	Result		Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current	574 - gemm_wmma_kernel		(32, 32, 1)x(128, 4, 1)	4.04 ms	6,539,972	0 - NVIDIA GeForce RTX 4060 Laptop GPU	1.62 Ghz	[37663] gemm_tensor	

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

Compare | Tools | View | Export

GPU Speed Of Light Throughput

Roofline Tensor Core

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.

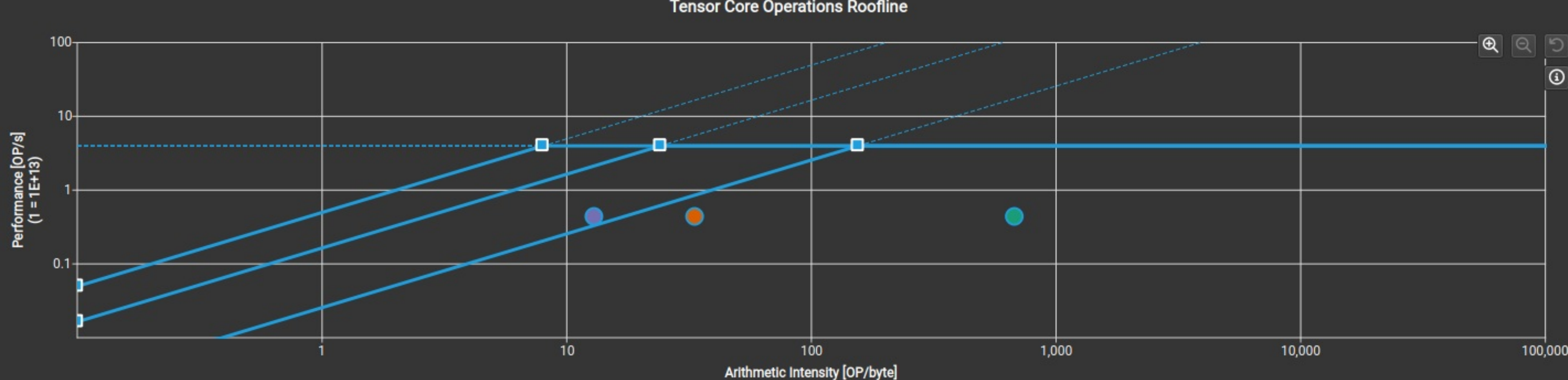
Compute (SM) Throughput [%]	38.94	Duration [ms]	4.04
Memory Throughput [%]	69.96	Elapsed Cycles [cycle]	6,539,972
L1/TEX Cache Throughput [%]	70.81	SM Active Cycles [cycle]	6,461,225.79
L2 Cache Throughput [%]	12.22	SM Frequency [Ghz]	1.62
DRAM Throughput [%]	2.44	DRAM Frequency [Ghz]	7.99

High Memory Throughput Memory is more heavily utilized than Compute: Look at the Memory Workload Analysis section to identify the L1 bottleneck. 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 (re)compute.

Key Performance Indicators

Roofline Analysis The ratio of peak float (FP32) to double (FP64) performance on this device is 64:1. The workload achieved 0% of this device's FP32 peak performance and 0% of its FP64 peak performance. See the Profiling Guide for more details on roofline analysis.

