Current	Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
	536 - applyMultiBandGainKernel	(103407, 1, 1)x(256, 1, 1)	7,02 ms	4.109.385	0 - Tesla T4	585,49 Mhz	[4361] CudaEQ	

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 [%]	33,22	Duration [ms]	7,02
Memory Throughput [%]	85,91	Elapsed Cycles [cycle]	4.109.385
L1/TEX Cache Throughput [%]	37,95	SM Active Cycles [cycle]	4.102.952,98
L2 Cache Throughput [%]	27,55	SM Frequency [Mhz]	585,49
DRAM Throughput [%]	85,91	DRAM Frequency [Ghz]	5,00

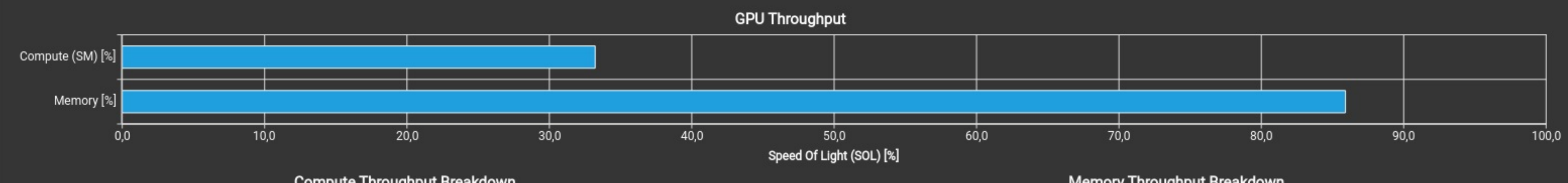
High Throughput The kernel is utilizing greater than 80.0% of the available compute or memory performance of the device. To further improve performance, work will likely need to be shifted from the most utilized to another unit. Start by analyzing DRAM in the Memory Workload Analysis section.

Compute Bottleneck Detect bottlenecks arising from compute capabilities — Apply

Memory Bottleneck Detect bottlenecks arising from memory capabilities — Apply

Roofline Analysis The ratio of peak float (fp32) to double (fp64) performance on this device is 32:1. The kernel achieved 1% 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.

GPU Throughput



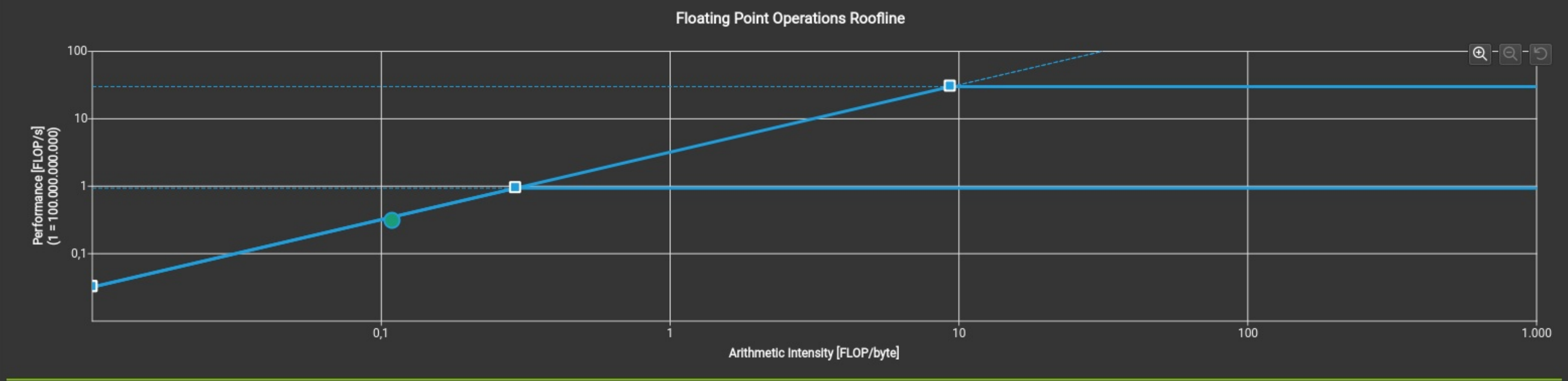
Compute Throughput Breakdown

SM: Inst Executed Pipe Lsu [%]	33,22
SM: Issue Active [%]	15,98
SM: Inst Executed [%]	15,98
SM: Mio Inst Issued [%]	13,09
SM: Mio Pq Write Cycles Active [%]	10,07
SM: Mio Pq Read Cycles Active [%]	10,07
SM: Pipe Alu Cycles Active [%]	9,31
SM: Pipe Fma Cycles Active [%]	8,05
SM: Inst Executed Pipe Adu [%]	6,04
SM: Mio2rf Writeback Active [%]	5,54
SM: Inst Executed Pipe Cbu Pred On Any [%]	5,05
IDC: Request Cycles Active [%]	4,03
SM: Inst Executed Pipe Xu [%]	3,02
SM: Inst Executed Pipe Uniform [%]	0,50
SM: Pipe Tensor Cycles Active [%]	0
SM: Pipe Shared Cycles Active [%]	0
SM: Pipe Fp64 Cycles Active [%]	0
SM: Inst Executed Pipe Tex [%]	0
SM: Inst Executed Pipe Ipa [%]	0
SM: Inst Executed Pipe Fp16 [%]	0

