Page: Details | Result: 20 - 19113 - ampere_sgemm_128x6… | Add Baseline | Apply Rules | Occupancy Calculator | Save as PDF

	Result	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	19113 - ampere_sgemm_128x64_nn (8, 16, 5)x...	995.71 usecond	558,943	122	0 - NVIDIA RTX A2000	561.35 cycle/usecond	8.6	[26036] 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 [%]	72.70	Duration [usecond]	995.71
Memory Throughput [%]	61.34	Elapsed Cycles [cycle]	558,943
L1/TEX Cache Throughput [%]	62.50	SM Active Cycles [cycle]	548,559.38
L2 Cache Throughput [%]	31.59	SM Frequency [cycle/usecond]	561.35
DRAM Throughput [%]	20.37	DRAM Frequency [cycle/nsecond]	5.68

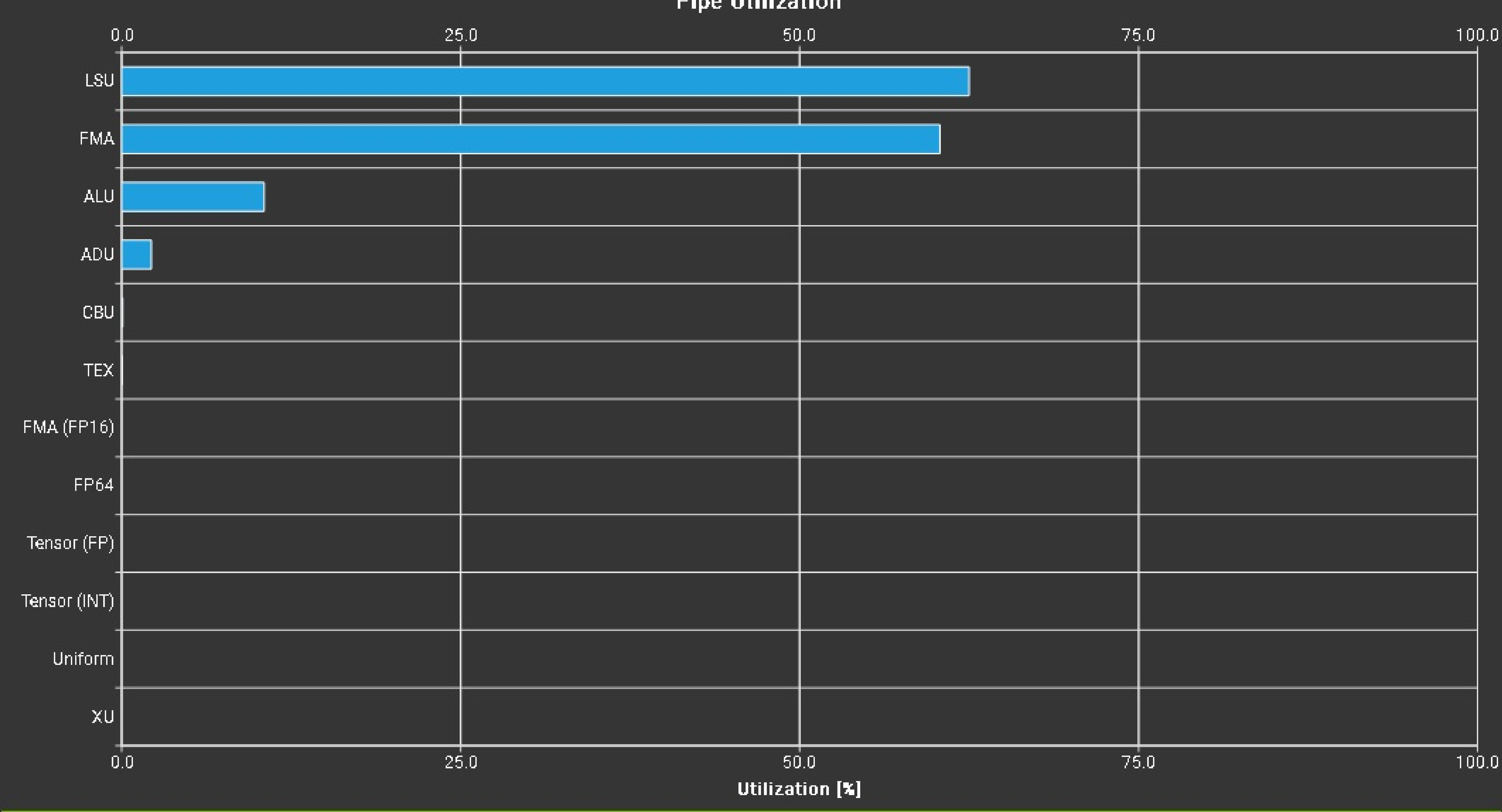
High Compute Throughput Compute is more heavily utilized than Memory: Look at the Compute Workload Analysis section to see what the compute pipelines are spending their time doing. Also, consider whether any computation is redundant and could be reduced or moved to look-up tables.