Tensor Core Operations Roofline

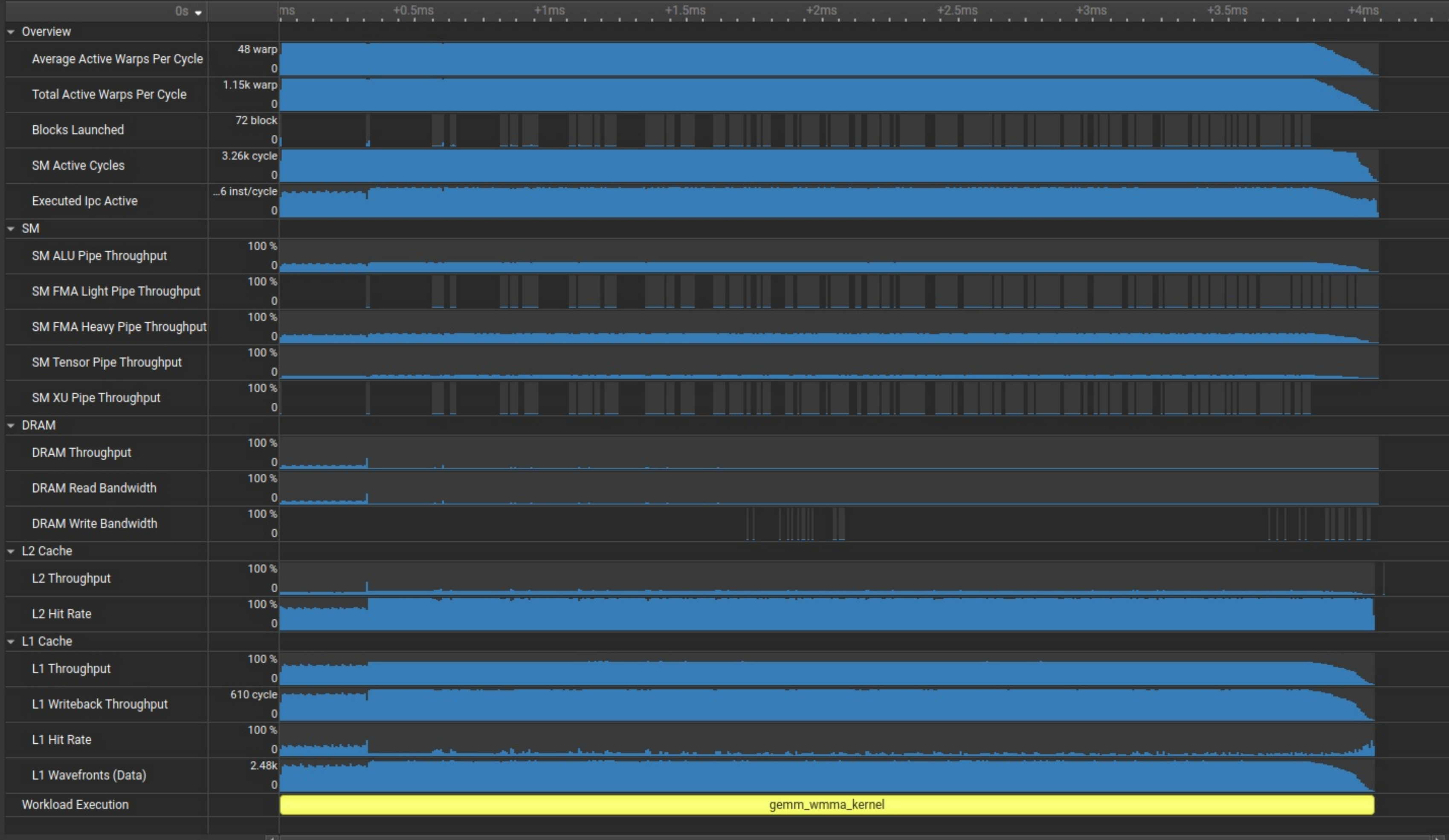


	# Operations	# Operations / Cycle	# Operations / s	Peak %	Peak Operations / Cycle	Peak Operations / s
Src:bf16 Dst:fp32 Sparsity:off	0	0	0	0	12,288	19,905,801,409,332.67
Src:bf16 Dst:fp32 Sparsity:on	0	0	0	0	24,576	39,811,602,818,665.34
Src:fp16 Dst:fp16 Sparsity:off	17,179,869,184	2,626.96	4,255,510,205,375.76	10.69	24,576	39,811,602,818,665.34
Src:fp16 Dst:fp16 Sparsity:on	0	0	0	0	49,152	79,623,205,637,330.67
Src:fp16 Dst:fp32 Sparsity:off	0	0	0	0	12,288	19,905,801,409,332.67
Src:fp16 Dst:fp32 Sparsity:on	0	0	0	0	24,576	39,811,602,818,665.34
Src:fp64	0	0	0	0	86.54	140,181,700,065.72
Src:fp8 Sparsity:off	0	0	0	0	49,152	79,623,205,637,330.67
Src:fp8 Sparsity:on	0	0	0	0	98,304	159,246,411,274,661...
Src:int1	0	0	0	0	393,216	636,985,645,098,645...
Src:int4 Sparsity:off	0	0	0	0	98,304	159,246,411,274,661...
Src:int4 Sparsity:on	0	0	0	0	196,608	318,492,822,549,322...
Src:int8 Sparsity:off	0	0	0	0	49,152	79,623,205,637,330.67
Src:int8 Sparsity:on	0	0	0	0	98,304	159,246,411,274,661...
Src:tf32 Dst:fp32 Sparsity:off	0	0	0	0	6,144	9,952,900,704,666.33
Src:tf32 Dst:fp32 Sparsity:on	0	0	0	0	12,288	19,905,801,409,332.67

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]	2	# Pass Groups	2
Maximum Buffer Size [Mbyte]	7.54	-	-