Memory Throughput Breakdown

DRAM: Cycles Active [%]	85,91
DRAM: Dram Sectors [%]	62,75
L1: Lsuin Requests [%]	33,22
L2: T Sectors [%]	27,55
L1: M L1tex2xbar Req Cycles Active [%]	18,98
L1: Data Pipe Lsu Wavefronts [%]	17,99
L2: Xbar2lts Cycles Active [%]	16,94
L1: M Xbar2l1tex Read Sectors [%]	16,10
L2: Lts2xbar Cycles Active [%]	13,78
L2: D Sectors Fill Device [%]	13,77
L2: D Sectors [%]	13,77
L1: Lsu Writeback Active [%]	8,56
L2: T Tag Requests [%]	6,89
L1: Data Bank Writes [%]	6,04
L1: Data Bank Reads [%]	6,04
L1: Texin Sm2tex Req Cycles Active [%]	0,06
L1: F Wavefronts [%]	0,00
L1: Data Pipe Tex Wavefronts [%]	0
L1: Tex Writeback Active [%]	0
L2: D Atomic Input Cycles Active [%]	0
L2: D Sectors Fill Sysmem [%]	0

Floating Point Operations Roofline



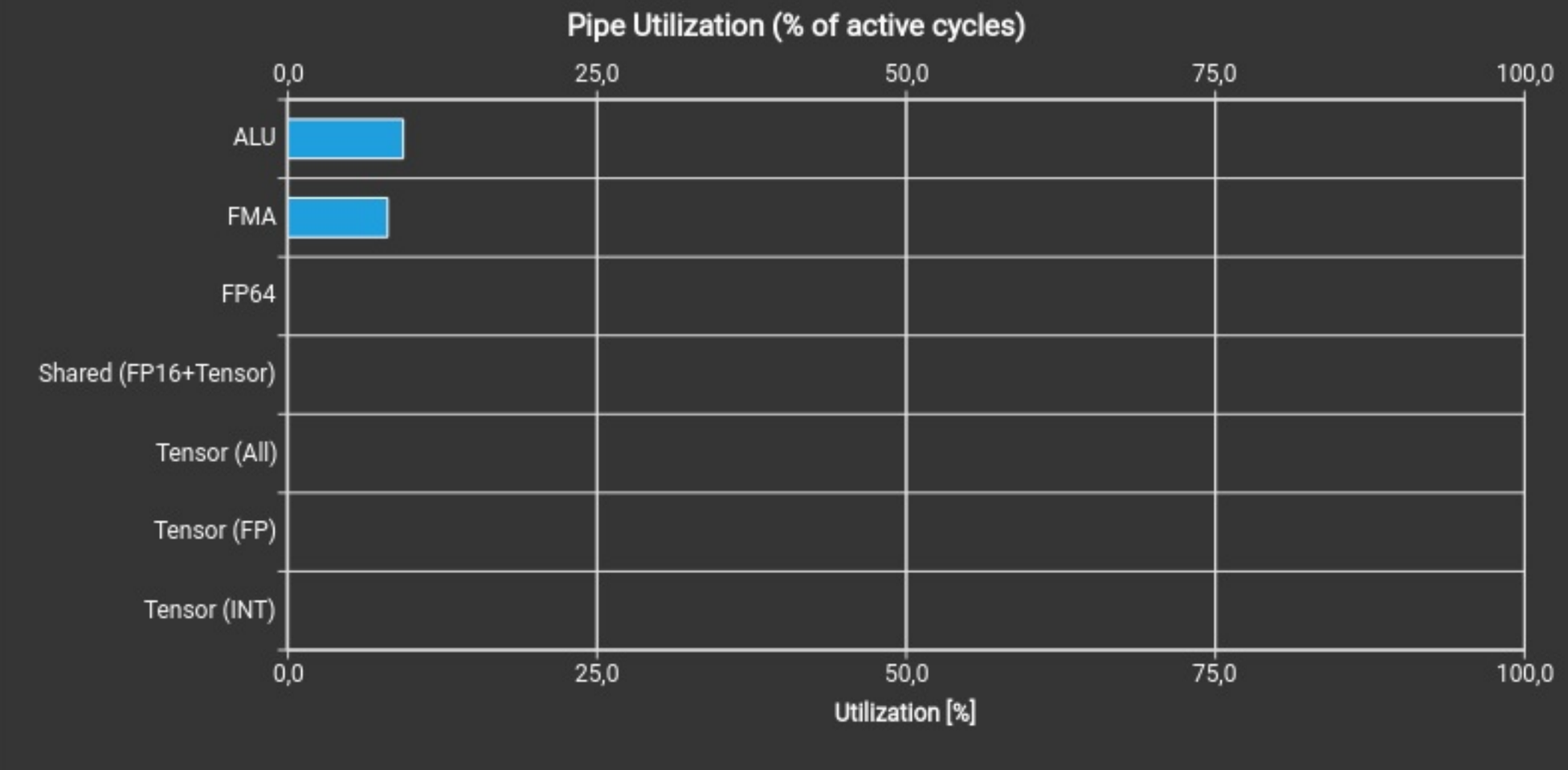
Compute Workload Analysis

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.

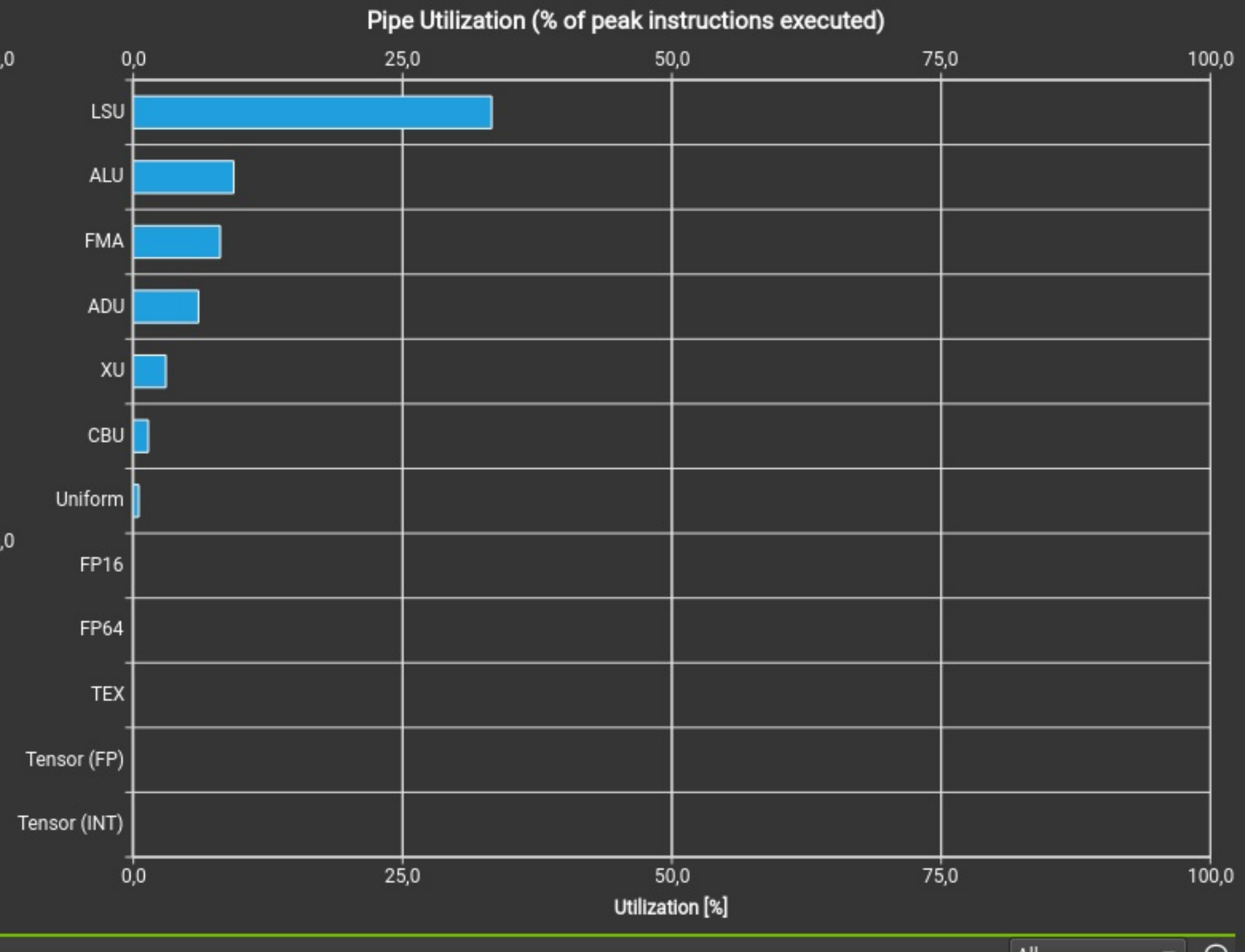
Executed Ipc Elapsed [inst/cycle]	0,64	SM Busy [%]	16,01
Executed Ipc Active [inst/cycle]	0,64	Issue Slots Busy [%]	16,01
Issued Ipc Active [inst/cycle]	0,64		

