	Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current	529 - applyMultiBandGainKernel	(6374, 1, 1)x(1024, 1, 1)	2,08 ms	1.214.252	0 - Tesla T4	584,81 Mhz	[4558] CudaEQ3	

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.

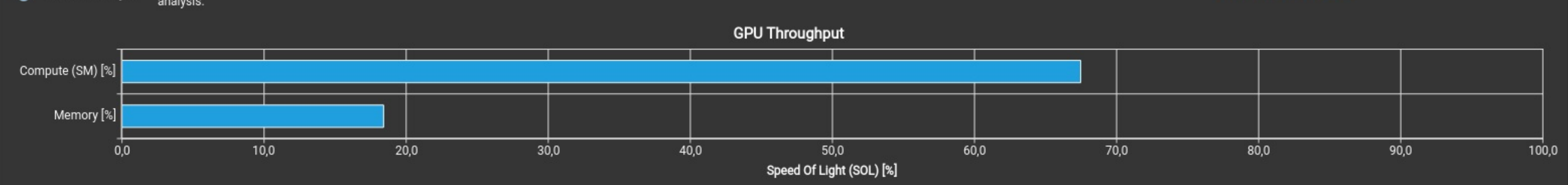
Compute (SM) Throughput [%]	67,48	Duration [ms]	2,08
Memory Throughput [%]	18,41	Elapsed Cycles [cycle]	1.214.252
L1/TEX Cache Throughput [%]	19,09	SM Active Cycles [cycle]	1.182.668,82
L2 Cache Throughput [%]	5,77	SM Frequency [Mhz]	584,81
DRAM Throughput [%]	18,41	DRAM Frequency [Ghz]	4,99

High Compute Throughput Compute is more heavily utilized than Memory: Look at the Compute Workload Analysis section to see what the compute pipelines are spending their time doing. Also, consider whether any computation is redundant and could be reduced or moved to look-up tables.

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 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.

