Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current 550 - brute_force	(1024, 1, 1)x(1024, 1, 1)	4.23 ms	2,473,819	0 - Tesla T4	584.99 Mhz	[19994] 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 [%]	44.32	Duration [ms]	4.23
Memory Throughput [%]	38.88	Elapsed Cycles [cycle]	2,473,819
L1/TEX Cache Throughput [%]	50.78	SM Active Cycles [cycle]	2,408,916
L2 Cache Throughput [%]	18.32	SM Frequency [Mhz]	584.99
DRAM Throughput [%]	30.42	DRAM Frequency [Ghz]	5.01

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 4% of this device's fp32 peak performance and 9% of its fp64 peak performance. See the Kernel Profiling Guide for more details on roofline analysis.

GPU Throughput

Compute Throughput Breakdown

SM: Pipe Fp64 Cycles Active [%]	44.32
SM: Inst Executed Pipe Lsu [%]	38.88
SM: Issue Active [%]	30.39
SM: Inst Executed [%]	30.39
SM: Pipe Alu Cycles Active [%]	20.89
SM: Pipe Fma Cycles Active [%]	20.28
SM: Mio Inst Issued [%]	14.88
SM: Inst Executed Pipe Xu [%]	13.66
SM: Mio Pq Write Cycles Active [%]	10.25
SM: Mio2rf Writeback Active [%]	9.00
SM: Mio Pq Read Cycles Active [%]	3.38
SM: Inst Executed Pipe Cbu Pred On Any [%]	2.42
SM: Inst Executed Pipe Adu [%]	0.20
IDC: Request Cycles Active [%]	0.14
SM: Inst Executed Pipe Uniform [%]	0.06
SM: Pipe Tensor Cycles Active [%]	0
SM: Inst Executed Pipe Fp16 [%]	0
SM: Pipe Shared Cycles Active [%]	0
SM: Memory Throughput Internal Activity [%]	0
SM: Instruction Throughput Internal Activity [%]	0
SM: Inst Executed Pipe Tex [%]	0
SM: Inst Executed Pipe Ipa [%]	0

Memory Throughput Breakdown

L1: Lsuin Requests [%]	38.88
DRAM: Cycles Active [%]	30.42
L1: Data Pipe Lsu Wavefronts [%]	25.39
DRAM: Dram Sectors [%]	22.30
L1: M L1tex2xbar Req Cycles Active [%]	21.10
L2: T Sectors [%]	18.32
L2: Xbar2lts Cycles Active [%]	18.06
L1: Lsu Writeback Active [%]	17.91
L2: D Sectors [%]	7.00
L1: Data Bank Reads [%]	6.80
GPU: Compute Memory Access Throughput Internal Activity [%]	4.81
L2: T Tag Requests [%]	4.78
L1: Data Bank Writes [%]	2.68
L1: M Xbar2l1tex Read Sectors [%]	1.57
L2: Lts2xbar Cycles Active [%]	1.35
L2: D Sectors Fill Device [%]	0.58
L1: Texin Sm2tex Req Cycles Active [%]	0.00
L1: F Wavefronts [%]	0.00
GPU: Compute Memory Request Throughput Internal Activity [%]	0
L1: Tex Writeback Active [%]	0
L2: D Atomic Input Cycles Active [%]	0
L2: D Sectors Fill Sysmem [%]	0
L1: Data Pipe Tex Wavefronts [%]	0

Floating Point Operations Roofline

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]	160,000	# Pass Groups	1
Maximum Buffer Size [Mbytes]	3.81	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]	1.22	SM Busy [%]	45.50
Executed Ipc Active [inst/cycle]	1.25	Issue Slots Busy [%]	31.20
Issued Ipc Active [inst/cycle]	1.25		

Balanced FP64 is the highest-utilized pipeline (45.5%) based on active cycles, taking into account the rates of its different instructions. It executes 64-bit floating point operations. It is well-utilized, but should not be a bottleneck.

Pipe Utilization (% of active cycles)

Pipe Utilization (% of peak instructions executed)

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]	97.61	Mem Busy [%]	25.39
L1/TEX Hit Rate [%]	84.90	Max Bandwidth [%]	38.88
L2 Hit Rate [%]	96.83	Mem Pipes Busy [%]	38.88

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.

L1TEX Global Load Access Pattern Est. Speedup: 45.24% The memory access pattern for global loads from L1TEX might not be optimal. On average, only 3.5 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 loads.

L1TEX Local Load Access Pattern Est. Speedup: 50.03% The memory access pattern for local loads from L1TEX might not be optimal. On average, only 0.5 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 local loads.

L1TEX Local Store Access Pattern Est. Speedup: 49.28% The memory access pattern for local stores to L1TEX might not be optimal. On average, only 0.9 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 local stores.

