	Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current	677 - brute_force_AL_coarsening	(32768, 1, 1)x(32, 1, 1)	2.16 ms	1,263,267	0 - Tesla T4	584.98 Mhz	[5967] 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 [%]	82.01	Duration [ms]	2.16
Memory Throughput [%]	2.01	Elapsed Cycles [cycle]	1,263,267
L1/TEX Cache Throughput [%]	1.56	SM Active Cycles [cycle]	1,258,336.57
L2 Cache Throughput [%]	0.67	SM Frequency [Mhz]	584.98
DRAM Throughput [%]	2.01	DRAM Frequency [Ghz]	5.00

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.

FP64/32 Utilization
Est. Speedup: 53.60% The ratio of peak float (fp32) to double (fp64) performance on this device is 32:1. The kernel achieved close to 1% of this device's fp32 peak performance and 61% of its fp64 peak performance. If Compute Workload Analysis determines that this kernel is fp64 bound, consider using 32-bit precision floating point operations to improve its performance. See the Kernel Profiling Guide for more details on roofline analysis.

GPU Throughput

Compute Throughput Breakdown

SM: Pipe Fp64 Cycles Active [%]	82.01
SM: Pipe Alu Cycles Active [%]	43.55
SM: Issue Active [%]	34.20
SM: Inst Executed [%]	34.20
SM: Mio Pq Write Cycles Active [%]	26.94
SM: Inst Executed Pipe Adu [%]	19.73
IDC: Request Cycles Active [%]	11.33
SM: Pipe Fma Cycles Active [%]	10.23
SM: Mio Inst Issued [%]	10.12
SM: Mio Pq Read Cycles Active [%]	5.92
SM: Inst Executed Pipe Cbu Pred On Any [%]	4.61
SM: Mio2rf Writeback Active [%]	2.90
SM: Inst Executed Pipe Lsu [%]	0.39
SM: Inst Executed Pipe Xu [%]	0.13
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 Uniform [%]	0
SM: Inst Executed Pipe Tex [%]	0
SM: Inst Executed Pipe Ipa [%]	0

Memory Throughput Breakdown

DRAM: Cycles Active [%]	2.01
DRAM: Dram Sectors [%]	1.47
L1: M L1tex2xbar Req Cycles Active [%]	0.78
L2: T Sectors [%]	0.67
L2: Xbar2lts Cycles Active [%]	0.67
L1: Lsuin Requests [%]	0.39
L2: D Sectors [%]	0.32
L1: Data Pipe Lsu Wavefronts [%]	0.28
L2: T Tag Requests [%]	0.17
GPU: Compute Memory Access Throughput Internal Activity [%]	0.17
L1: Data Bank Reads [%]	0.10
L1: Data Bank Writes [%]	0.10
L1: Lsu Writeback Active [%]	0.06
L2: D Sectors Fill Device [%]	0.01
L2: Lts2xbar Cycles Active [%]	0.00
L1: Texin Sm2tex Req Cycles Active [%]	0.00
L1: F Wavefronts [%]	0.00
L1: M Xbar2l1tex Read Sectors [%]	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
GPU: Compute Memory Request Throughput Internal Activity [%]	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]	40,000	# Pass Groups	1
Maximum Buffer Size [Mbytes]	3.06	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]	1.37	SM Busy [%]	82.29
Executed Ipc Active [inst/cycle]	1.37	Issue Slots Busy [%]	34.32
Issued Ipc Active [inst/cycle]	1.37		

Very High Utilization FP64 is the highest-utilized pipeline (82.3%) based on active cycles, taking into account the rates of its different instructions. It executes 64-bit floating point operations. The pipeline is over-utilized and likely a performance bottleneck. Based on the number of executed instructions, the highest utilized pipeline (82.3%) is FP64. It executes 64-bit floating point operations. Comparing the two, the overall pipeline utilization appears to be caused by frequent, low-latency instructions. 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 (% 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]	6.42	Mem Busy [%]	0.67
L1/TEX Hit Rate [%]	0	Max Bandwidth [%]	2.01
L2 Hit Rate [%]	99.19	Mem Pipes Busy [%]	26.94

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.