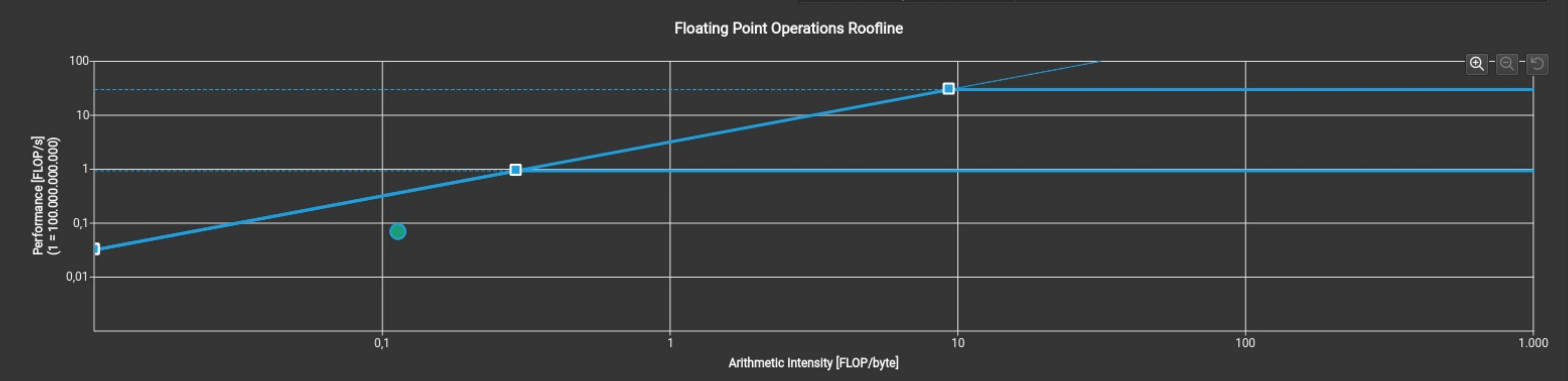


Compute Throughput Breakdown

SM: Pipe Alu Cycles Active [%]	67,48
SM: Issue Active [%]	38,12
SM: Inst Executed [%]	38,12
SM: Inst Executed Pipe Lsu [%]	17,25
SM: Mio Inst Issued [%]	7,99
SM: Inst Executed Pipe Adu [%]	6,73
SM: Mio Pq Write Cycles Active [%]	5,13
SM: Mio Pq Read Cycles Active [%]	5,13
IDC: Request Cycles Active [%]	4,21
SM: Mio2rf Writeback Active [%]	3,85
SM: Inst Executed Pipe Cbu Pred On Any [%]	2,51
SM: Inst Executed Pipe Tex [%]	0,84
SM: Pipe Fma Cycles Active [%]	0,66
SM: Pipe Tensor Cycles Active [%]	0
SM: Pipe Shared Cycles Active [%]	0
SM: Pipe Fp64 Cycles Active [%]	0
SM: Inst Executed Pipe Xu [%]	0
SM: Inst Executed Pipe Uniform [%]	0
SM: Inst Executed Pipe Ipa [%]	0
SM: Inst Executed Pipe Fp16 [%]	0

Memory Throughput Breakdown

DRAM: Cycles Active [%]	18,41
L1: Lsuin Requests [%]	17,25
DRAM: Dram Sectors [%]	13,45
L1: Data Pipe Lsu Wavefronts [%]	9,54
L2: T Sectors [%]	5,77
L1: Lsu Writeback Active [%]	5,18
L1: M L1tex2xbar Req Cycles Active [%]	3,85
L2: Xbar2lts Cycles Active [%]	3,78
L1: M Xbar2l1tex Read Sectors [%]	3,38
L2: Lts2xbar Cycles Active [%]	2,90
L2: D Sectors Fill Device [%]	2,88
L2: D Sectors [%]	2,85
L1: Data Bank Reads [%]	1,74
L1: Data Bank Writes [%]	1,72
L2: T Tag Requests [%]	1,44
L1: Texin Sm2tex Req Cycles Active [%]	0,45
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

