Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current 563 - reduce_argmin	(1024, 1, 1)x(1024, 1, 1)	123.62 us	72,301	0 - Tesla T4	584.83 Mhz	[10345] 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. High-level overview of the utilization for compute and memory resources of the GPU presented as a roofline chart.

Compute (SM) Throughput [%]	24.41	Duration [us]	123.62
Memory Throughput [%]	40.42	Elapsed Cycles [cycle]	72,301
L1/TEX Cache Throughput [%]	33.50	SM Active Cycles [cycle]	66,030.25
L2 Cache Throughput [%]	11.73	SM Frequency [Mhz]	584.83
DRAM Throughput [%]	40.42	DRAM Frequency [Ghz]	5.00

Latency Issue This kernel exhibits low compute throughput and memory bandwidth utilization relative to the peak performance of this device. Achieved compute throughput and/or memory bandwidth below 60.0% of peak typically indicate latency issues. Look at Scheduler Statistics and Warp State Statistics for potential reasons.

Roofline Analysis The ratio of peak float (fp32) to double (fp64) performance on this device is 32:1. The kernel achieved 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: Inst Executed Pipe Lsu [%]	24.41
SM: Issue Active [%]	10.59
SM: Inst Executed [%]	10.53
SM: Mio Inst Issued [%]	9.75
SM: Pipe Alu Cycles Active [%]	6.22
SM: Mio Pq Write Cycles Active [%]	5.42
SM: Mio Pq Read Cycles Active [%]	5.35
SM: Inst Executed Pipe Adu [%]	4.83
SM: Mio2rf Writeback Active [%]	4.55
SM: Pipe Fma Cycles Active [%]	3.87
SM: Inst Executed Pipe Cbu Pred On Any [%]	2.59
SM: Inst Executed Pipe Uniform [%]	1.14
SM: Memory Throughput Internal Activity [%]	0
SM: Pipe Tensor Cycles Active [%]	0
SM: Pipe Shared Cycles Active [%]	0
SM: Pipe Fp64 Cycles Active [%]	0
IDC: Request Cycles Active [%]	0
SM: Instruction Throughput Internal Activity [%]	0
SM: Inst Executed Pipe Xu [%]	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 [%]	40.42
DRAM: Dram Sectors [%]	29.58
L1: Lsuin Requests [%]	24.41
L1: Data Pipe Lsu Wavefronts [%]	16.75
L1: M Xbar2l1tex Read Sectors [%]	13.67
L2: T Sectors [%]	11.73
L2: Lts2xbar Cycles Active [%]	11.70
L2: D Sectors Fill Device [%]	11.65
L1: Lsu Writeback Active [%]	7.95
L2: D Sectors [%]	6.10
L1: Data Bank Writes [%]	3.79
L1: Data Bank Reads [%]	3.09
GPU: Compute Memory Access Throughput Internal Activity [%]	3.03
L2: T Tag Requests [%]	3.00
L1: M L1tex2xbar Req Cycles Active [%]	1.93
L2: Xbar2lts Cycles Active [%]	1.89
L1: Texin Sm2tex Req Cycles Active [%]	0.36
L1: F Wavefronts [%]	0.00
L2: D Atomic Input Cycles Active [%]	0
L1: Tex Writeback Active [%]	0
L2: D Sectors Fill Sysmem [%]	0
L1: Data Pipe Tex Wavefronts [%]	0
GPU: Compute Memory Request Throughput Internal Activity [%]	0

PM Sampling

Timeline view of PM metrics sampled periodically over the workload duration. Data is collected across multiple passes. Use this section to understand how workload behavior changes over its runtime.

Maximum Sampling Interval [cycle]	20,000	# Pass Groups	1
Maximum Buffer Size [Kbytes]	512	Dropped Samples [sample]	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]	0.42	SM Busy [%]	11.56
Executed Ipc Active [inst/cycle]	0.46	Issue Slots Busy [%]	11.56
Issued Ipc Active [inst/cycle]	0.46		

Low Utilization
Est. Local Speedup: 93.21% 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.

