Page: Details | Result: 1 - 540 - gemm_shared_kernel | Add Baseline | Apply Rules | Occupancy Calculator | Save as PDF

	Result	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	540 - gemm_shared_kernel (8, 16, 5)x(128, ...	860.61 usecond	482,972	34	0 - NVIDIA RTX A2000	561.20 cycle/usecond	8.6	[2156] benchmark.exe

GPU Speed Of Light Throughput

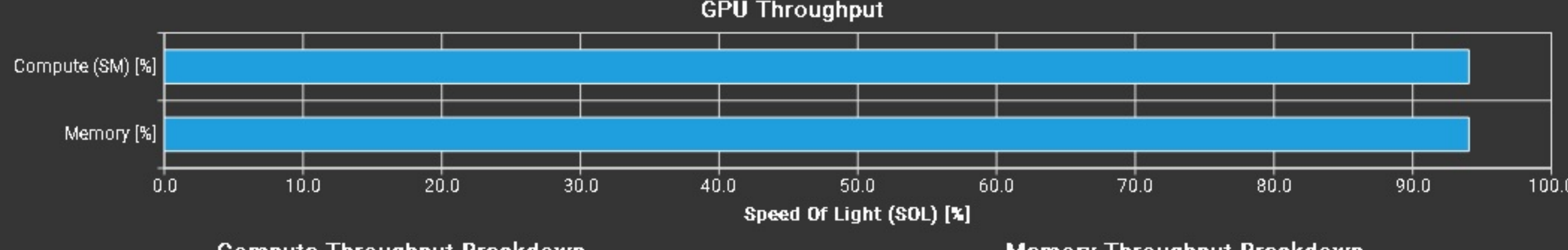
All

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.

Compute (SM) Throughput [%]	94.06	Duration [usecond]	860.61
Memory Throughput [%]	94.06	Elapsed Cycles [cycle]	482,972
L1/TEX Cache Throughput [%]	96.06	SM Active Cycles [cycle]	472,943.85
L2 Cache Throughput [%]	2.16	SM Frequency [cycle/usecond]	561.20
DRAM Throughput [%]	0.06	DRAM Frequency [cycle/nsecond]	5.68

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 workloads in the Compute Workload Analysis section.

GPU Throughput



Compute Throughput Breakdown

SM: Inst Executed Pipe Lsu [%]	94.06
SM: Mio Inst Issued [%]	33.12
SM: Issue Active [%]	23.67
SM: Inst Executed [%]	23.64
SM: Mio2rf Writeback Active [%]	22.34
SM: Pipe Alu Cycles Active [%]	7.98
SM: Pipe Fma Cycles Active [%]	7.21
SM: Inst Executed Pipe Adu [%]	5.30
SM: Pipe Fmaheavy Cycles Active [%]	4.07
SM: Mio Pq Read Cycles Active [%]	2.63
SM: Mio Pq Write Cycles Active [%]	2.63
SM: Inst Executed Pipe Uniform [%]	0.67
SM: Inst Executed Pipe Cbu Pred On Any [%]	0.02
IDC: Request Cycles Active [%]	0
SM: Inst Executed Pipe Ipa [%]	0
SM: Inst Executed Pipe Tex [%]	0
SM: Inst Executed Pipe Xu [%]	0
SM: Pipe Fp64 Cycles Active [%]	0
SM: Pipe Tensor Cycles Active [%]	0

