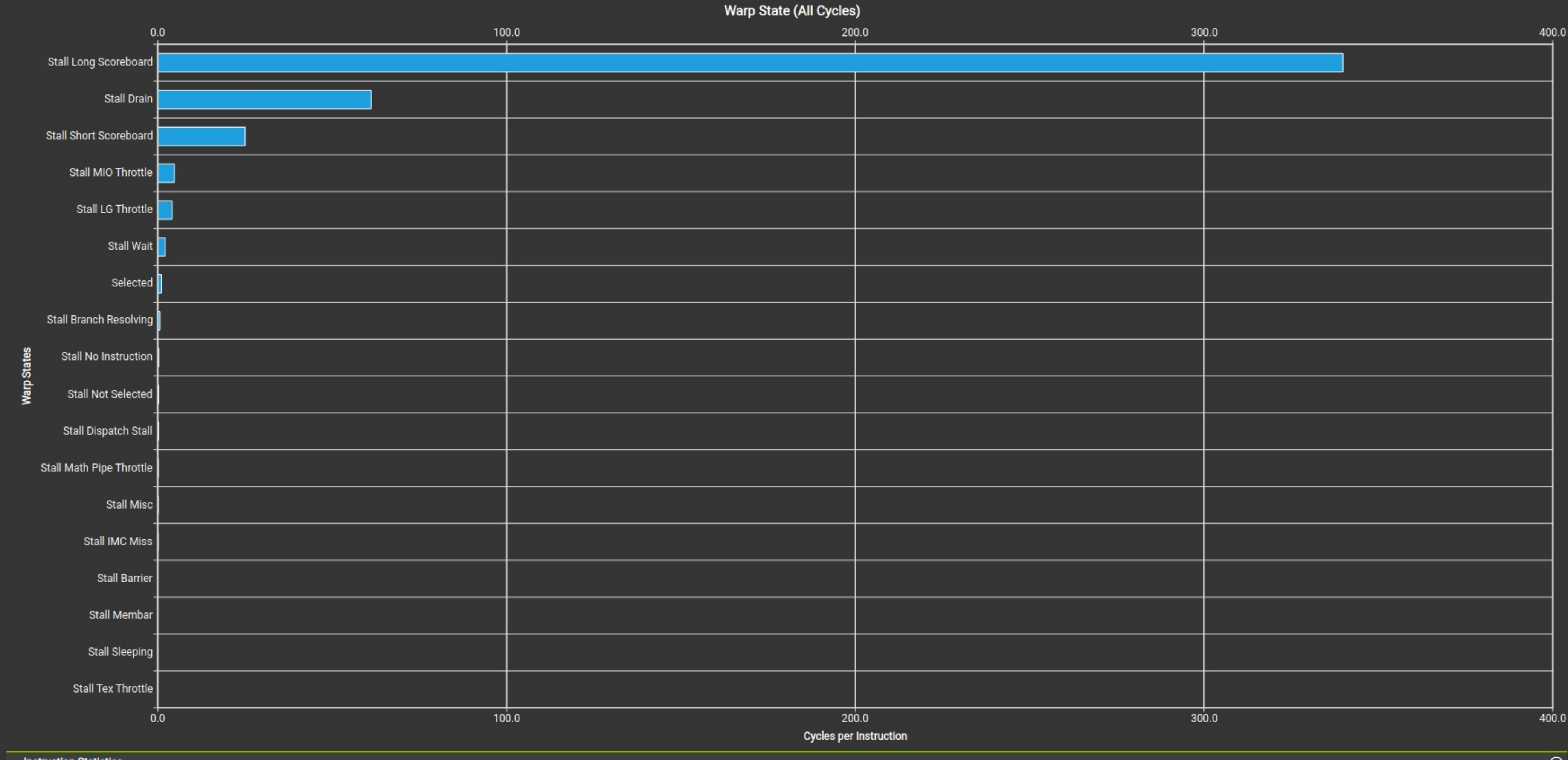
Warp State Statistics

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]	433.61	Avg. Active Threads Per Warp	32.00
Warp Cycles Per Executed Instruction [cycle]	434.08	Avg. Not Predicated Off Threads Per Warp	30.77

Long Scoreboard Stalls
Est. Speedup: 6.63% On average, each warp of this kernel spends 339.8 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 78.4% of the total average of 433.6 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.