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

Pipe Utilization (% of active cycles)



Pipe Utilization (% of peak instructions executed)



Memory Workload Analysis

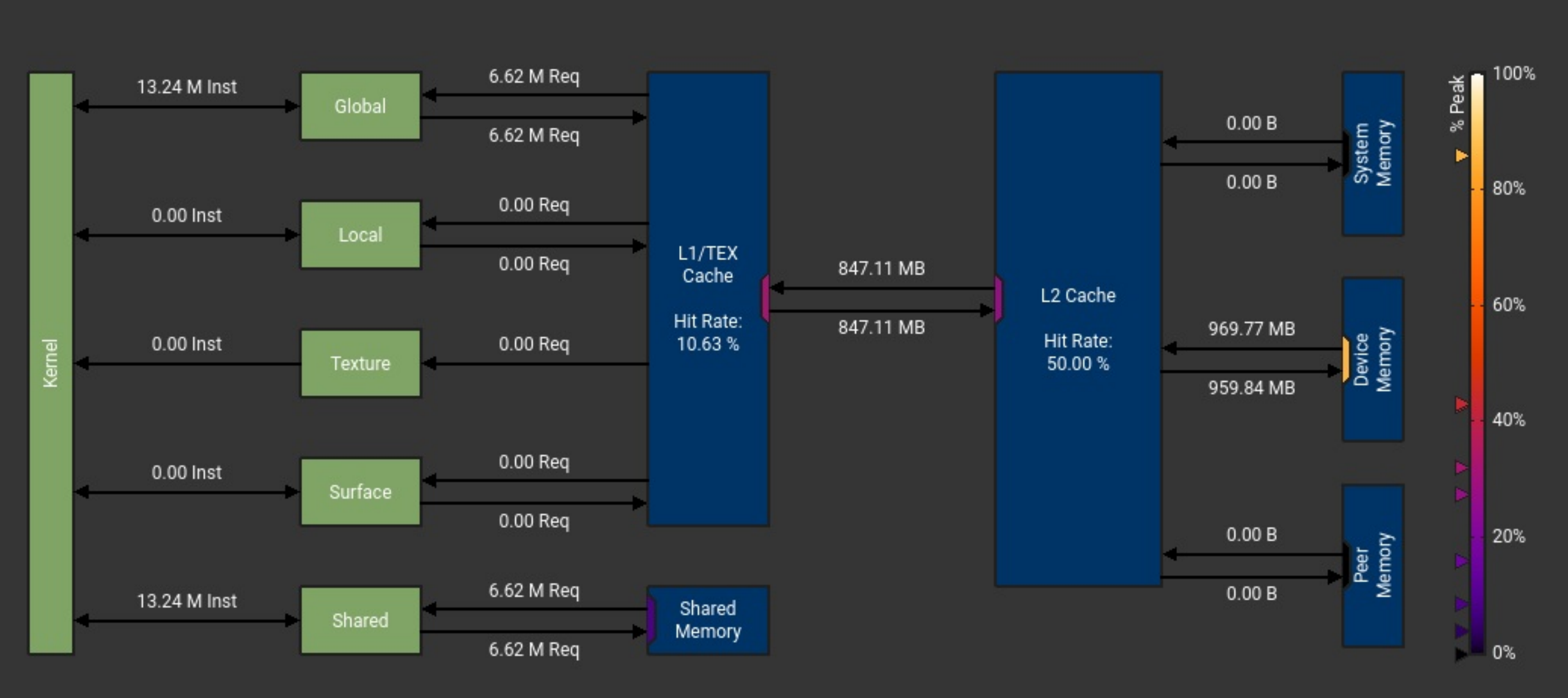
All

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]	274,93	Mem Busy [%]	27,55
L1/TEX Hit Rate [%]	10,63	Max Bandwidth [%]	85,91
L2 Hit Rate [%]	50,00	Mem Pipes Busy [%]	33,22

Memory Chart

Show As: Transfer Size



Shared Memory

	Instructions	Requests	Wavefronts	% Peak	Bank Conflicts
Shared Load	6.618.048	6.618.048	6.618.048	4,03	0
Shared Load Matrix	0	0			
Shared Store	6.618.048	6.618.048	6.755.340	4,11	0
Shared Atomic	0	0	0	0	0
Other	-	-	930.309	0,57	0
Total	13.236.096	13.236.096	14.303.697	8,70	0