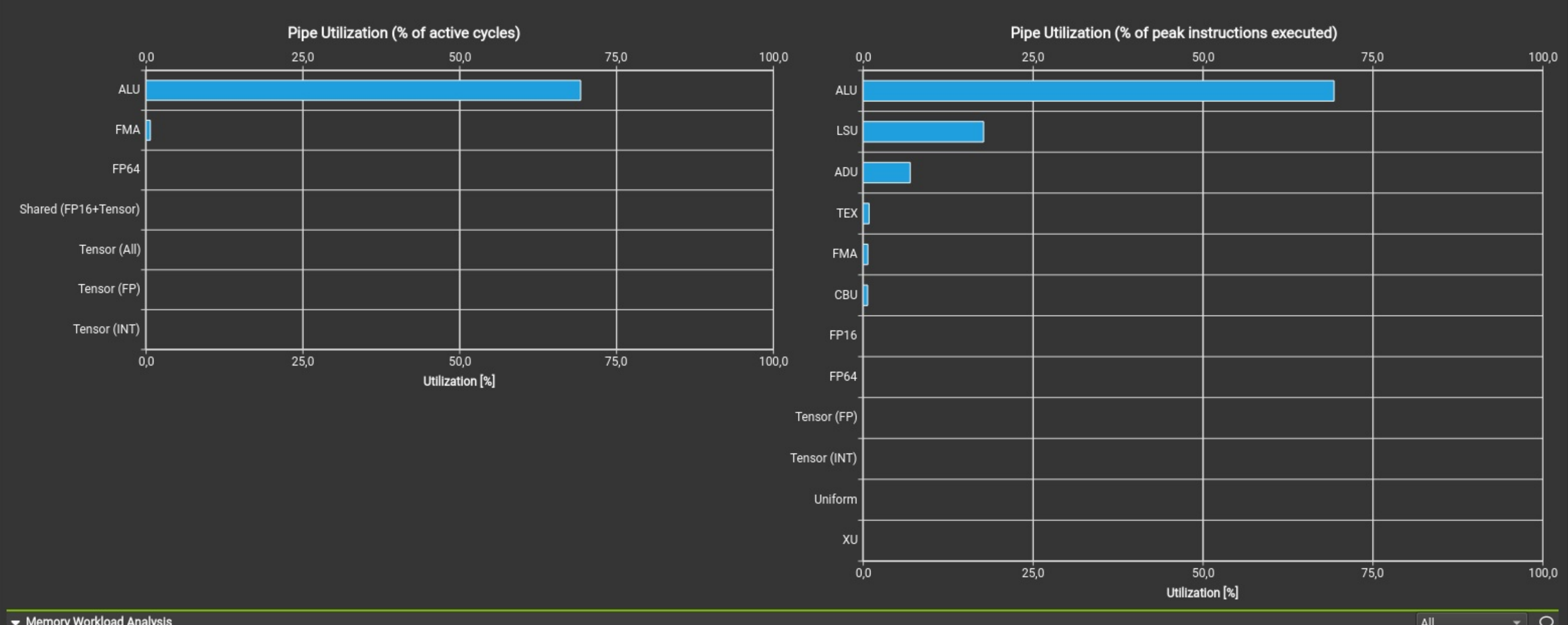


Compute Workload Analysis

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]	1,52	SM Busy [%]	69,28
Executed Ipc Active [inst/cycle]	1,57	Issue Slots Busy [%]	39,14
Issued Ipc Active [inst/cycle]	1,57		

High Utilization ALU is the highest-utilized pipeline (69.3%) based on active cycles, taking into account the rates of its different instructions. It executes integer and logic operations. The pipeline is well-utilized, but might become a bottleneck if more work is added. Based on the number of executed instructions, the highest utilized pipeline (69.3%) is ALU. It executes integer and logic operations. Comparing the two, the overall pipeline utilization appears to be caused by frequent, low-latency instructions. See the Kernel Profiling Guide or hover over the pipeline name to understand the workloads handled by each pipeline. The Instruction Statistics section shows the mix of executed instructions in this kernel. Check the Warp State Statistics section for which reasons cause warps to stall.