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/s]	129.44	Mem Busy [%]	16.75
L1/TEX Hit Rate [%]	0.15	Max Bandwidth [%]	40.42
L2 Hit Rate [%]	0.92	Mem Pipes Busy [%]	24.41

Memory L2 Compression The optional metric lts__average_gcomp_input_sector_success_rate.pct could not be found. Collecting it as an additional metric could enable the rule to provide more guidance.

DRAM Global Load Access Pattern
Est. Speedup: 0.09% The memory access pattern for global loads from DRAM might not be optimal. On average, only 31.9 of the 32 bytes transmitted per sector are utilized by each thread. This applies to the 99.7% of sectors missed in L2. This could possibly be caused by a stride between threads. Check the Source Counters section for uncoalesced global loads.

L1TEX Global Store Access Pattern
Est. Speedup: 27.22% The memory access pattern for global stores to L1TEX might not be optimal. On average, only 6.0 of the 32 bytes transmitted per sector are utilized by each thread. This could possibly be caused by a stride between threads. Check the Source Counters section for uncoalesced global stores.

Shared Load Bank Conflicts
Est. Speedup: 4.27% The memory access pattern for shared loads might not be optimal and causes on average a 1.2 - way bank conflict across all 94060 shared load requests.This results in 14219 bank conflicts, which represent 12.75% of the overall 111509 wavefronts for shared loads. Check the Source Counters section for uncoalesced shared loads.

Memory Chart

Show As: Transfer Size

Shared Memory

	Instructions	Requests	Wavefronts	% Peak	Bank Conflicts
Shared Load	94,060	94,060	111,509	3.87	14,219
Shared Load Matrix	0	0			
Shared Store	102,992	102,992	153,209	5.31	0
Shared Atomic	0	0	0	0	0
Other	-	-	97,416	3.38	0
Total	197,052	197,052	362,134	12.56	14,219

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	66,560	66,560	66,560	2.31	394,240	5.92	0.00	12,615,680	394,237	13.67	99,328
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	1,160	1,160	1,160	0.04	1,160	1	50.52	37,120	1,160	0.04	-
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	66,560	66,560	66,560	2.31	394,240	5.92	0.00	12,615,680	394,237	13.67	99,328
Stores	1,160	1,160	1,160	0.04	1,160	1	50.52	37,120	1,160	0.04	-
Atomics & Reductions	0	0	0	0	0	0	0	0	0	0	-
Total	67,720	67,720	67,720	2.35	395,400	5.84	0.15	12,652,800	395,397	13.71	99,328

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	99,322	394,235	3.97	11.66	0.25	12,615,520	102,054,103,028.73	393,300	0	0
L1/TEX Store	1,160	1,160	1	0.03	89.66	37,120	300,284,752.78	120	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	100,482	395,397	3.94	11.69	0.51	12,652,704	102,354,905,513.85	393,420	0	0
ECC Total	-	131	-	0.00	-	4,192	33,911,467.77	131	-	-
GPU Total	101,353	396,717	3.91	11.73	0.92	12,694,944	102,696,608,853.22	393,438	0	0

Device Memory

	Sectors	% Peak	Bytes	Throughput
Load	460,669	37.24	14,741,408	119,251,617,913.54
Store	39,358	3.18	1,259,456	10,188,454,568.99
Total	500,027	40.42	16,000,864	129,440,072,482.53

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]	4.55	No Eligible [%]	82.75
Eligible Warps Per Scheduler [warp]	0.34	One or More Eligible [%]	17.25
Issued Warp Per Scheduler	0.17		

Issue Slot Utilization
Est. Local Speedup: 59.58% Every scheduler is capable of issuing one instruction per cycle, but for this kernel each scheduler only issues an instruction every 5.8 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 4.55 active warps per scheduler, but only an average of 0.34 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]	26.39	Avg. Active Threads Per Warp	30.21
Warp Cycles Per Executed Instruction [cycle]	26.53	Avg. Not Predicated Off Threads Per Warp	26.65

Long Scoreboard Stalls
Est. Speedup: 52.65% On average, each warp of this kernel spends 13.9 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 52.6% of the total average of 26.4 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]	1,214,872	Avg. Executed Instructions Per Scheduler [inst]	7,592.95
Issued Instructions [inst]	1,221,272	Avg. Issued Instructions Per Scheduler [inst]	7,632.95