L1/TEX Cache

	Instructions	Requests	Wavefronts	% Peak	Sectors	Sectors/Req	Hit Rate	Bytes	Sector Misses to L2	% Peak to L2	Returns to SM
Global Load	6.618.048	6.618.048	6.618.048	4,03	26.472.190	4,00	0	847.110.080	26.472.190	16,10	6.618.048
Surface Load	0	0	0	0	0	0	0	0	0	0	0
Texture Load	0	0	0	0	0	0	0	0	0	0	
Global Store	6.618.048	6.618.048	6.618.048	4,03	26.472.190	4,00	21,45	847.110.080	26.472.190	16,10	-
Local Store	0	0	0	0	0	0	0	0			-
Surface Store	0	0	0	0	0	0	0	0	0	0	-
Global Reduction	0	0	0	0	0	0	0	0	0	0	-
DSMEM Reduction						-	-		0	0	-
Surface Reduction	0	0	0	0	0	0	0	0	0	0	-
Global Atomic ALU	0	0	0	0	0	0	0	0	0	0	see above
Global Atomic CAS	0										
Surface Atomic ALU	0	0	0	0	0	0	0	0	0	0	see above
Surface Atomic CAS	0										
Loads	6.618.048	6.618.048	6.618.048	4,03	26.472.190	4,00	0	847.110.080	26.472.190	16,10	6.618.048
Stores	6.618.048	6.618.048	6.618.048	4,03	26.472.190	4,00	21,45	847.110.080	26.472.190	16,10	-
Atomics & Reductions	0	0	0	0	0	0	0	0	0	0	-
Total	13.236.096	13.236.096	13.236.096	8,05	52.944.380	4,00	10,73	1.694.220.160	52.944.380	32,21	6.618.048

L2 Cache

	Requests	Sectors	Sectors/Req	% Peak	Hit Rate	Bytes	Throughput	Sector Misses to Device	Sector Misses to System	Sector Misses to Peer
L1/TEX Load	6.618.048	26.472.190	4,00	13,77	0	847.110.080	120.694.058.805,56	26.472.192	0	0
L1/TEX Store	6.618.048	26.472.190	4,00	13,77	100	847.110.080	120.694.058.805,56	0	0	0
L1/TEX Atomic ALU	0	0	0	0	0	0	0	0	0	0
L1/TEX Atomic CAS	0	0	0	0	0	0	0	0	0	0
L1/TEX Reduction	0	0	0	0	0	0	0	0	0	0
L1/TEX Total	13.236.096	52.944.380	4,00	27,55	50,00	1.694.220.160	241.388.117.611,12	26.472.192	0	0
ECC Total	-	323	-	0,00	-	10.336	1.472.646,61	323	-	-
GPU Total	13.237.416	52.946.076	4,00	27,55	50,00	1.694.274.432	241.395.850.145,67	26.472.212	0	0

Device Memory

	Sectors	% Peak	Bytes	Throughput
Load	30.305.164	43,18	969.765.248	138.169.650.713,75
Store	29.995.152	42,73	959.844.864	136.756.219.994,26
Total	60.300.316	85,91	1.929.610.112	274.925.870.708,01

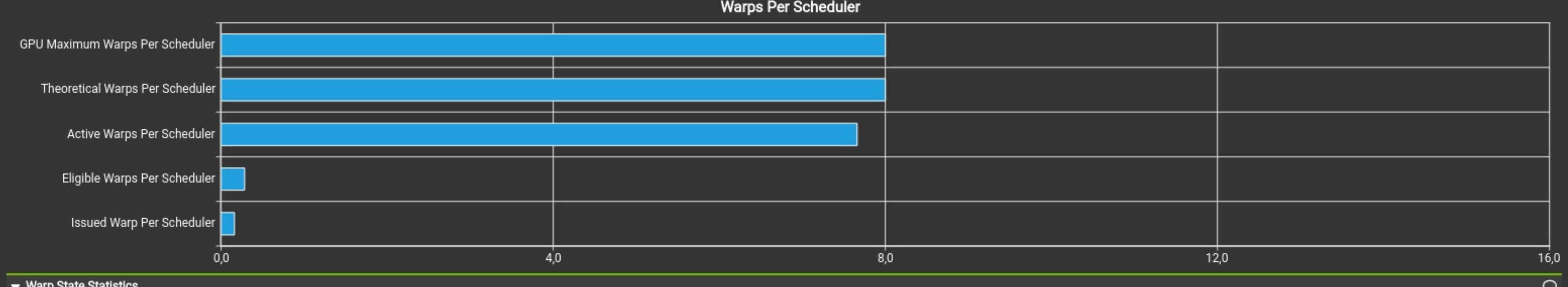
Scheduler Statistics

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]	7,66	No Eligible [%]	83,99
Eligible Warps Per Scheduler [warp]	0,28	One or More Eligible [%]	16,01
Issued Warp Per Scheduler	0,16		

Issue Slot Utilization
Est. Local Speedup: 83.99% Every scheduler is capable of issuing one instruction per cycle, but for this kernel each scheduler only issues an instruction every 6.2 cycles. This might leave hardware resources underutilized and may lead to less optimal performance. Out of the maximum of 8 warps per scheduler, this kernel allocates an average of 7.66 active warps per scheduler, but only an average of 0.28 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.