Memory Throughput Breakdown

L1: Lsuin Requests [%]	94.06
L1: Data Pipe Lsu Wavefronts [%]	49.28
L1: Lsu Writeback Active [%]	44.63
L1: Data Bank Reads [%]	12.04
L2: T Sectors [%]	2.16
L2: Lts2xbar Cycles Active [%]	2.10
L1: M Xbar2l1tex Read Sectors [%]	2.06
L1: Data Bank Writes [%]	1.57
L2: Xbar2lts Cycles Active [%]	0.82
L2: T Tag Requests [%]	0.82
L1: M L1tex2xbar Req Cycles Active [%]	0.73
L1: Texin Sm2tex Req Cycles Active [%]	0.65
L2: D Sectors [%]	0.57
DRAM: Dram Sectors [%]	0.06
DRAM: Cycles Active [%]	0.05
L2: D Sectors Fill Device [%]	0.03
L1: Data Pipe Tex Wavefronts [%]	0
L1: F Wavefronts [%]	0
L1: Tex Writeback Active [%]	0
L2: D Atomic Input Cycles Active [%]	0
L2: D Sectors Fill Sysmem [%]	0

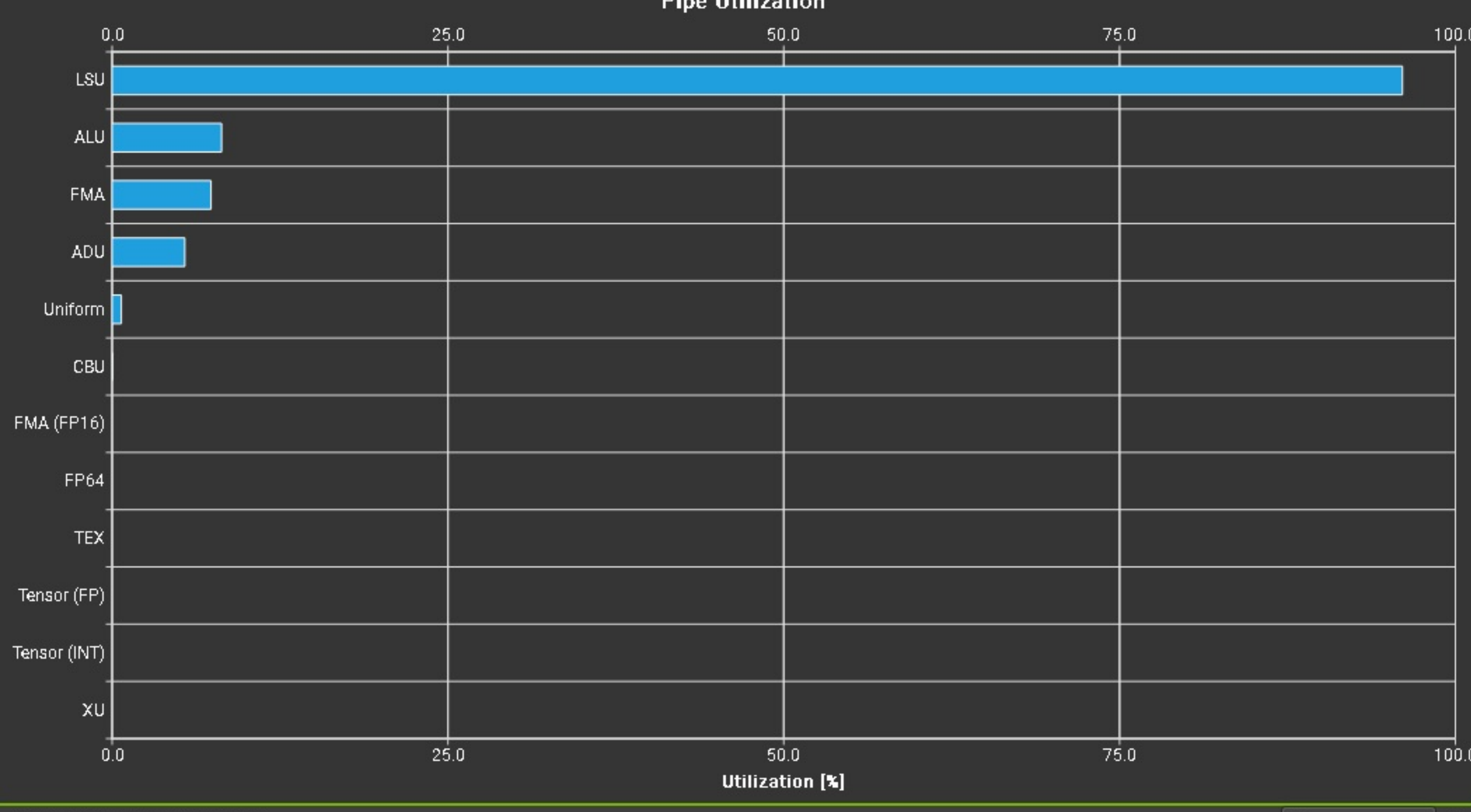