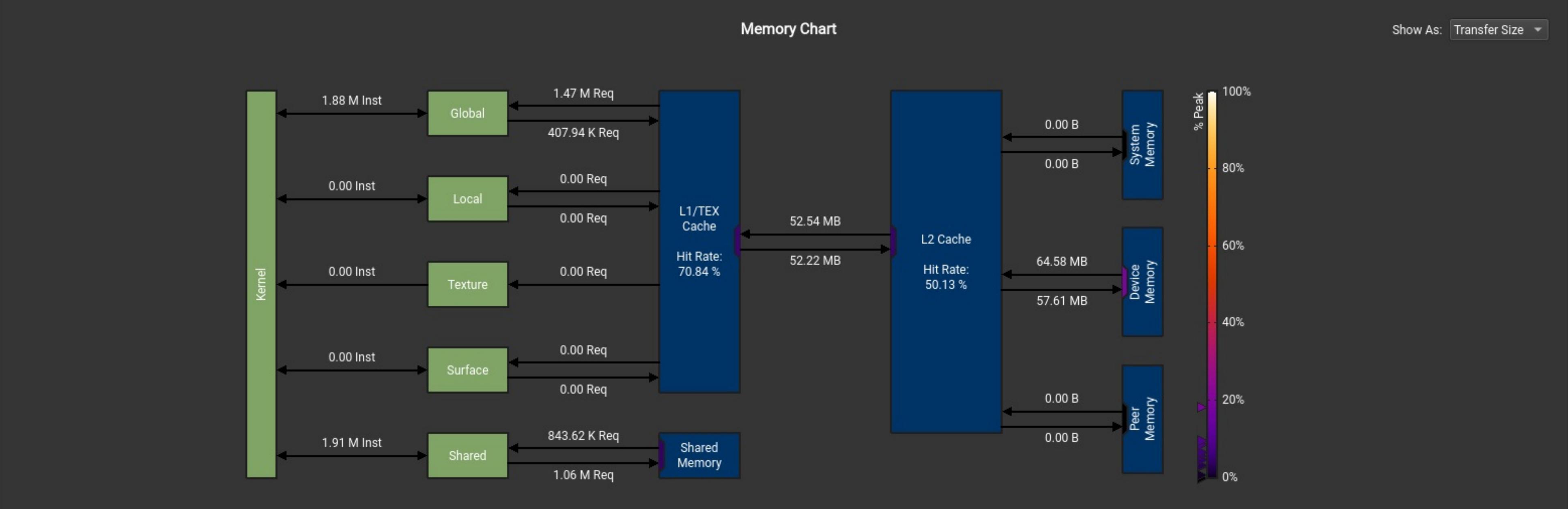
Memory Workload Analysis

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]	58,85	Mem Busy [%]	9,54
L1/TEX Hit Rate [%]	70,84	Max Bandwidth [%]	18,41
L2 Hit Rate [%]	50,13	Mem Pipes Busy [%]	17,25

Show As: Transfer Size



Shared Memory

	Instructions	Requests	Wavefronts	% Peak	Bank Conflicts
Shared Load	843.624	843.624	843.624	1,74	0
Shared Load Matrix	0	0			
Shared Store	1.061.468	1.061.468	1.061.468	2,19	0
Shared Atomic	0	0	0	0	0
Other	-	-	420.684	0,87	0
Total	1.905.092	1.905.092	2.325.776	4,79	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	1.468.017	1.468.017	1.468.017	3,02	3.998.881	2,72	58,94	127.964.192	1.641.808	3,38	1.468.017
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	407.936	407.936	407.936	0,84	1.631.744	4	100	52.215.808	1.631.744	3,36	-
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	1.468.017	1.468.017	1.468.017	3,02	3.998.881	2,72	58,94	127.964.192	1.641.808	3,38	1.468.017