Warps Per Scheduler



Warp State Statistics

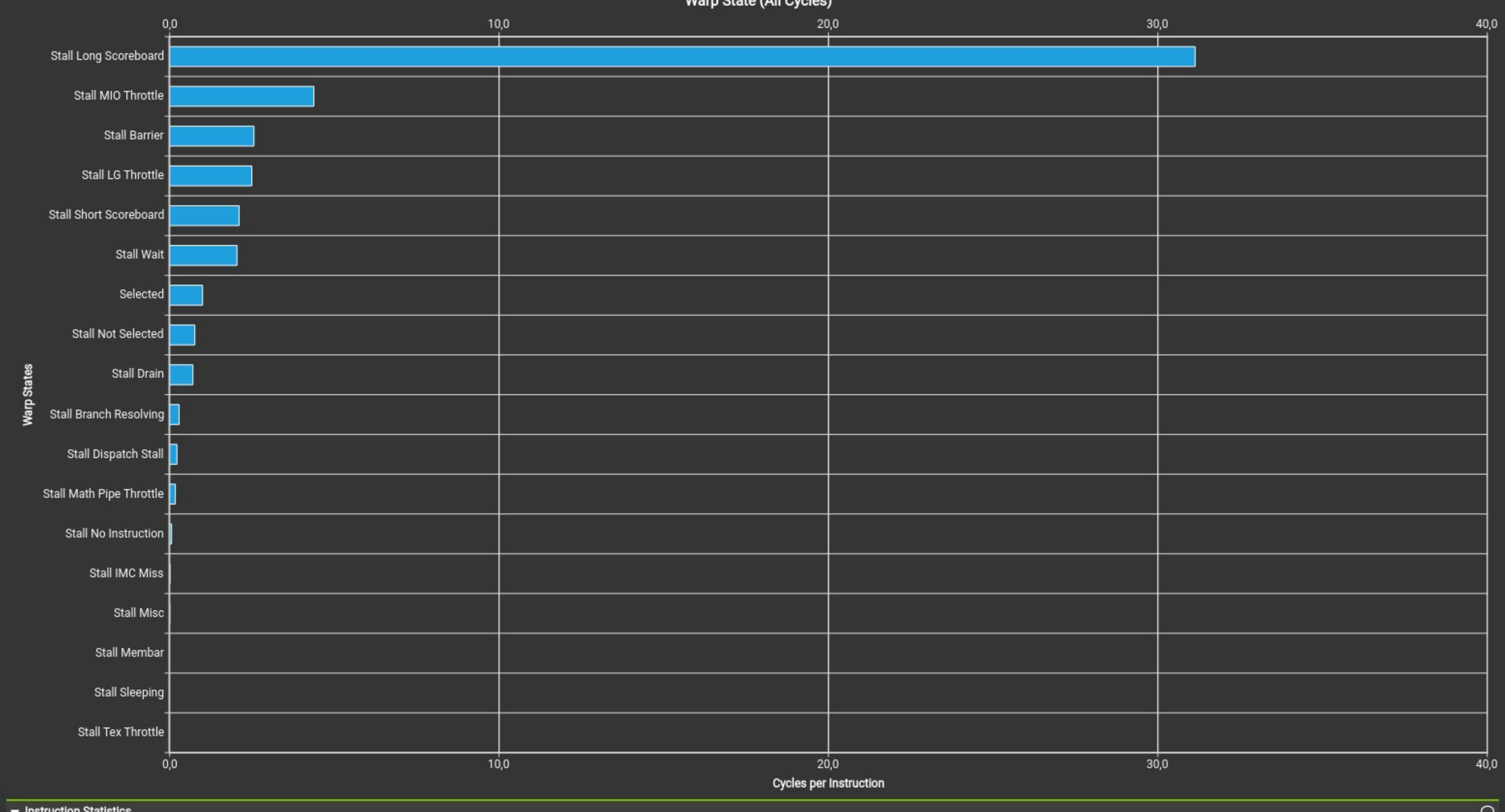
Analysis of the states in which all warps spent cycles during the kernel execution. The warp states describe a warp's readiness or inability to issue its next instruction. The warp cycles per instruction define the latency between two consecutive instructions. The higher the value, the more warp parallelism 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]	47,86	Avg. Active Threads Per Warp	32,00
Warp Cycles Per Executed Instruction [cycle]	47,87	Avg. Not Predicated Off Threads Per Warp	29,75

Long Scoreboard
Est. Local Speedup: 65.05% On average, each warp of this kernel spends 31.1 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 65.0% of the total average of 47.9 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.