Memory Chart

Show As: Transfer Size

Shared Memory

	Instructions	Requests	Wavefronts	% Peak	Bank Conflicts
Shared Load	0	0	0	0	0
Shared Load Matrix	0	0			
Shared Store	0	0	0	0	0
Shared Atomic	0	0	0	0	0
Other	-	-	339,968	0.34	0
Total	0	0	339,968	0.34	0

L1/TEX Cache

	Instructions	Requests	Wavefronts	% Peak	Sectors	Sectors/Req	Hit Rate	Bytes	Sector Misses to L2	% Peak to L2	Returns to SM
Local Load	8,583,168	9,906,176	9,906,176	10.01	37,827,648	3.82	95.95	1,210,484,736	1,558,730	1.58	17,680,384
Global Load	7,774,208	7,774,208	7,774,208	7.86	7,774,208	1	99.68	248,774,656			
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	32,768	32,768	32,768	0.03	131,072	4	0	4,194,304	19,650,636	19.86	-
Local Store	2,502,656	5,398,528	5,398,528	5.46	20,801,536	3.85	59.82	665,649,152			-
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	16,357,376	17,680,384	17,680,384	17.87	45,601,856	2.58	96.59	1,459,259,392	1,558,730	1.58	17,680,384
Stores	2,535,424	5,431,296	5,431,296	5.49	20,932,608	3.85	59.44	669,843,456	19,650,636	19.86	-
Atomics & Reductions	0	0	0	0	0	0	0	0	0	0	-

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	419,666	1,559,244	3.72	1.35	77.00	49,895,808	11,799,046,538.02	357,218	0	0
L1/TEX Store	5,110,855	19,635,179	3.84	16.97	98.42	628,325,728	148,582,512,296.63	322,180	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	5,526,534	21,210,414	3.84	18.33	96.81	678,733,248	160,502,565,266.74	691,480	0	0
ECC Total	-	330,599	-	0.04	-	10,579,168	2,501,695,043.51	330,599	-	-
GPU Total	5,528,831	21,200,623	3.83	18.32	96.73	678,419,936	160,428,475,217.56	680,579	0	0

Device Memory

	Sectors	% Peak	Bytes	Throughput
Load	1,007,113	2.38	32,227,616	7,620,983,730.61
Store	11,892,637	28.05	380,564,384	89,993,469,542.19
Total	12,899,750	30.42	412,792,000	97,614,453,272.80

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]	7.79	No Eligible [%]	67.54
Eligible Warps Per Scheduler [warp]	1.07	One or More Eligible [%]	32.46
Issued Warp Per Scheduler	0.32		

Issue Slot Utilization Est. Local Speedup: 55.68% Every scheduler is capable of issuing one instruction per cycle, but for this kernel each scheduler only issues an instruction every 3.1 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 7.79 active warps per scheduler, but only an average of 1.07 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.

Warps Per Scheduler

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]	23.99	Avg. Active Threads Per Warp	25.81
Warp Cycles Per Executed Instruction [cycle]	23.99	Avg. Not Predicated Off Threads Per Warp	24.61

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]	120,243,488	Avg. Executed Instructions Per Scheduler [inst]	751,521.80
Issued Instructions [inst]	120,249,157	Avg. Issued Instructions Per Scheduler [inst]	751,557.23

FP32 Non-Fused Instructions Est. Speedup: 5.20% This kernel executes 6604800 fused and 6604800 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 25% (relative to its current performance). Check the Source page to identify where this kernel executes FP32 instructions.

FP64 Non-Fused Instructions Est. Speedup: 22.75% This kernel executes 0 fused and 1323008 non-fused FP64 instructions. By converting pairs of non-fused instructions to their fused, higher-throughput equivalent, the achieved FP64 performance could be increased by up to 50% (relative to its current performance). Check the Source page to identify where this kernel executes FP64 instructions.

Executed Instruction Mix

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]	33	Static Shared Memory Per Block [byte/block]	0
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]	1
Theoretical Active Warps per SM [warp]	32	Block Limit Shared Mem [block]	16
Achieved Occupancy [%]	93.47	Block Limit Warps [block]	1
Achieved Active Warps Per SM [warp]	29.91	Block Limit SM [block]	16

Impact of Varying Register Count Per Thread

Impact of Varying Block Size

Impact of Varying Shared Memory Usage Per Block

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]	2,408,916	Average L1 Active Cycles [cycle]	2,408,916
Average L2 Active Cycles [cycle]	2,945,925.22	Average SMSP Active Cycles [cycle]	2,315,669.29
Average DRAM Active Cycles [cycle]	6,449,875	Total SM Elapsed Cycles [cycle]	98,921,872
Total L1 Elapsed Cycles [cycle]	98,921,872	Total L2 Elapsed Cycles [cycle]	115,697,920
Total SMSP Elapsed Cycles [cycle]	395,687,488	Total DRAM Elapsed Cycles [cycle]	169,606,144

Workload Distribution

	Average	Min	Max	Sum
SM Active Cycles	2,408,916	2,358,789	2,468,585	96,356,640
SMSP Active Cycles	2,315,669.29	2,244,087	2,399,226	370,507,086
L1 Active Cycles	2,408,916	2,358,789	2,468,585	96,356,640
L2 Active Cycles	2,945,925.22	2,849,307	3,051,629	94,269,607
DRAM Active Cycles	6,449,875	6,324,092	6,607,336	51,599,000

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]	8,404,256	Branch Efficiency [%]	97.34
Branch Instructions Ratio [%]	0.07	Avg. Divergent Branches	1,237.40

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.