Stores	407.936	407.936	407.936	0,84	1.631.744	4	100	52.215.808	1.631.744	3,36	-
Atomics & Reductions	0	0	0	0	0	0	0	0	0	0	-
Total	1.875.953	1.875.953	1.875.953	3,86	5.630.625	3,00	70,84	180.180.000	3.273.552	6,74	1.468.017

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	410.512	1.641.808	4,00	2,89	0,61	52.537.856	25.303.352.084,46	1.631.748	0	0
L1/TEX Store	407.936	1.631.744	4	2,87	100	52.215.808	25.148.246.898,36	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	818.449	3.273.552	4,00	5,76	50,15	104.753.664	50.451.598.982,82	1.631.748	0	0
ECC Total	-	8	-	0,00	-	256	123.295,06	8	-	-
GPU Total	819.225	3.277.142	4,00	5,77	50,17	104.868.544	50.506.927.641,21	1.631.759	0	0

Device Memory

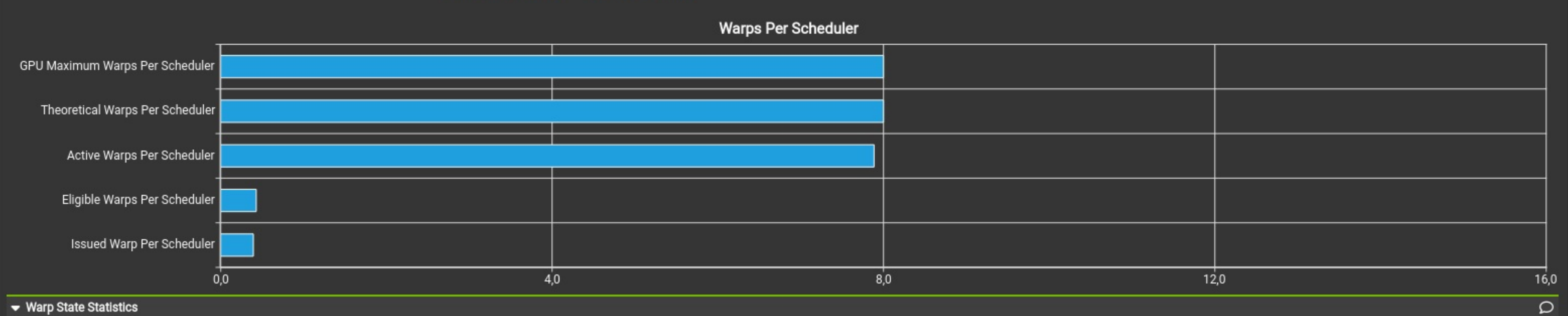
	Sectors	% Peak	Bytes	Throughput
Load	2.017.969	9,73	64.575.008	31.100.701.240,66
Store	1.800.258	8,68	57.608.256	27.745.364.876,32
Total	3.818.227	18,41	122.183.264	58.846.066.116,98