Warp State (All Cycles)



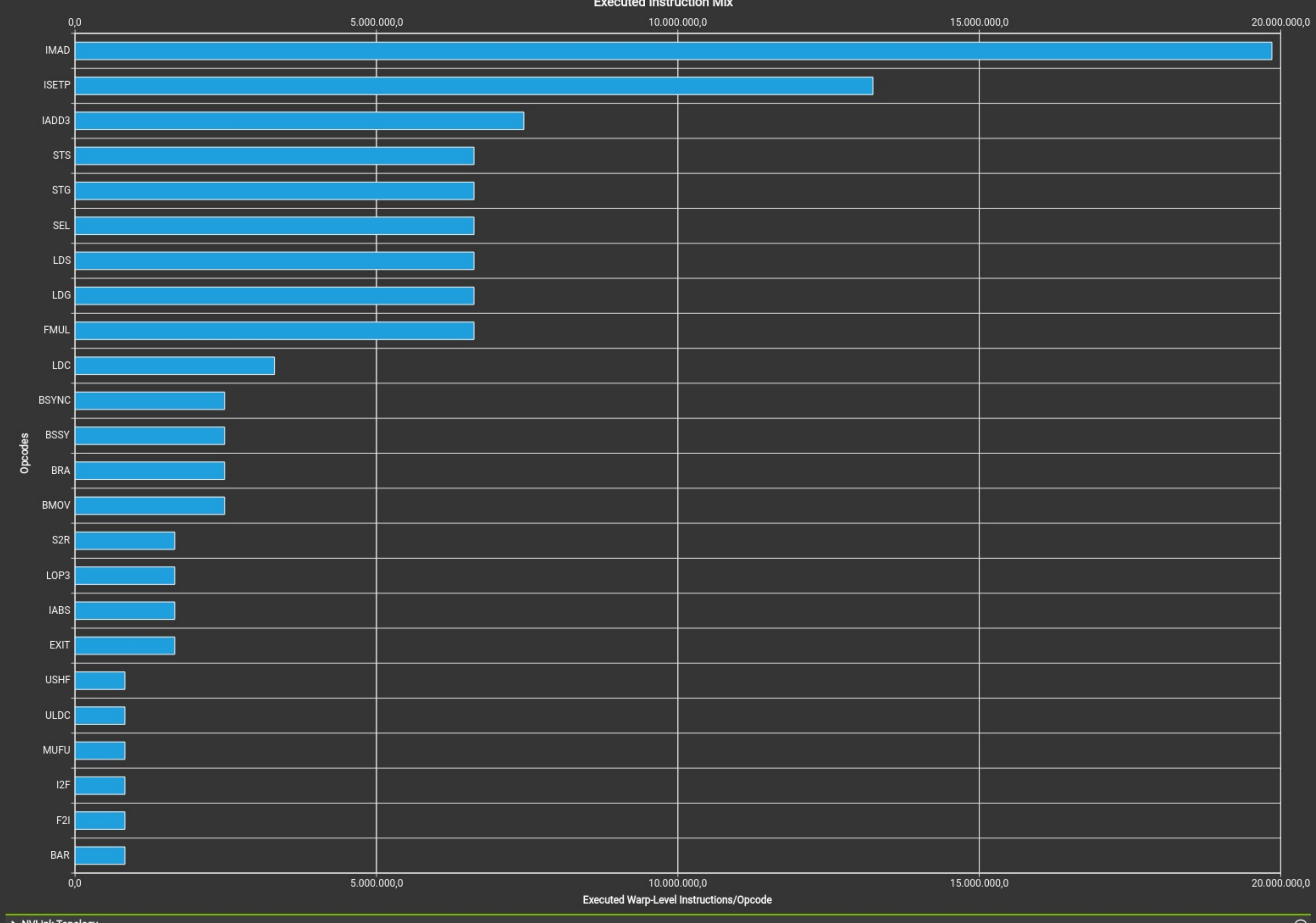
Instruction Statistics

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]	105.061.512	Avg. Executed Instructions Per Scheduler [inst]	656.634,45
Issued Instructions [inst]	105.073.672	Avg. Issued Instructions Per Scheduler [inst]	656.710,45

FP32/64 Instructions
Est. Local Speedup: 50.00% This kernel executes 0 fused and 6618048 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.

Executed Instruction Mix



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	103.407	Function Cache Configuration	CachePreferNone
Registers Per Thread [register/thread]	41	Static Shared Memory Per Block [Kbyte/block]	8,19
Block Size	256	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	26.472.192	Driver Shared Memory Per Block [byte/block]	0
Waves Per SM	646,29	Shared Memory Configuration Size [Kbyte]	32,77

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.