Compute Workload Analysis

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]	0.95	SM Busy [%]	33.82
Executed Ipc Active [inst/cycle]	0.97	Issue Slots Busy [%]	24.17
Issued Ipc Active [inst/cycle]	0.97		

Very High Utilization LSU is the highest-utilized pipeline (96.1%). It executes load/store memory operations. The pipeline is over-utilized and likely a performance bottleneck. 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.

Pipe Utilization



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.

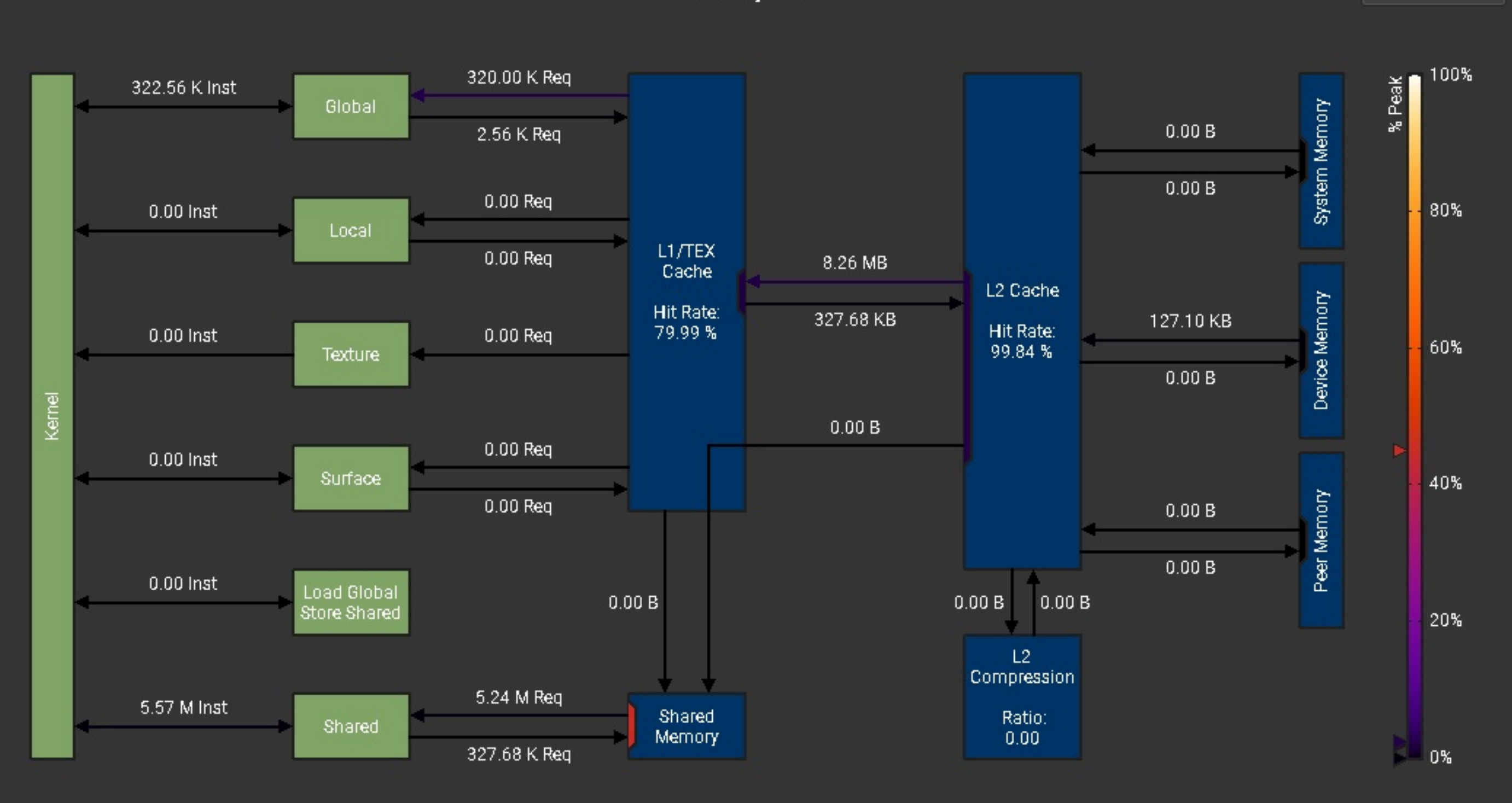
Memory Throughput [Mbyte/second]	147.69	Mem Busy [%]	49.28
L1/TEX Hit Rate [%]	79.99	Max Bandwidth [%]	94.06
L2 Hit Rate [%]	99.84	Mem Pipes Busy [%]	94.06
L2 Compression Success Rate [%]	0	L2 Compression Ratio	0