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]	7,89	No Eligible [%]	60,65
Eligible Warps Per Scheduler [warp]	0,43	One or More Eligible [%]	39,35
Issued Warp Per Scheduler	0,39		

Issue Slot Utilization
Est. Local Speedup: 60.65%
Every scheduler is capable of issuing one instruction per cycle, but for this kernel 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 8 warps per scheduler, this kernel allocates an average of 7.89 active warps per scheduler, but only an average of 0.43 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.



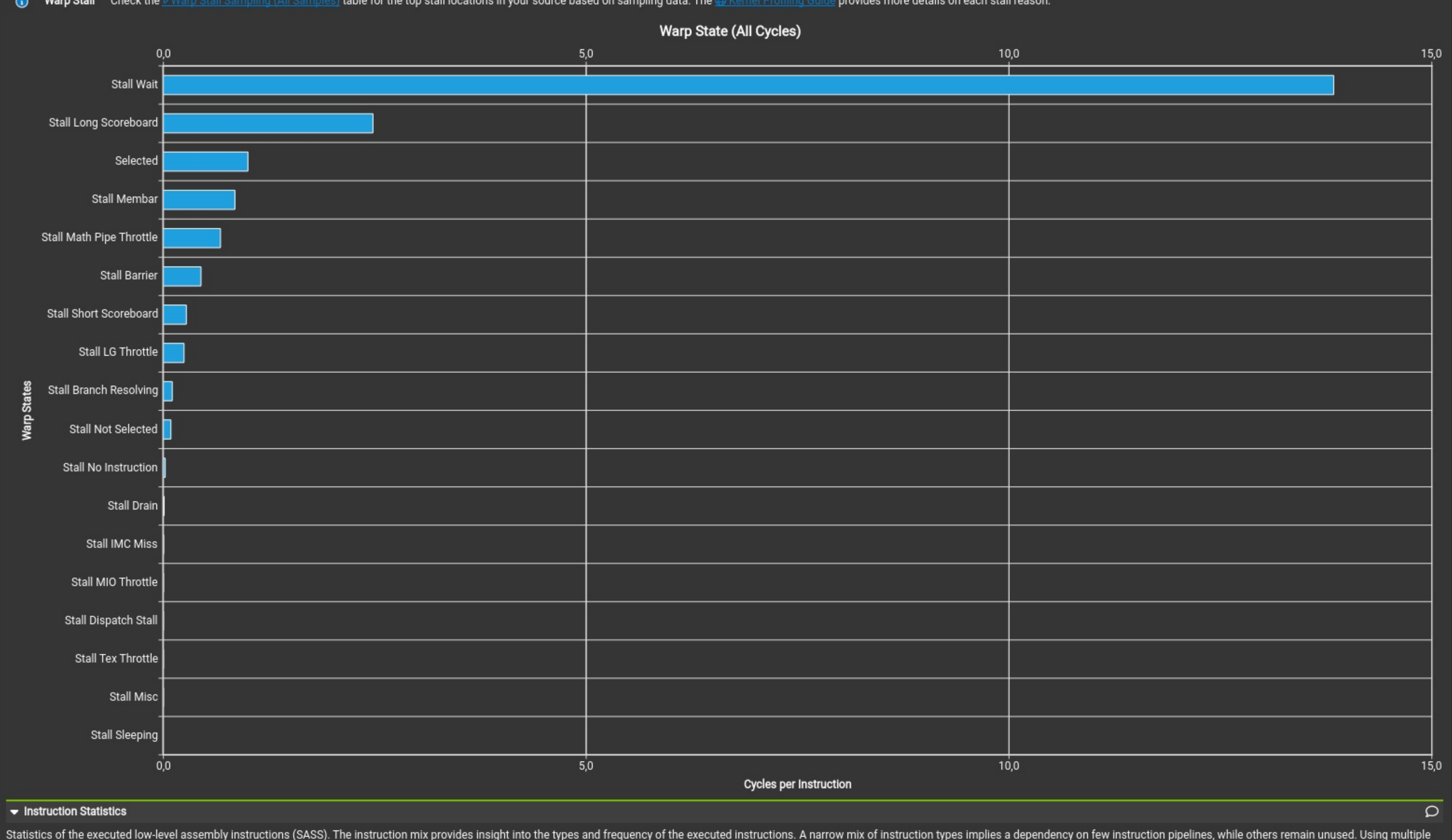
Warp State Statistics

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]	20,04	Avg. Active Threads Per Warp	32,00
Warp Cycles Per Executed Instruction [cycle]	20,04	Avg. Not Predicated Off Threads Per Warp	31,99

Wait
Est. Local Speedup: 60.65%
On average, each warp of this kernel spends 13.8 cycles being stalled waiting on a fixed latency execution dependency. Typically, this stall reason should be very low and only shows up as a top contributor in already highly optimized kernels. Try to hide the corresponding instruction latencies by increasing the number of active warps, restructuring the code or unrolling loops. Furthermore, consider switching to lower-latency instructions, e.g. by making use of fast math compiler options. This stall type represents about 69.1% of the total average of 20.0 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.