Compute 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]	2.91	SM Busy [%]	74.08
Executed Ipc Active [inst/cycle]	2.96	Issue Slots Busy [%]	74.08
Issued Ipc Active [inst/cycle]	2.96		

High Utilization LSU is the highest-utilized pipeline (62.5%). It executes load/store memory operations. The pipeline is well-utilized and might become a bottleneck if more work is added. 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

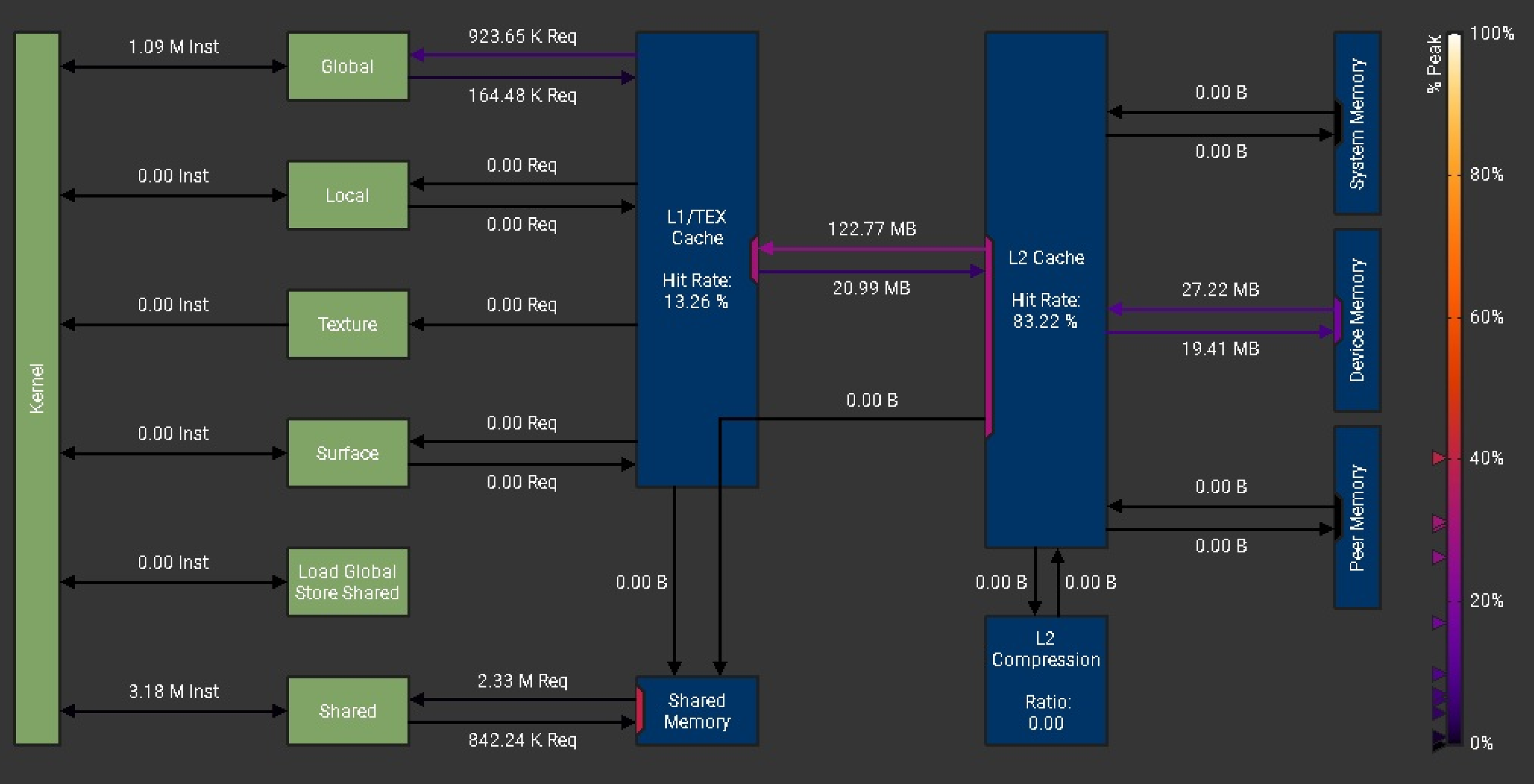
Detailed analysis of the memory resources of the GPU. Memory can become a limiting factor for the overall kernel performance when fully utilizing the involved hardware units (Mem Busy), exhausting the available communication bandwidth between those units (Max Bandwidth), or by reaching the maximum throughput of issuing memory instructions (Mem Pipes Busy). Detailed chart of the memory units. Detailed tables with data for each memory unit.