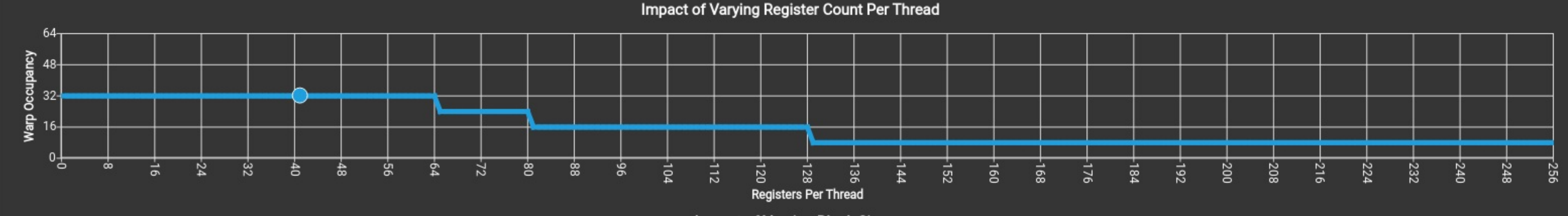
Theoretical Occupancy [%]	100	Block Limit Registers [block]	5
Theoretical Active Warps per SM [warp]	32	Block Limit Shared Mem [block]	4
Achieved Occupancy [%]	96,08	Block Limit Warps [block]	4
Achieved Active Warps Per SM [warp]	30,74	Block Limit SM [block]	16

Occupancy Limiters This kernel's theoretical occupancy is not impacted by any block limit.

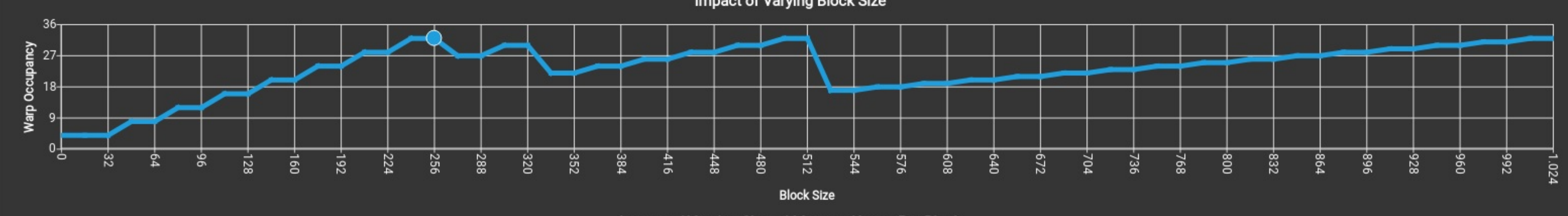
Achieved Occupancy Analysis of the Achieved Occupancy — Apply

Theoretical Occupancy Analysis of Theoretical Occupancy and its Limiters — Apply

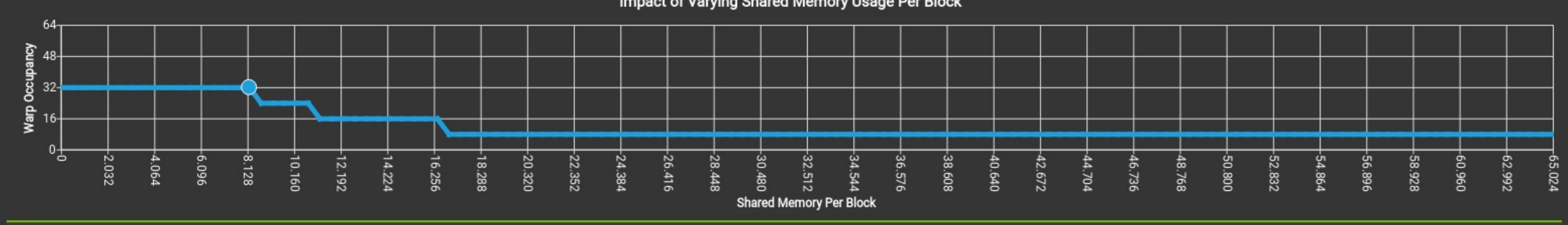
Impact of Varying Register Count Per Thread



Impact of Varying Block Size



Impact of Varying Shared Memory Usage Per Block



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]	6.618.048	Branch Efficiency [%]	100
Branch Instructions Ratio [%]	0,06	Avg. Divergent Branches	0

Warp Stall Sampling (All Samples)

Location	Value	Value (%)
0x7a95230330e0 in applyMultiBandGainKe...	83.858	52
0x7a9523033110 in applyMultiBandGainK...	12.044	8
0x7a9523033170 in applyMultiBandGainK...	9.033	6
0x7a95230330f0 in applyMultiBandGainKe...	5.390	3
0x7a9523032fd0 in applyMultiBandGainKe...	3.513	2

Most Instructions Executed

Location	Value	Value (%)
0x7a95230334e0 in applyMultiBandGainK...	827.256	1
0x7a95230334d0 in applyMultiBandGainK...	827.256	1
0x7a95230334c0 in applyMultiBandGainK...	827.256	1
0x7a95230334b0 in applyMultiBandGainK...	827.256	1
0x7a95230334a0 in applyMultiBandGainK...	827.256	1

To customize your report even further, you might want to learn about custom sections and writing your own rules.
You might also want to consider adding individual metrics.