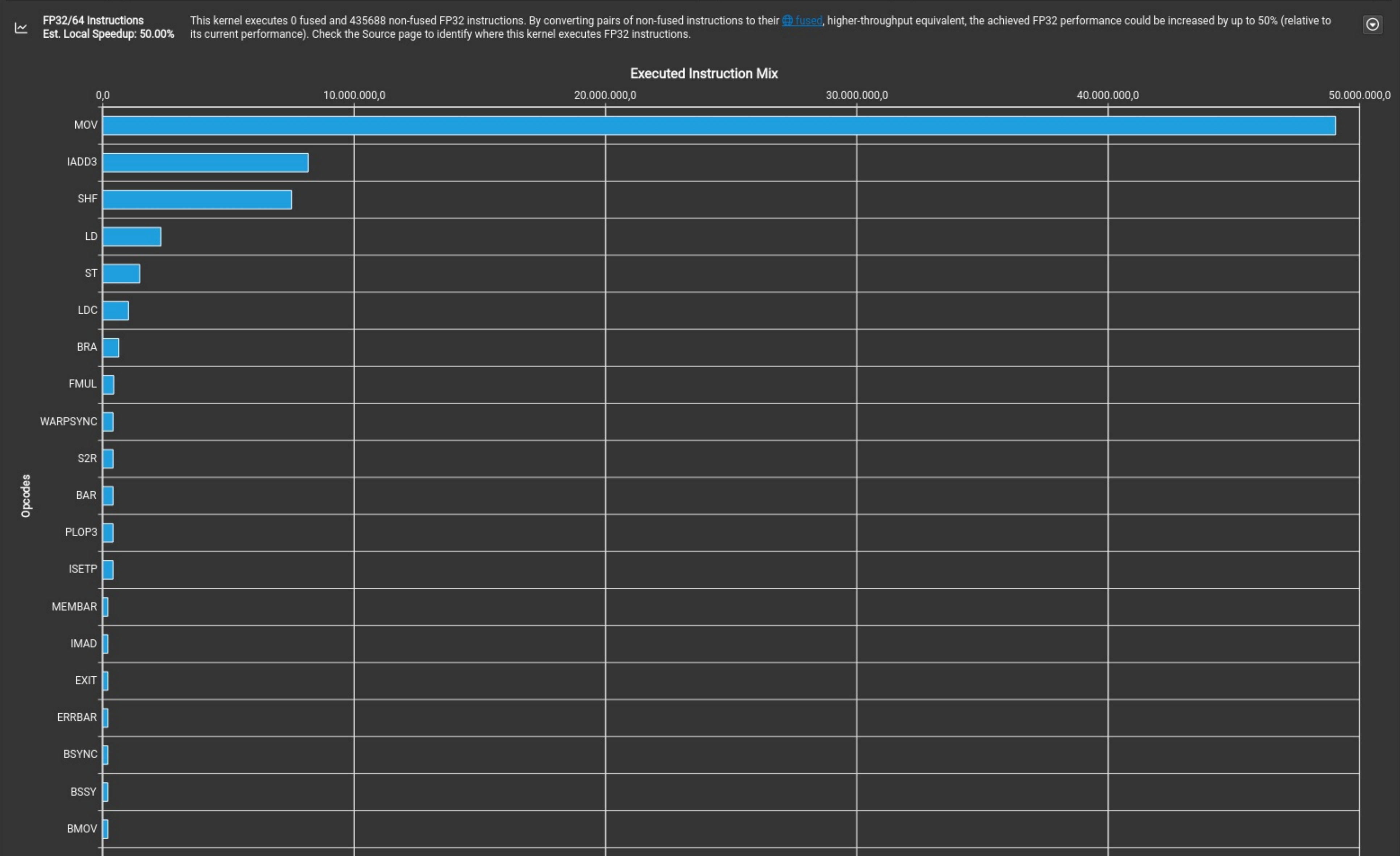


Instruction Statistics

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]	74.066.832	Avg. Executed Instructions Per Scheduler [inst]	462.917,70
Issued Instructions [inst]	74.068.113	Avg. Issued Instructions Per Scheduler [inst]	462.925,71

FP32/64 Instructions
Est. Local Speedup: 50.00%
This kernel executes 0 fused and 435688 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 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	6.374	Function Cache Configuration	CachePreferNone
Registers Per Thread [register/thread]	25	Static Shared Memory Per Block [Kbyte/block]	8,20
Block Size	1.024	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	6.526.976	Driver Shared Memory Per Block [byte/block]	0
Waves Per SM	159,35	Shared Memory Configuration Size [Kbyte]	32,77

Occupancy

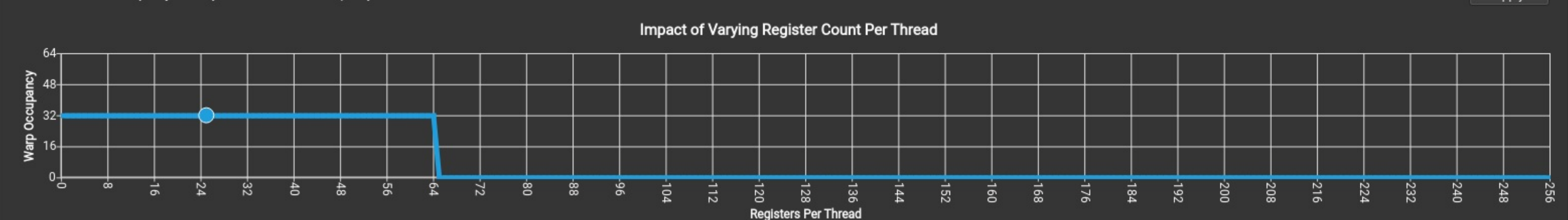
Occupancy is the ratio of the number of active warps per multiprocessor to the maximum number of possible active warps. Another way to view occupancy is the percentage of the hardware's ability to process warps that is actively in use. Higher occupancy does not always result in higher performance, however, 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.