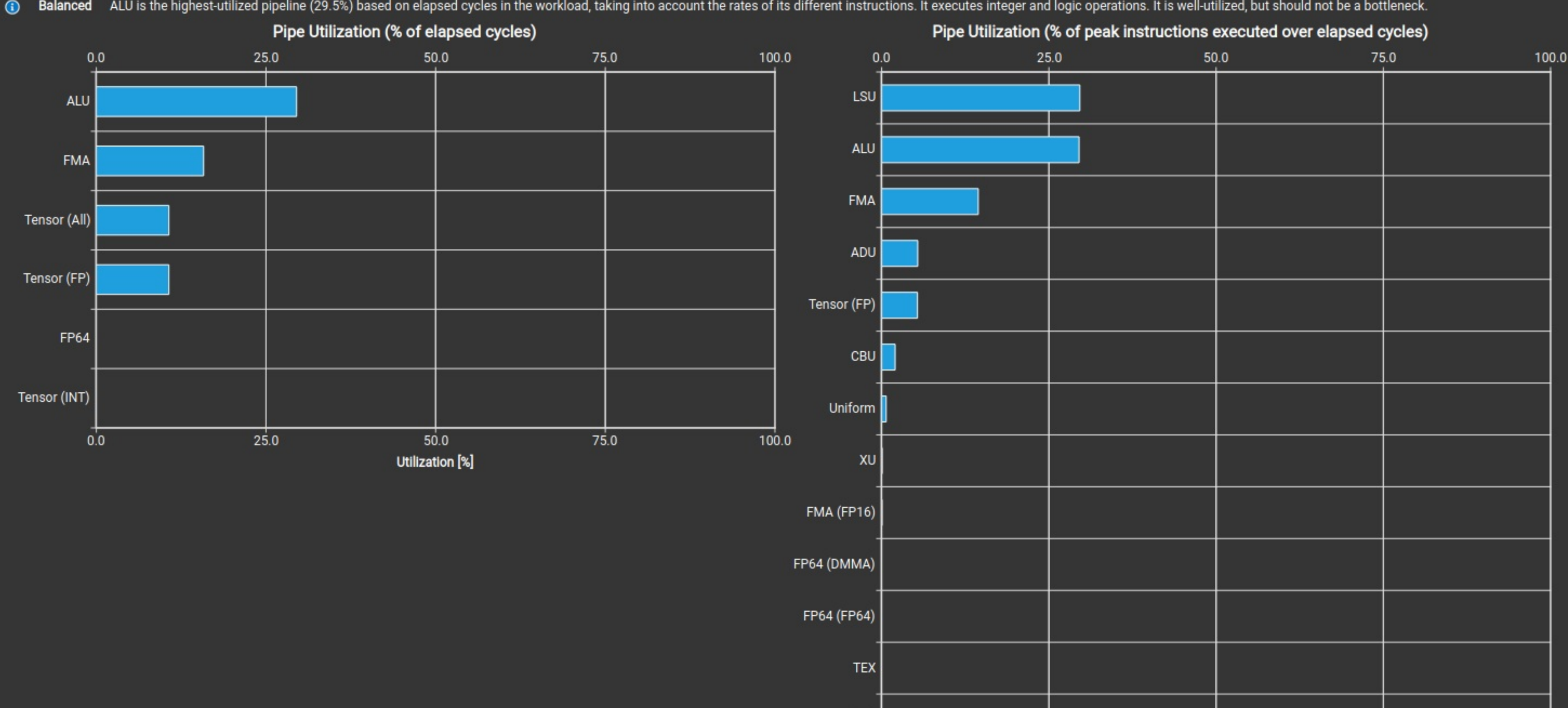
Compute Workload Analysis

Pipe Utilization (Elapsed Cycles)

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]	1.56	SM Busy [%]	38.94
Executed Ipc Active [inst/cycle]	1.58	Issue Slots Busy [%]	38.94
Issued Ipc Active [inst/cycle]	1.58		

Balanced ALU is the highest-utilized pipeline (29.5%) based on elapsed cycles in the workload, taking into account the rates of its different instructions. It executes integer and logic operations. It is well-utilized, but should not be a bottleneck.



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]	6.24	Mem Busy [%]	69.96
L1/TEX Hit Rate [%]	12.69	Max Bandwidth [%]	29.62
L2 Hit Rate [%]	97.43	Mem Pipes Busy [%]	29.62
L2 Compression Input Sectors [sector]	0	Local Memory Spilling Requests	0
L2 Compression Ratio	0	Local Memory Spilling Request Overhead [%]	0
L2 Compression Success Rate [%]	0	L2 Persisting Size [Mbyte]	6.29

L1TEX Global Load Access Pattern
Est. Speedup: 1.06%
The memory access pattern for global loads from L1TEX might not be optimal. On average, only 31.5 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.

Key Performance Indicators

L1TEX Global Store Access Pattern
Est. Speedup: 34.98%
The memory access pattern for global stores to L1TEX might not be optimal. On average, only 16.0 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.

