	Result	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	19072 - gemm_shared_kernel (64, 64, 1)x(1...	10.53 msecond	5,923,428	34	0 - NVIDIA RTX A2000	562.52 cycle/usecond	8.6	[26376] benchmark.exe

GPU Speed Of Light Throughput

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 [%]	98.17	Duration [msecond]	10.53
Memory Throughput [%]	98.17	Elapsed Cycles [cycle]	5,923,428
L1/TEX Cache Throughput [%]	98.54	SM Active Cycles [cycle]	5,901,399.42
L2 Cache Throughput [%]	9.27	SM Frequency [cycle/usecond]	562.52
DRAM Throughput [%]	10.36	DRAM Frequency [cycle/usecond]	5.70

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.

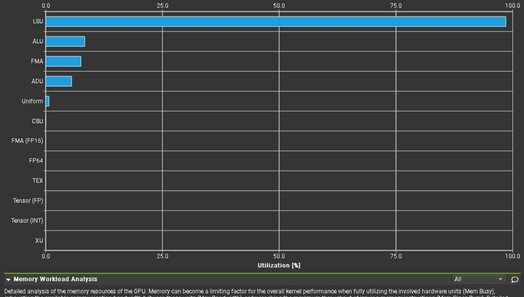
Compute Workload Analysis

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]	0.99	SM Busy [%]	34.70
Executed Ipc Active [inst/cycle]	0.99	Issue Slots Busy [%]	24.77
Issued Ipc Active [inst/cycle]	0.99		

Very High Utilization LSU is the highest-utilized pipeline (98.5%). 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 Pipeline Statistics section shows the mix of executed instructions in this kernel. Check the Pipeline Utilization section for which resource causes warps to stall.

Pipe Utilization



Memory Workload Analysis

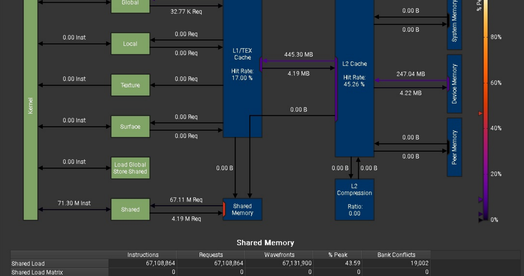
Detailed analysis of the memory resources of the GPU. Memory can become a limiting factor for the overall kernel performance when fully utilizing the involved hardware units (Mem Busy), exhausting the available communication bandwidth between those units (Max Bandwidth), or by reaching the maximum throughput 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]	23.86	Mem Busy [%]	53.00
L1/TEX Hit Rate [%]	17.00	Max Bandwidth [%]	98.17
L2 Hit Rate [%]	45.26	Mem Pipes Busy [%]	98.17
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 2.0 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 2.0 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



Shared Memory

	Instructions	Requests	Wavefronts	% Peak	Bank Conflicts
Shared Load	67,108,864	67,108,864	67,131,900	43.59	19,002
Shared Load Matrix	0	0	0	0	0
Shared Store	4,194,304	4,194,304	4,194,304	0.68	0
Shared Store From Global Load	0	0	0	0	0
Shared Atomic	0	0	0	0	0
Other			778,001	2.55	188,016
Total	71,303,168	71,303,168	72,104,205	46.82	207,018

L1/TEX Cache

	Instructions	Requests	Wavefronts	% Peak	Sectors	Sectors/Req	Hit
Global Load	4,194,304	4,194,304			16,705,574	3.98	
Global Load To Shared Store (access)	0	0	4,194,304	2.72	0	0	
Global Load To Shared Store (bypass)	0	0	0		0	0	
Surface Load	0	0	0	0	0	0	
Texture Load	0	0	0	0	0	0	
Global Store	32,768	32,768	32,768	0.02	131,072	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						
Global Atomic CAS	0						
Surface Atomic ALU	0	0	0	0	0	0	
Surface Atomic CAS	0						
Loads	4,194,304	4,194,304	4,194,304	2.72	16,705,574	3.98	
Stores	32,768	32,768	32,768	0.02	131,072	4	
Atomics & Reductions	0	0	0	0	0	0	
Total	4,227,072	4,227,072	4,227,072	2.74	16,836,646	3.98	