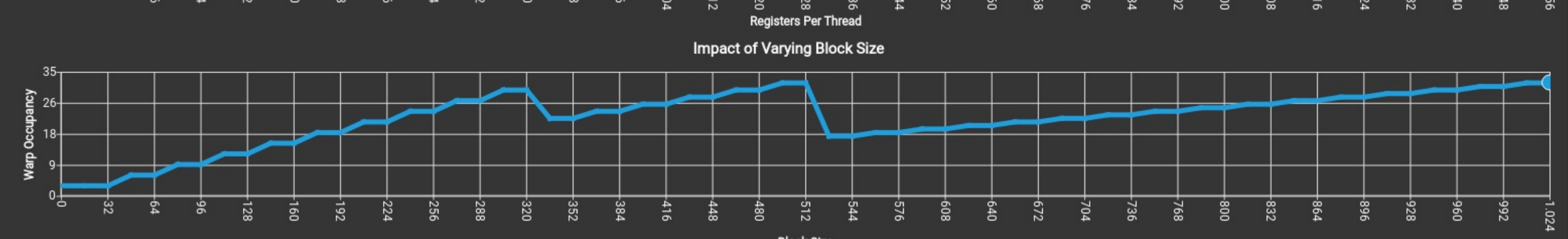
Theoretical Occupancy [%]	100	Block Limit Registers [block]	2
Theoretical Active Warps per SM [warp]	32	Block Limit Shared Mem [block]	3
Achieved Occupancy [%]	98,50	Block Limit Warps [block]	1
Achieved Active Warps Per SM [warp]	31,52	Block Limit SM [block]	16

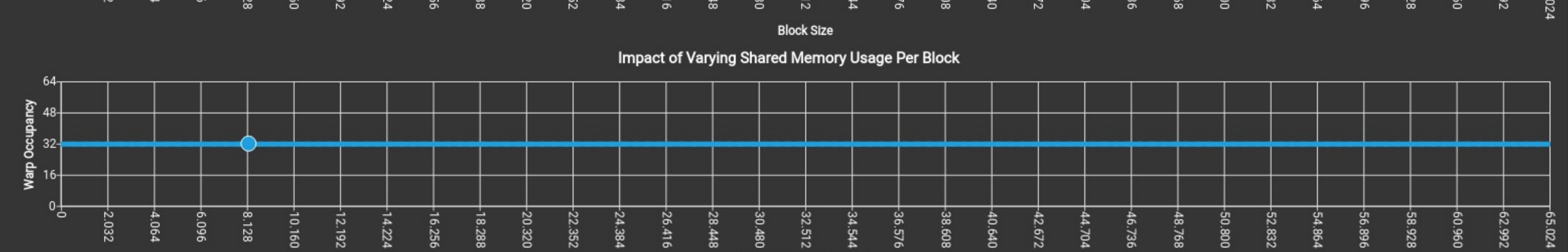
Occupancy Limiters This kernel's theoretical occupancy is not impacted by any block limit.

Achieved Occupancy Analysis of the Achieved Occupancy — Apply

Theoretical Occupancy Analysis of Theoretical Occupancy and its Limiters — Apply







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]	1.451.369	Branch Efficiency [%]	100,00
Branch Instructions Ratio [%]	0,02	Avg. Divergent Branches	0,01

Warp Stall Sampling (All Samples)

Location	Value	Value (%)
prova3.cu:41 (0x7d05cf033830 in applyMu...	2.364	5
prova3.cu:60 (0x7d05cf035220 in applyMu...	2.143	5
prova3.cu:42 (0x7d05cf033ab0 in applyMu...	2.059	4
prova3.cu:57 (0x7d05cf034cb0 in applyMu...	721	2
prova3.cu:40 (0x7d05cf0335d0 in applyMu...	456	1

Most Instructions Executed

Location	Value	Value (%)
prova3.cu:53 (0x7d05cf034c90 in applyMu...	203.969	0
prova3.cu:60 (0x7d05cf035230 in applyMu...	203.968	0
prova3.cu:60 (0x7d05cf035220 in applyMu...	203.968	0
prova3.cu:60 (0x7d05cf035210 in applyMu...	203.968	0
prova3.cu:58 (0x7d05cf035200 in applyMu...	203.968	0

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.