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	1,024	Function Cache Configuration	CachePreferNone
Registers Per Thread [register/thread]	16	Static Shared Memory Per Block [Kbyte/block]	12.29
Block Size	1,024	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	1,048,576	Driver Shared Memory Per Block [byte/block]	0
Waves Per SM	25.60	Shared Memory Configuration Size [Kbyte]	32.77
Uses Green Context	0	# SMs [SM]	40

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]	4
Theoretical Active Warps per SM [warp]	32	Block Limit Shared Mem [block]	2
Achieved Occupancy [%]	38.89	Block Limit Warps [block]	1
Achieved Active Warps Per SM [warp]	12.44	Block Limit SM [block]	16

Achieved Occupancy
Est. Speedup: 59.58% The difference between calculated theoretical (100.0%) and measured achieved occupancy (38.9%) 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.

GPU and Memory Workload Distribution

Analysis of workload distribution in active cycles of SM, SMP, SMSP, L1 & L2 caches, and DRAM

Average SM Active Cycles [cycle]	66,030.25	Average L1 Active Cycles [cycle]	66,030.25
Average L2 Active Cycles [cycle]	74,383.84	Average SMSP Active Cycles [cycle]	44,255.52
Average DRAM Active Cycles [cycle]	250,013.50	Total SM Elapsed Cycles [cycle]	2,883,992
Total L1 Elapsed Cycles [cycle]	2,883,992	Total L2 Elapsed Cycles [cycle]	3,381,088
Total SMSP Elapsed Cycles [cycle]	11,535,968	Total DRAM Elapsed Cycles [cycle]	4,947,968

Workload Distribution

	Average	Min	Max	Sum
SM Active Cycles	66,030.25	65,113	66,895	2,641,210
SMSP Active Cycles	44,255.52	42,480	45,994	7,080,883
L1 Active Cycles	66,030.25	65,113	66,895	2,641,210
L2 Active Cycles	74,383.84	72,251	75,951	2,380,283
DRAM Active Cycles	250,013.50	249,468	251,028	2,000,108

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]	145,988	Branch Efficiency [%]	92.65
Branch Instructions Ratio [%]	0.12	Avg. Divergent Branches	32

Uncoalesced Shared Accesses
Est. Speedup: 10.59% This kernel has uncoalesced shared accesses resulting in a total of 30622 excessive wavefronts (12% of the total 264718 wavefronts). Check the L1 Wavefronts Shared Excessive table for the primary source locations. The CUDA Best Practices Guide has an example on optimizing shared memory accesses.

L1 Wavefronts Shared Excessive

Location	Value	Value (%)
kernels.cu:219 (0x7930d703a500 in reduce_argmin)	15,311	50
kernels.cu:219 (0x7930d703a4f0 in reduce_argmin)	15,311	50
kernels.cu:227 (0x7930d703a5a0 in reduce_argmin)	0	0
kernels.cu:225 (0x7930d703a550 in reduce_argmin)	0	0
kernels.cu:218 (0x7930d703a4d0 in reduce_argmin)	0	0

Warp Stall Sampling (All Samples)

Location	Value	Value (%)
kernels.cu:217 (0x7930d703a490 in reduc...	622	24
kernels.cu:217 (0x7930d703a4b0 in reduc...	286	11
kernels.cu:208 (0x7930d703a410 in reduc...	269	10
kernels.cu:219 (0x7930d703a500 in reduc...	249	10
kernels.cu:225 (0x7930d703a580 in reduc...	182	7

Most Instructions Executed

Location	Value	Value (%)
kernels.cu:225 (0x7930d703a530 in reduc...	37,888	3
kernels.cu:213 (0x7930d703a520 in reduc...	36,864	3
kernels.cu:213 (0x7930d703a510 in reduc...	36,864	3
kernels.cu:219 (0x7930d703a500 in reduc...	36,864	3
kernels.cu:219 (0x7930d703a4f0 in reduce...	36,864	3

Follow the *rules outputs* to get guidance on how to navigate through the report and quickly discover performance bottlenecks in this kernel.
You could also disable individual sections to focus on selected performance aspects and make profiling faster.