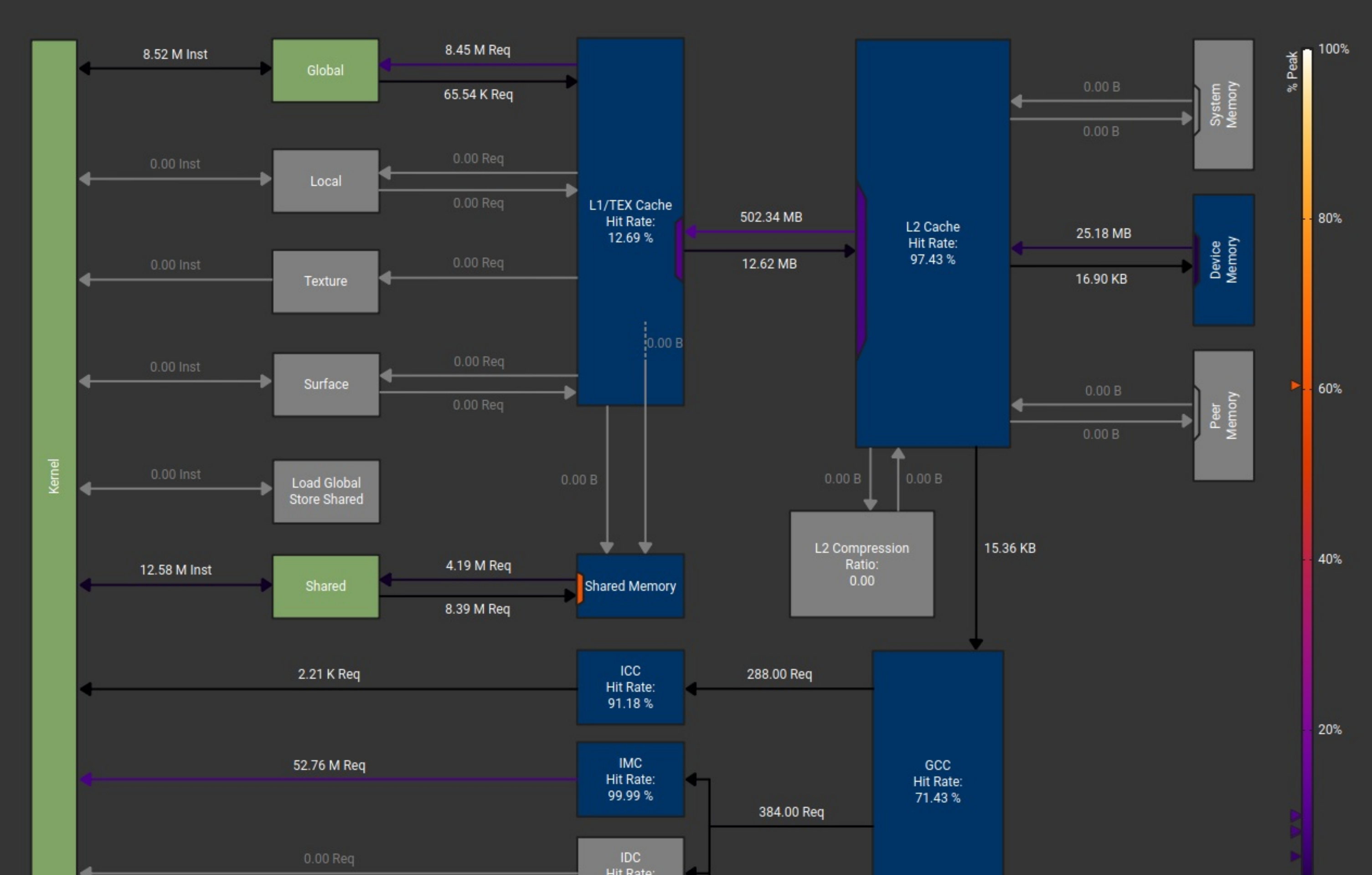
Key Performance Indicators

Shared Load Bank Conflicts
Est. Speedup: 56.44%
The memory access pattern for shared loads might not be optimal and causes on average a 20.1 - way bank conflict across all 4194304 shared load requests.This results in 67109702 bank conflicts, which represent 79.70% of the overall 84200519 wavefronts for shared loads. Check the Source Counters section for uncoalesced shared loads.