Memory Throughput [Gbyte/second]	46.83	Mem Busy [%]	55.81
L1/TEX Hit Rate [%]	13.26	Max Bandwidth [%]	61.34
L2 Hit Rate [%]	83.22	Mem Pipes Busy [%]	61.34
L2 Compression Success Rate [%]	0	L2 Compression Ratio	0

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 2.2 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	2,334,720	2,334,720	4,507,467	31.02	0
Shared Load Matrix	0	0	0	0	0
Shared Store	842,240	842,240	1,128,960	1.94	163,840
Shared Store From Global Load	0	0	0	0	0
Shared Atomic	0	0	0	0	0
Other	-	-	228,223	7.40	0
Total	3,176,960	3,176,960	5,864,650	40.36	163,840

L1/TEX Cache

	Instructions	Requests	Wavefronts	% Peak	Sectors	Sectors/Req	Hit
Global Load	923,648	923,648			4,469,806	4.84	
Global Load To Shared Store (access)	0	0	974,901	6.71	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	164,480	164,480	164,480	1.13	656,000	3.99	
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	923,648	923,648	974,901	6.71	4,469,806	4.84	
Stores	164,480	164,480	164,480	1.13	656,000	3.99	
Atomics & Reductions	0	0	0	0	0	0	
Total	1,088,128	1,088,128	1,139,381	7.84	5,125,806	4.71	

L2 Cache

	Requests	Sectors	Sectors/Req	% Peak	Hit Rate	Bytes	Throughput
L1/TEX Store	164,480	656,000	3.99	4.58	100	20,992,000	21,082,401,336
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	1,901,304	4,505,007	2.37	31.48	81.51	144,160,224	144,781,045,121
ECC Total	-	0	-	0	-	0	
GPU Total	1,915,997	4,520,004	2.36	31.59	81.53	144,640,128	145,263,015,811

L2 Cache Eviction Policies

	First	Hit Rate	Last	Hit Rate	Normal	Hit Rate	Normal Demote
L1/TEX Load	0	0	0	0	3,944,393	78.44	
L1/TEX Store	656,000	100	0	0	0	0	
L1/TEX Atomic	0	0	0	0	0	0	
L1/TEX Total	656,000	100	0	0	3,946,748	78.46	

Device Memory

	Sectors	% Peak	Bytes	Throughput
Load	850,516	10.02	27,216,512	27,333,718,987.02
Store	606,512	7.15	19,408,384	19,491,965,548.27
Total	1,457,028	17.16	46,624,896	46,825,684,535.29

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]	3.84	No Eligible [%]	25.88
Eligible Warps Per Scheduler [warp]	1.94	One or More Eligible [%]	74.12
Issued Warp Per Scheduler	0.74		