L2 Cache

	Requests	Sectors	Sectors/Req	% Peak	Hit Rate	Bytes	Throughput
L1/TEX Store	65,536	131,072	2	0.09	100	4,194,304	398,310,391
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	7,023,315	14,049,522	2.00	9.26	45.17	449,552,704	42,691,591,454
ECC Total				0		0	
DRAM Total	7,025,393	14,049,560	2.00	9.27	45.15	449,584,000	42,694,563,464

L2 Cache Eviction Policies

	First	Hit Rate	Last	Hit Rate	Normal	Hit Rate	Normal Demote
L1/TEX Load	0	0	0	0	13,953,796	44.50	
L1/TEX Store	0	0	0	0	131,072	100	
L1/TEX Atomic	0	0	0	0	0	0	
L1/TEX Total	0	0	0	0	14,046,722	45.13	

Device Memory

	Sectors	% Peak	Bytes	Throughput
Load	7,720,156	8.58	247,044,992	23,460,528,155.19
Store	131,776	0.15	4,216,832	400,449,752.33
Total	7,851,932	8.73	251,261,824	23,860,977,907.44

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]	11.81	No Eligible [%]	75.23
Eligible Warps Per Scheduler [warp]	1.35	One or More Eligible [%]	24.77
Issued Warp Per Scheduler	0.25		

Issue Slot Utilization Every scheduler is capable of issuing one instruction per cycle, but for this kernel each scheduler only issues an instruction every 4.0 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 11.81 active warps per scheduler, but only an average of 1.35 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 Scheduler Statistics sections can help, too.

Warp State Statistics

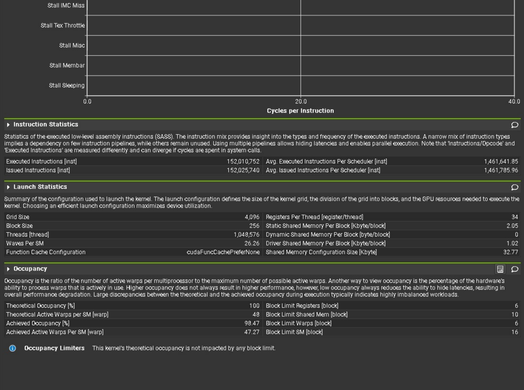
Analysis of the states in which all warps spent cycles during the kernel execution. The warp states describe a warp's readiness or inability to issue its next instruction. The warp cycles per instruction define the latency between two consecutive instructions. The higher the value, the more warp parallelism is required to hide this latency. For each warp state, the chart shows the average number of cycles spent in that state for 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]	47.70	Avg. Active Threads Per Warp	32
Warp Cycles Per Executed Instruction [cycle]	47.70	Avg. Not Predicated Off Threads Per Warp	31.98

mio_throttle On average, each warp of this kernel spends 27.9 cycles being stalled waiting for an MIO instruction queue to be not full. This represents about 58.4% of the total average of 47.7 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]	152,012,752	Avg. Executed Instructions Per Scheduler [inst]	1,461,641.85
Issued Instructions [inst]	152,025,740	Avg. Issued Instructions Per Scheduler [inst]	1,461,785.96

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	4,096	Registers Per Thread [register/thread]	34
Block Size	256	Static Shared Memory Per Block [byte/block]	2.05
Threads [thread]	1,048,576	Dynamic Shared Memory Per Block [byte/block]	0
Waves Per SM	26.26	Driver Shared Memory Per Block [byte/block]	1.02
Function Cache Configuration	cudaFuncCachePreferNone	Shared Memory Configuration Size [byte]	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]	6
Theoretical Active Warps per SM [warp]	48	Block Limit Shared Mem [block]	10
Achieved Occupancy [%]	98.47	Block Limit Warps [block]	6
Achieved Active Warps Per SM [warp]	47.27	Block Limit SM [block]	16

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