Key Performance Indicators

Memory Chart

Values: Transfer Size Inactivity: Greyed Out



Scheduler Statistics

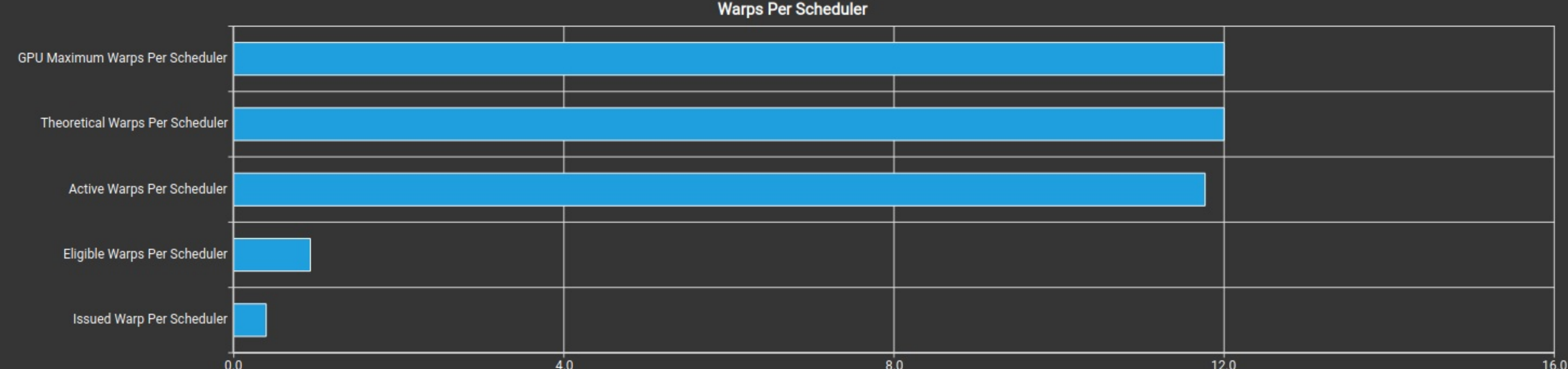
Summary of the activity of the schedulers issuing instructions. Each scheduler maintains a pool of warps that it can issue instructions for. The upper bound of warps in the pool (Theoretical Warps) is limited by the launch configuration. On every cycle each scheduler checks the state of the allocated 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]	11.77	No Eligible [%]	60.71
Eligible Warps Per Scheduler [warp]	0.93	One or More Eligible [%]	39.29
Issued Warp Per Scheduler	0.39		

Issue Slot Utilization
Est. Local Speedup: 30.04%
Every scheduler is capable of issuing one instruction per cycle, but for this workload each scheduler only issues an instruction every 2.5 cycles. This might leave hardware resources underutilized and may lead to less optimal performance. Out of the maximum of 12 warps per scheduler, this workload allocates an average of 11.77 active warps per scheduler, but only an average of 0.93 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.

Key Performance Indicators

Warps Per Scheduler



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.

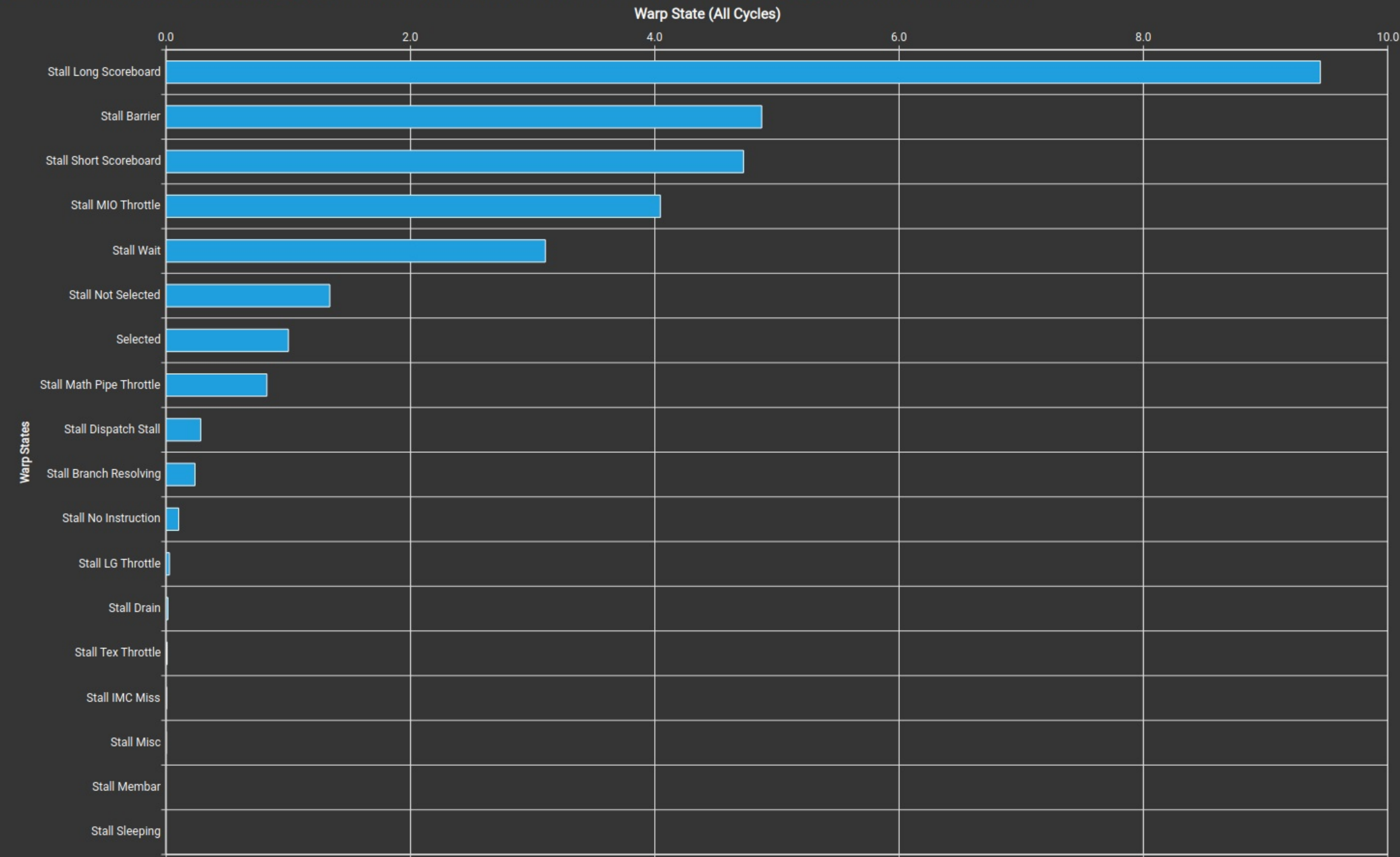
Warp Cycles Per Issued Instruction [cycle]	29.96	Avg. Active Threads Per Warp	32
Warp Cycles Per Executed Instruction [cycle]	29.96	Avg. Not Predicated Off Threads Per Warp	29.52