L2 Store Access Pattern The memory access pattern for stores from L1TEX to L2 is not optimal. The granularity of an L1TEX request to L2 is a 128 byte cache line. That is 4 consecutive 32-byte sectors per L2 request. However, this kernel only accesses an average of 3.3 sectors out of the possible 4 sectors per cache line. Check the Source Counters section for uncoalesced stores and try to minimize how many cache lines need to be accessed per memory request.

L2 Load Access Pattern The memory access pattern for loads from L1TEX to L2 is not optimal. The granularity of an L1TEX request to L2 is a 128 byte cache line. That is 4 consecutive 32-byte sectors per L2 request. However, this kernel only accesses an average of 2.7 sectors out of the possible 4 sectors per cache line. Check the Source Counters section for uncoalesced loads and try to minimize how many cache lines need to be accessed per memory request.

Memory Chart

Show As: Transfer Size



Shared Memory

	Instructions	Requests	Wavefronts	% Peak	Bank Conflicts
Shared Load	5,242,880	5,242,880	5,243,048	41.75	0
Shared Load Matrix	0	0	0	0	0
Shared Store	327,680	327,680	327,680	0.65	0
Shared Store From Global Load	0	0	0	0	0
Shared Atomic	0	0	0	0	0
Other	-	-	94,717	2.71	0
Total	5,570,560	5,570,560	5,665,445	45.12	0

L1/TEX Cache

	Instructions	Requests	Wavefronts	% Peak	Sectors	Sectors/Req	Hit
Local Load	0	0	0	0	0	0	
Global Load	320,000	320,000			1,266,078	3.96	
Global Load To Shared Store (access)	0	0	325,120	2.59	0	0	
Global Load To Shared Store (bypass)	0	0			0	0	
Surface Load	0	0	0	0	0	0	
Texture Load	0	0	0	0	0	0	
Global Store	2,560	2,560	2,560	0.02	10,240	4	
Local Store	0	0	0	0	0	0	
Surface Store	0	0	0	0	0	0	
Global Reduction	0	0	0	0	0	0	
Surface Reduction	0	0	0	0	0	0	
Global Atomic ALU	0	0	0	0	0	0	
Global Atomic CAS	0						
Surface Atomic ALU	0	0	0	0	0	0	
Surface Atomic CAS	0						
Loads	320,000	320,000	325,120	2.59	1,266,078	3.96	
Stores	2,560	2,560	2,560	0.02	10,240	4	
Atomics & Reductions	0	0	0	0	0	0	