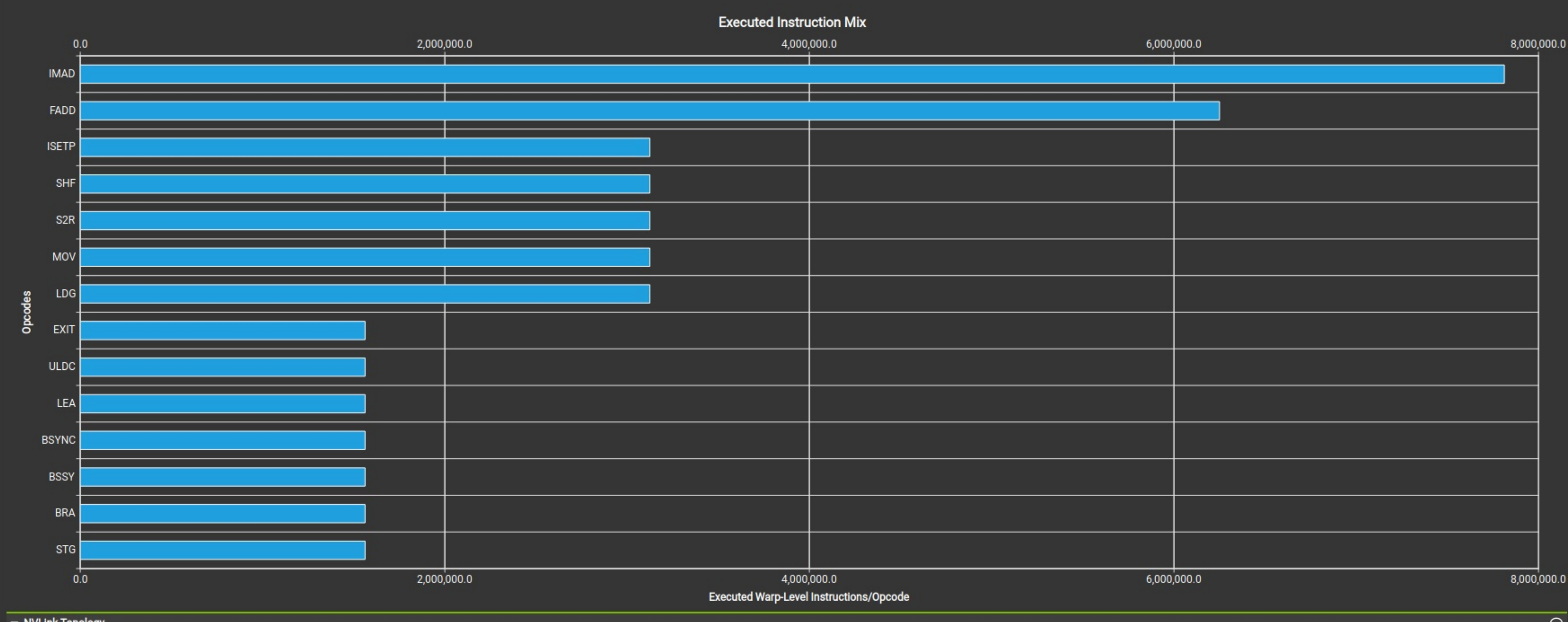


Instruction Statistics

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]	40625063	Avg. Executed Instructions Per Scheduler [inst]	423177.74
Issued Instructions [inst]	40669551	Avg. Issued Instructions Per Scheduler [inst]	423641.16

FP32 Non-Fused Instructions
Est. Speedup: 0.54% This kernel executes 0 fused and 6250000 non-fused FP32 instructions. By converting pairs of non-fused instructions to their fused, higher-throughput equivalent, the achieved FP32 performance could be increased by up to 50% (relative to its current performance). Check the Source page to identify where this kernel executes FP32 instructions.