Long Scoreboard Stalls
Est. Speedup: 30.04%
On average, each warp of this workload spends 9.4 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 31.5% of the total average of 30.0 cycles between issuing two instructions.

Key Performance Indicators

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

Warp State (All Cycles)



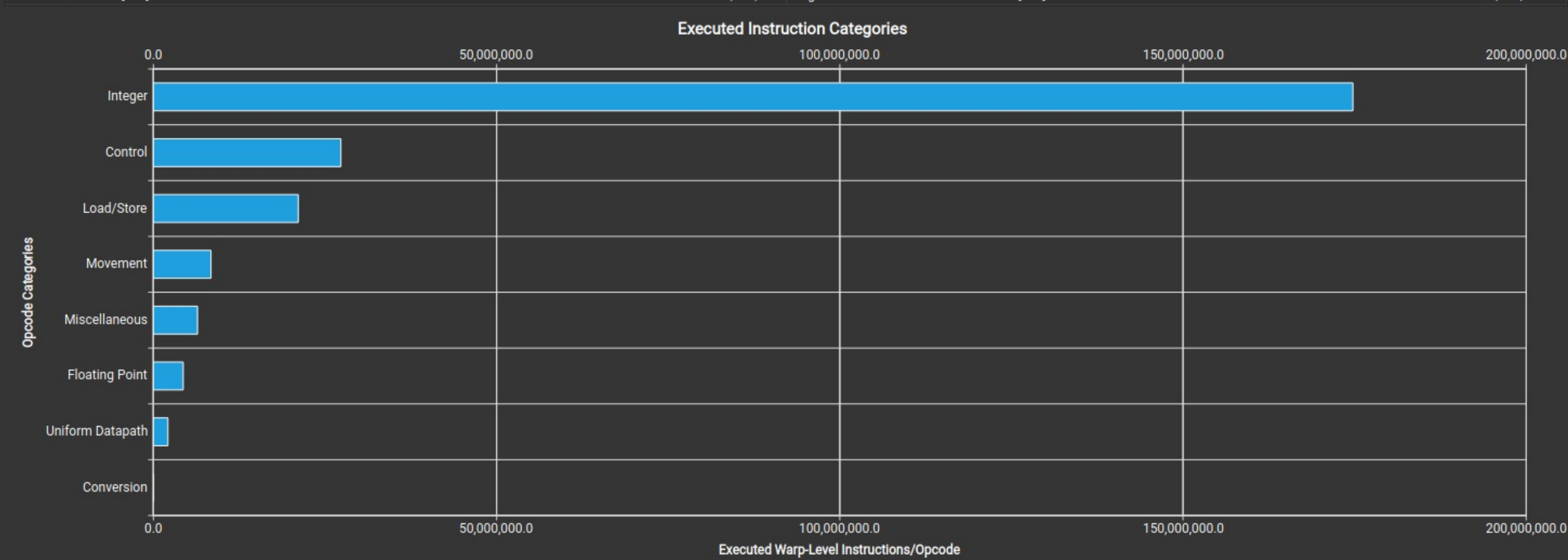
Instruction Statistics

Opcode Category Chart

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]	244,465,664	Avg. Executed Instructions Per Scheduler [inst]	2,546,517.33
Issued Instructions [inst]	244,474,241	Avg. Issued Instructions Per Scheduler [inst]	2,546,606.68