L2 Cache

	Requests	Sectors	Sectors/Req	% Peak	Hit Rate	Bytes	Throughput
L1/TEX Load	96,168	260,484	2.71	2.11	98.47	8,335,488	9,685,580,426
L1/TEX Store	3,092	10,240	3.31	0.08	100	327,680	380,754,071
L1/TEX Atomic ALU	0	0	0	0	0	0	
L1/TEX Atomic CAS	0	0	0	0	0	0	
L1/TEX Reduction	0	0	0	0	0	0	
L1/TEX Total	99,532	266,668	2.68	2.16	98.54	8,533,376	9,915,520,190
ECC Total	-	0	-	0	-	0	

L2 Cache Eviction Policies

	First	Hit Rate	Last	Hit Rate	Normal	Hit Rate	Normal Demote
L1/TEX Load	0	0	0	0	259,642	98.47	
L1/TEX Store	0	0	0	0	10,240	100	
L1/TEX Atomic	0	0	0	0	0	0	
L1/TEX Total	0	0	0	0	270,554	98.53	

Device Memory

	Sectors	% Peak	Bytes	Throughput
Load	3,972	0.05	127,104	147,690,934.78
Store	0	0	0	0
Total	3,972	0.05	127,104	147,690,934.78

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]	9.60	No Eligible [%]	75.83
Eligible Warps Per Scheduler [warp]	1.05	One or More Eligible [%]	24.17
Issued Warp Per Scheduler	0.24		

Issue Slot Utilization Every scheduler is capable of issuing one instruction per cycle, but for this kernel each scheduler only issues an instruction every 4.1 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 9.60 active warps per scheduler, but only an average of 1.05 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, reduce the time the active warps are stalled by inspecting the top stall reasons on the Warp State Statistics and Source Counters sections.