NVLink Topology

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

NVLink Topology

The system does not have any NVLink connections.

NVLink Tables

Detailed tables with properties for each NVLink.

Logical NVLink Properties

The system does not have any NVLink connections.

NUMA Affinity

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

NUMA ID Table

NUMA information is not available on the target system.

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.

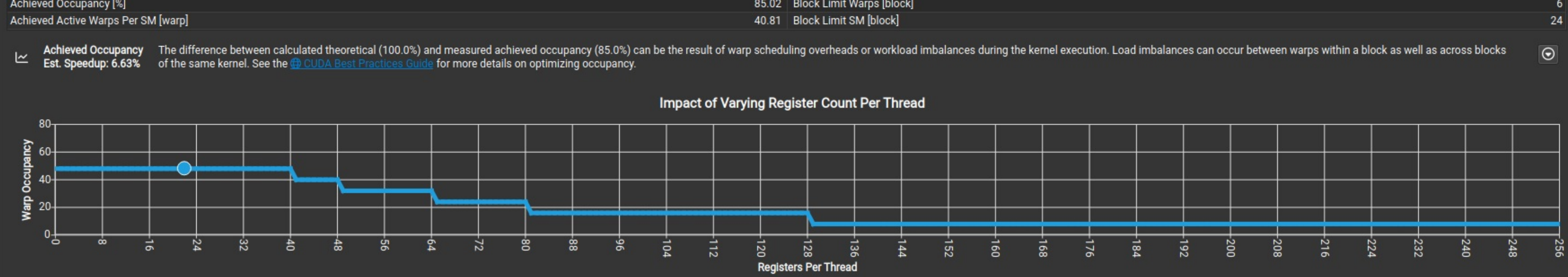
Grid Size	195313	Function Cache Configuration	CachePreferNone
Registers Per Thread [register/thread]	22	Static Shared Memory Per Block [byte/block]	0
Block Size	256	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	50000128	Driver Shared Memory Per Block [Kbyte/block]	1.02
Waves Per SM	1356.34	Shared Memory Configuration Size [Kbyte]	16.38
Uses Green Context	0	# SMs [SM]	24