Executed Instruction Categories



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

GPU ID	GPU Name	NUMA ID by CPU Affinity	CPU Affinity	NUMA ID by Memory Affinity
0	NVIDIA GeForce RTX 4060 Laptop GPU	0	0-15	0

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	1,024	Function Cache Configuration	CachePreferNone
Registers Per Thread [register/thread]	24	Static Shared Memory Per Block [Kbyte/block]	4.10
Block Size	512	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	524,288	Driver Shared Memory Per Block [Kbyte/block]	1.02
Waves Per SM	14.22	Shared Memory Configuration Size [Kbyte]	32.77
Uses Green Context	0	Stack Size	1,024
# SMs [SM]	24	# TPCs	12
Enabled TPC IDs	all	-	-

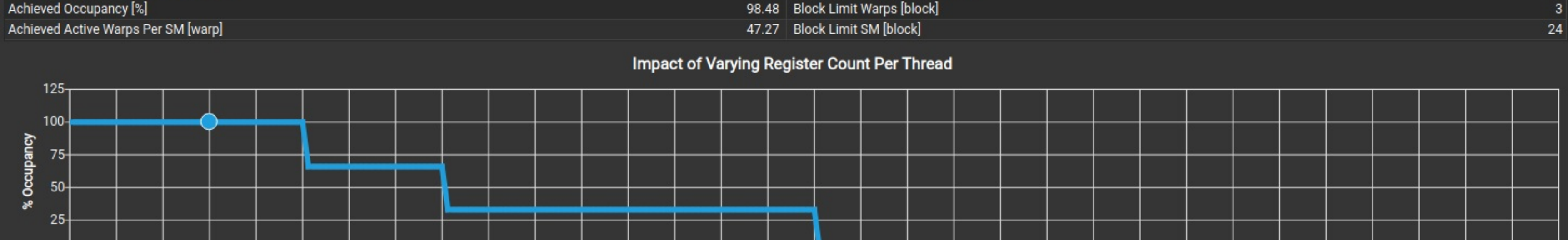
Occupancy

% Occupancy Graphs

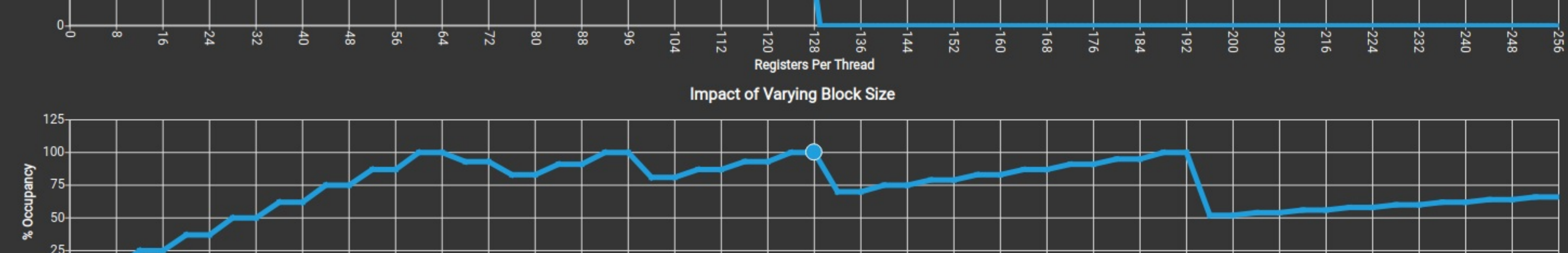
Occupancy is the ratio of the number of active warps per multiprocessor to the maximum number of possible active warps. Another way to view occupancy is the percentage of the hardware's ability to process warps that is actively in use. Higher occupancy does not always result in higher performance, however, 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]	5
Theoretical Active Warps per SM [warp]	48	Block Limit Shared Mem [block]	6
Achieved Occupancy [%]	98.48	Block Limit Warps [block]	3
Achieved Active Warps Per SM [warp]	47.27	Block Limit SM [block]	24