Warp State Statistics

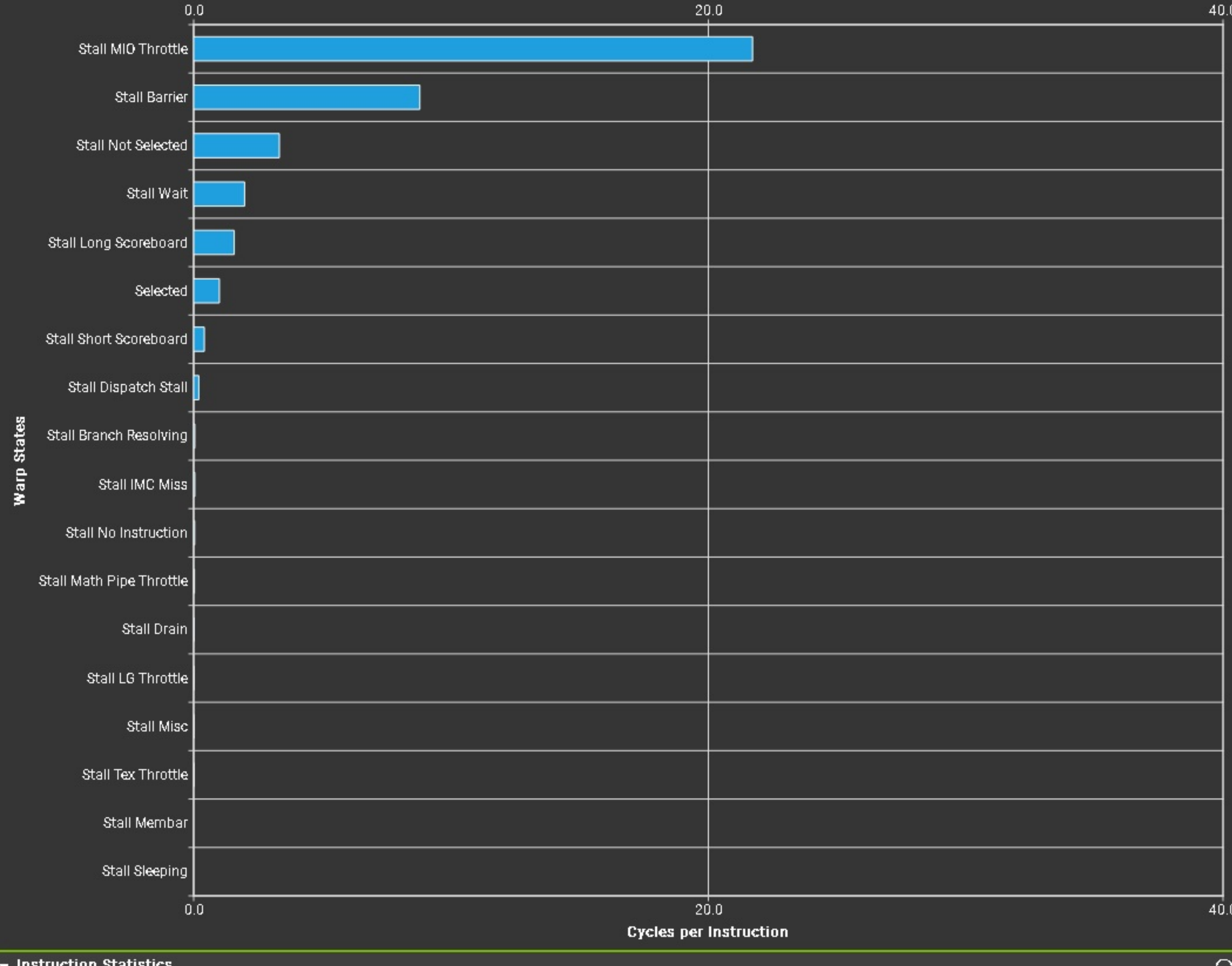
Analysis of the states in which all warps spent cycles during the kernel execution. The warp states describe a warp's readiness or inability to issue its next instruction. The warp cycles per instruction define the latency between two consecutive instructions. The higher the value, the more warp parallelism 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]	39.72	Avg. Active Threads Per Warp	32
Warp Cycles Per Executed Instruction [cycle]	39.76	Avg. Not Predicated Off Threads Per Warp	31.91

mio_throttle On average, each warp of this kernel spends 21.7 cycles being stalled waiting for an MIO instruction queue to be not full. This represents about 54.7% of the total average of 39.7 cycles between issuing two instructions. This stall reason is high in cases of utilization of the MIO pipelines, which include special math instructions, dynamic branches, as well as shared memory instructions. When caused by shared memory accesses, trying to use fewer but wider loads can reduce pipeline pressure.

Warp Stall Check the Source Counters section 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)