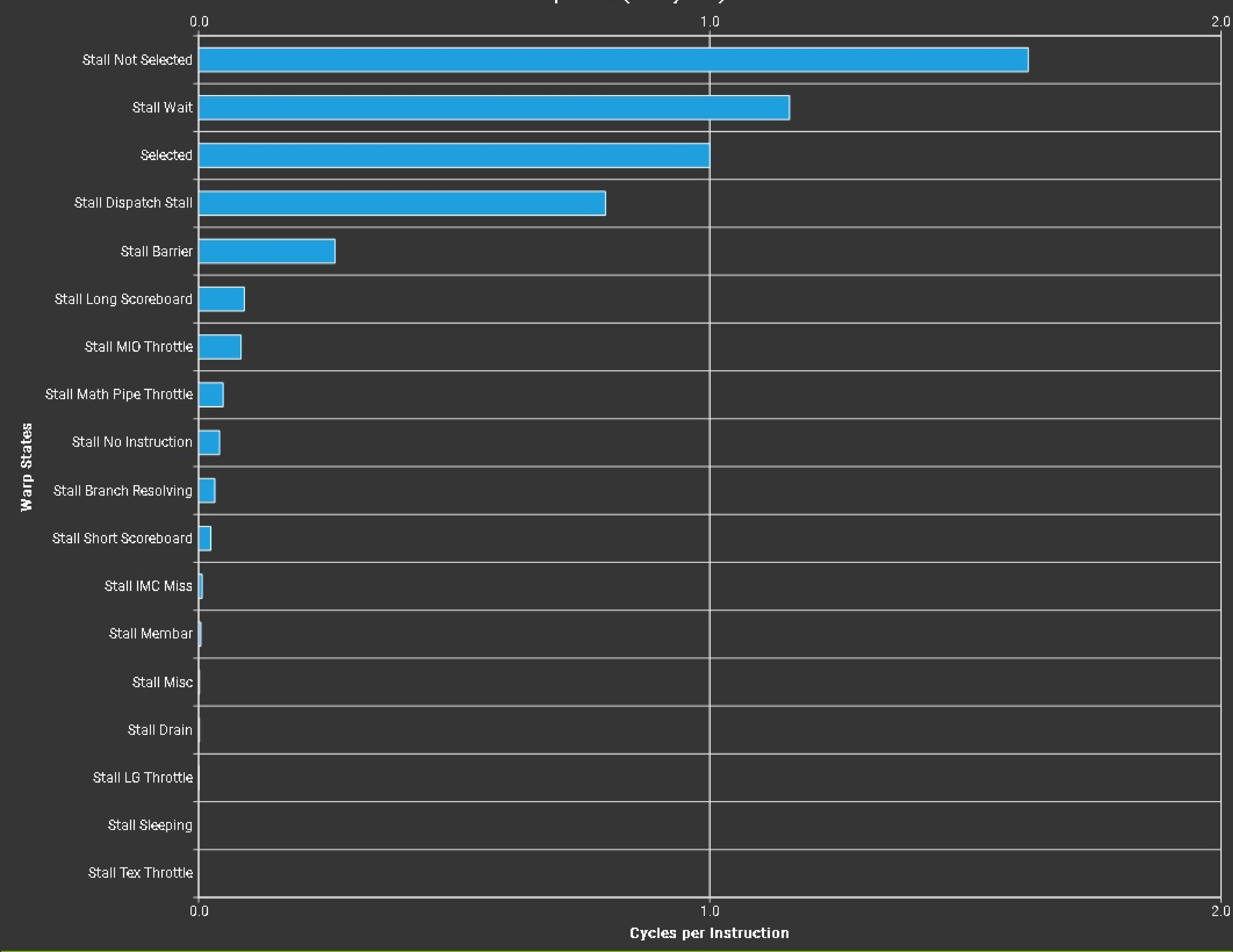
Warp State Statistics

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]	5.18	Avg. Active Threads Per Warp	32
Warp Cycles Per Executed Instruction [cycle]	5.18	Avg. Not Predicated Off Threads Per Warp	31.58

not_selected On average, each warp of this kernel spends 1.6 cycles being stalled due to not being selected by the scheduler. This represents about 31.3% of the total average of 5.2 cycles between issuing two instructions. Not selected warps are eligible warps that were not picked by the scheduler to issue that cycle as another warp was selected. A high number of not selected warps typically means you have sufficient warps to cover warp latencies and you may consider reducing the number of active warps to possibly increase cache coherence and data locality.

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.



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]	42,254,848	Avg. Executed Instructions Per Scheduler [inst]	406,296.62
Issued Instructions [inst]	42,259,996	Avg. Issued Instructions Per Scheduler [inst]	406,346.12

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]	122
Block Size	128	Static Shared Memory Per Block [Kbyte/block]	12.54
Threads [thread]	81,920	Dynamic Shared Memory Per Block [byte/block]	0
Waves Per SM	6.15	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 [%]	33.33	Block Limit Registers [block]	4
Theoretical Active Warps per SM [warp]	16	Block Limit Shared Mem [block]	4
Achieved Occupancy [%]	31.96	Block Limit Warps [block]	12
Achieved Active Warps Per SM [warp]	15.34	Block Limit SM [block]	16

Occupancy Limiters This kernel's theoretical occupancy (33.3%) is limited by the number of required registers This kernel's theoretical occupancy (33.3%) is limited by the required amount of shared memory See the CUDA Best Practices Guide for more details on optimizing occupancy.