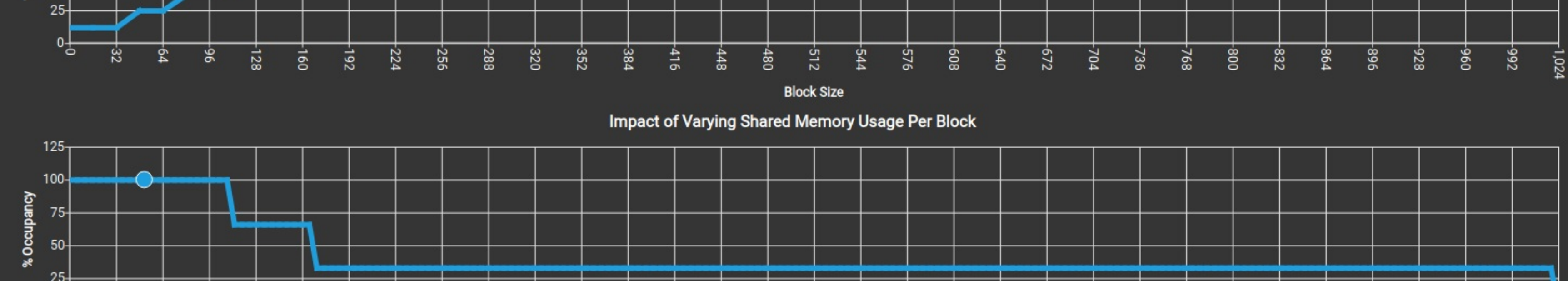
Impact of Varying Register Count Per Thread



Impact of Varying Block Size



Impact of Varying Shared Memory Usage Per Block



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]	6,461,225.79	Average L1 Active Cycles [cycle]	6,461,225.79
Average L2 Active Cycles [cycle]	6,078,539.75	Average SMSP Active Cycles [cycle]	6,481,626.09
Average DRAM Active Cycles [cycle]	787,248	Total SM Elapsed Cycles [cycle]	156,956,000
Total L1 Elapsed Cycles [cycle]	156,956,000	Total L2 Elapsed Cycles [cycle]	103,658,608
Total SMSP Elapsed Cycles [cycle]	627,824,000	Total DRAM Elapsed Cycles [cycle]	129,073,152

Workload Distribution

	Average	Min	Max	Sum
SM Active Cycles	6,461,225.79	6,418,869	6,538,015	155,069,419
SMSP Active Cycles	6,481,626.09	6,429,879	6,560,102	622,236,105
L1 Active Cycles	6,461,225.79	6,418,869	6,538,015	155,069,419
L2 Active Cycles	6,078,539.75	6,073,080	6,082,545	97,256,636
DRAM Active Cycles	787,248	786,976	787,536	3,148,992

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]	23,117,824	Branch Efficiency [%]	100
Branch Instructions Ratio [%]	0.09	Avg. Divergent Branches [branches]	0

Uncoalesced Global Accesses
Est. Speedup: 2.76%
This kernel has uncoalesced global accesses resulting in a total of 524288 excessive sectors (3% of the total 17825792 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.

Key Performance Indicators

L2 Theoretical Sectors Global Excessive

Location	Value	Value (%)
0x79999d272c10 in gemm_wmma_kernel	65,536	13
0x79999d272c00 in gemm_wmma_kernel	65,536	13
0x79999d272be0 in gemm_wmma_kernel	65,536	13
0x79999d272bb0 in gemm_wmma_kernel	65,536	13
0x79999d272b50 in gemm_wmma_kernel	65,536	13

Uncoalesced Shared Accesses
Est. Speedup: 71.85%
This kernel has uncoalesced shared accesses resulting in a total of 67108864 excessive wavefronts (73% of the total 92274688 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.

Key Performance Indicators

L1 Wavefronts Shared Excessive

Location	Value	Value (%)
0x79999d272960 in gemm_wmma_kernel	58,720,256	88
0x79999d272950 in gemm_wmma_kernel	8,388,608	13
0x79999d272840 in gemm_wmma_kernel	0	0
0x79999d2726d0 in gemm_wmma_kernel	0	0

Warp Stall Sampling (All Samples)

Location	Value	Value (%)
0x79999d2726d0 in gemm_wmma_ke...	27,474	18
0x79999d272840 in gemm_wmma_ke...	24,022	16
0x79999d2729e0 in gemm_wmma_ke...	16,237	11
0x79999d272970 in gemm_wmma_ke...	15,856	10
0x79999d272880 in gemm_wmma_ke...	9,482	6

Most Instructions Executed

Location	Value	Value (%)
0x79999d272850 in gemm_wmma_ke...	4,194,304	2
0x79999d272840 in gemm_wmma_ke...	4,194,304	2
0x79999d272830 in gemm_wmma_ke...	4,194,304	2
0x79999d272820 in gemm_wmma_ke...	4,194,304	2
0x79999d272810 in gemm_wmma_ke...	4,194,304	2

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.