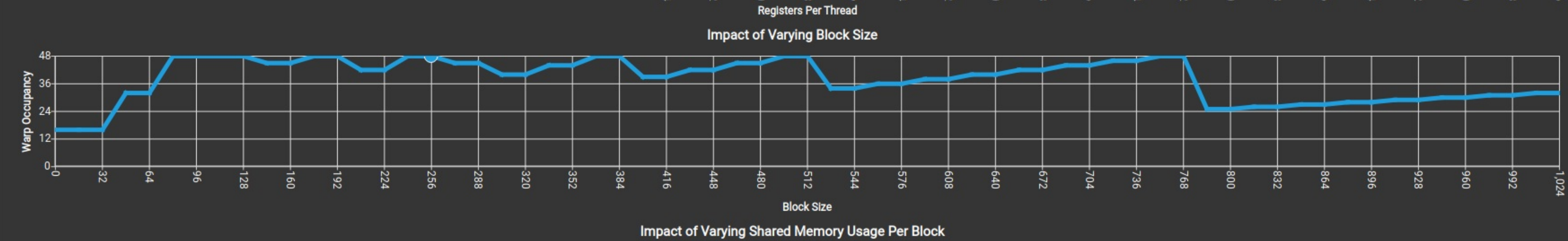
Occupancy

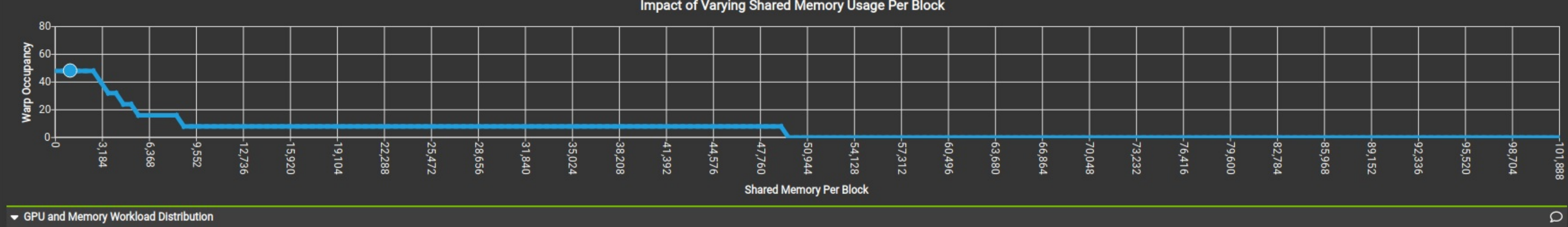
Occupancy is the ratio of the number of active warps per multiprocessor to the maximum number of possible active warps. Another way to view occupancy is the percentage of the hardware's ability to process warps that is actively in use. Higher occupancy does not always result in higher performance, however, 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 [%]	100	Block Limit Registers [block]	10
Theoretical Active Warps per SM [warp]	48	Block Limit Shared Mem [block]	16
Achieved Occupancy [%]	85.02	Block Limit Warps [block]	6
Achieved Active Warps Per SM [warp]	40.81	Block Limit SM [block]	24

Achieved Occupancy
Est. Speedup: 6.63% The difference between calculated theoretical (100.0%) and measured achieved occupancy (85.0%) can be the result of warp scheduling overheads or workload imbalances during the kernel execution. Load imbalances can occur between warps within a block as well as across blocks of the same kernel. See the CUDA Best Practices Guide for more details on optimizing occupancy.







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]	17934817.96	Average L1 Active Cycles [cycle]	17934817.96
Average L2 Active Cycles [cycle]	16429430.17	Average SMSP Active Cycles [cycle]	18621449.78
Average DRAM Active Cycles [cycle]	77781524	Total SM Elapsed Cycles [cycle]	443865424
Total L1 Elapsed Cycles [cycle]	443865424	Total L2 Elapsed Cycles [cycle]	198284304
Total SMSP Elapsed Cycles [cycle]	1775461696	Total DRAM Elapsed Cycles [cycle]	333212672

Workload Distribution

	Average	Min	Max	Sum
SM Active Cycles	17934817.96	17917997	17941538	430435631
SMSP Active Cycles	18621449.78	18617588	18624184	1787659179
L1 Active Cycles	17934817.96	17917997	17941538	430435631
L2 Active Cycles	16429430.17	16425131	16434784	197153162
DRAM Active Cycles	77781524	77773616	77798928	311126096

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]	4690939	Branch Efficiency [%]	100
Branch Instructions Ratio [%]	0.12	Avg. Divergent Branches	0

Warp Stall Sampling (All Samples)

Location	Value	Value (%)
vectorSum.cu:66 (0x500e75b30 in vectorizedVec...	326,569	75
vectorSum.cu:71 (0x500e75ba0 in vectorizedVec...	74,962	17
vectorSum.cu:53 (0x500e75a90 in vectorizedVec...	20,898	5
vectorSum.cu:62 (0x500e75b70 in vectorizedVec...	4,312	1
vectorSum.cu:53 (0x500e75a50 in vectorizedVec...	2,324	1

Most Instructions Executed

Location	Value	Value (%)
vectorSum.cu:70 (0x500e75b90 in vectorizedVec...	1,562,504	4
vectorSum.cu:62 (0x500e75b80 in vectorizedVec...	1,562,504	4
vectorSum.cu:59 (0x500e75ac0 in vectorizedVec...	1,562,504	4
vectorSum.cu:70 (0x500e75ab0 in vectorizedVec...	1,562,504	4
vectorSum.cu:59 (0x500e75aa0 in vectorizedVec...	1,562,504	4

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.