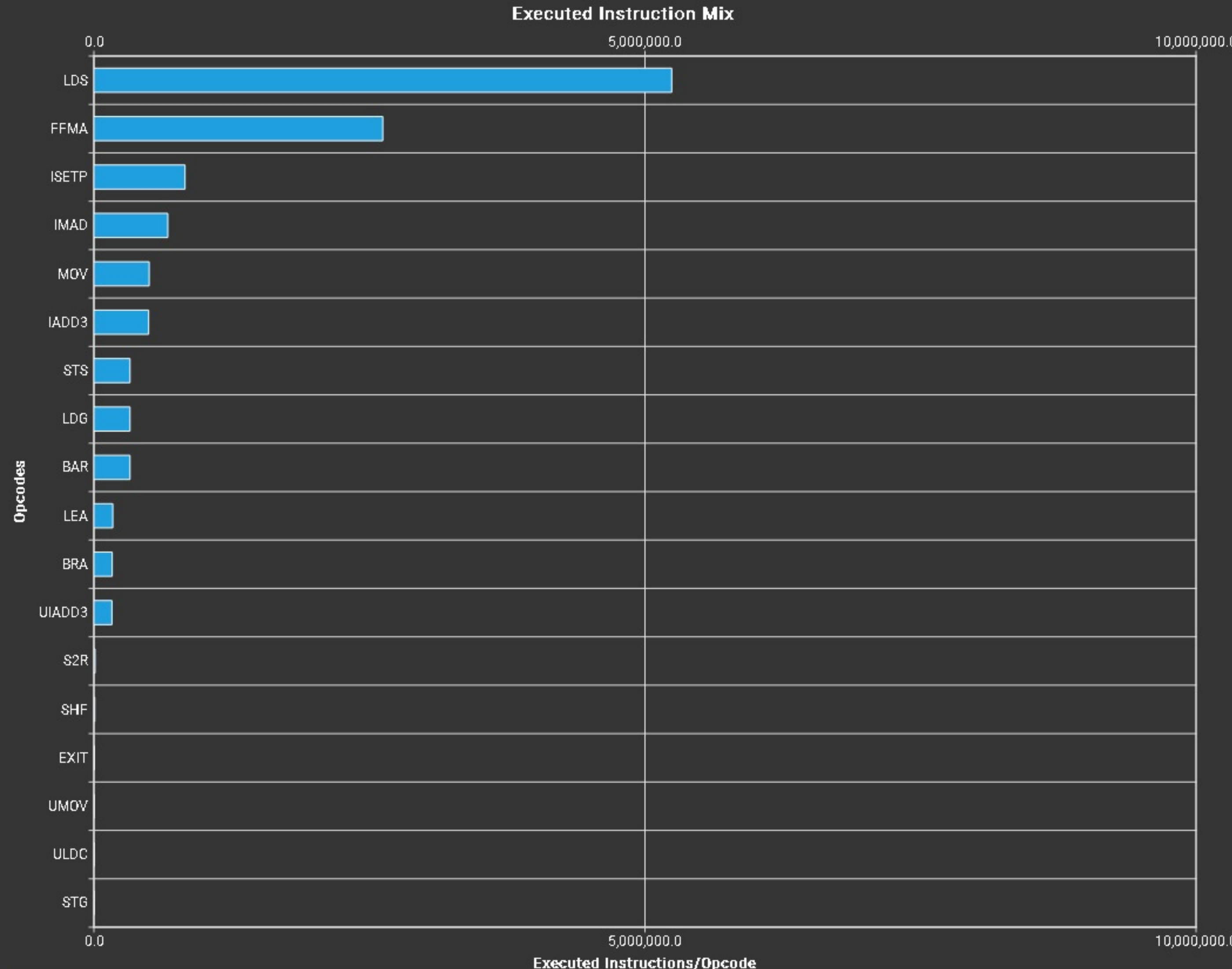


Instruction Statistics

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]	11,875,840	Avg. Executed Instructions Per Scheduler [inst]	114,190.77
Issued Instructions [inst]	11,888,736	Avg. Issued Instructions Per Scheduler [inst]	114,314.77

Executed Instruction Mix



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	640	Registers Per Thread [register/thread]	34
Block Size	128	Static Shared Memory Per Block [Kbyte/block]	2.05
Threads [thread]	81,920	Dynamic Shared Memory Per Block [byte/block]	0
Waves Per SM	2.05	Driver Shared Memory Per Block [Kbyte/block]	1.02
Function Cache Configuration	cudaFuncCachePreferNone	Shared Memory Configuration Size [Kbyte]	65.54

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]	12
Theoretical Active Warps per SM [warp]	48	Block Limit Shared Mem [block]	21
Achieved Occupancy [%]	80.01	Block Limit Warps [block]	12
Achieved Active Warps Per SM [warp]	38.41	Block Limit SM [block]	16

Occupancy Limiters This kernel's theoretical occupancy is not impacted by any block limit. The difference between calculated theoretical (100.0%) and measured achieved occupancy (80.0%) 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.