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	-	-	32,768	0.06	0
Total	0	0	32,768	0.06	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	0	0	0	0	0	0	0	0	0	0	0
Global Load	0	0	0	0	0	0	0	0			
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	65,536	65,536	65,536	0.13	393,216	6	0	12,582,912	393,216	0.78	-
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	0	0	0	0	0	0	0	0	0	0	0
Stores	65,536	65,536	65,536	0.13	393,216	6	0	12,582,912	393,216	0.78	-
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	0	0	0	0	0	0	0	0	0	0
L1/TEX Store	98,304	393,216	4	0.67	100	12,582,912	5,826,803,390.43	0	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	98,304	393,216	4	0.67	100	12,582,912	5,826,803,390.43	0	0	0
ECC Total	-	32	-	0.00	-	1,024	474,186.47	32	-	-
GPU Total	100,008	397,868	3.98	0.67	99.78	12,731,776	5,895,738,249.07	44	0	0

Device Memory

	Sectors	% Peak	Bytes	Throughput
Load	360	0.00	11,520	5,334,597.83
Store	432,660	2.00	13,845,120	6,411,297,492.74
Total	433,020	2.01	13,856,640	6,416,632,090.57

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]	3.95	No Eligible [%]	65.62
Eligible Warps Per Scheduler [warp]	0.48	One or More Eligible [%]	34.38
Issued Warp Per Scheduler	0.34		

Issue Slot Utilization
Est. Local Speedup: 17.99% Every scheduler is capable of issuing one instruction per cycle, but for this kernel each scheduler only issues an instruction every 2.9 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 3.95 active warps per scheduler, but only an average of 0.48 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]	11.50	Avg. Active Threads Per Warp	24.77
Warp Cycles Per Executed Instruction [cycle]	11.50	Avg. Not Predicated Off Threads Per Warp	22.48

Long Scoreboard Stalls
Est. Speedup: 17.99% On average, each warp of this kernel spends 5.7 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 49.4% of the total average of 11.5 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.

Thread Divergence
Est. Speedup: 24.41% Instructions are executed in warps, which are groups of 32 threads. Optimal instruction throughput is achieved if all 32 threads of a warp execute the same instruction. The chosen launch configuration, early thread completion, and divergent flow control can significantly lower the number of active threads in a warp per cycle. This kernel achieves an average of 24.8 threads being active per cycle. This is further reduced to 22.5 threads per warp due to predication. The compiler may use predication to avoid an actual branch. Instead, all instructions are scheduled, but a per-thread condition code or predicate controls which threads execute the instructions. Try to avoid different execution paths within a warp when possible.

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]	69,091,328	Avg. Executed Instructions Per Scheduler [inst]	431,820.80
Issued Instructions [inst]	69,093,412	Avg. Issued Instructions Per Scheduler [inst]	431,833.83

FP32 Non-Fused Instructions
Est. Speedup: 5.13% This kernel executes 0 fused and 4915200 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 50% (relative to its current performance). Check the Source page to identify where this kernel executes FP32 instructions.

FP64 Non-Fused Instructions
Est. Speedup: 19.24% This kernel executes 1343488 fused and 1179648 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 23% (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	32,768	Function Cache Configuration	CachePreferNone
Registers Per Thread [register/thread]	26	Static Shared Memory Per Block [byte/block]	0
Block Size	32	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	1,048,576	Driver Shared Memory Per Block [byte/block]	0
Waves Per SM	51.20	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 [%]	50	Block Limit Registers [block]	64
Theoretical Active Warps per SM [warp]	16	Block Limit Shared Mem [block]	16
Achieved Occupancy [%]	49.37	Block Limit Warps [block]	32
Achieved Active Warps Per SM [warp]	15.80	Block Limit SM [block]	16

Theoretical Occupancy
Est. Speedup: 17.99% The 4.00 theoretical warps per scheduler this kernel can issue according to its occupancy are below the hardware maximum of 8. This kernel's theoretical occupancy (50.0%) is limited by the number of blocks that can fit on the SM. This kernel's theoretical occupancy (50.0%) is limited by the required amount of shared memory.

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]	1,258,336.57	Average L1 Active Cycles [cycle]	1,258,336.57
Average L2 Active Cycles [cycle]	111,777.84	Average SMSP Active Cycles [cycle]	1,256,019.22
Average DRAM Active Cycles [cycle]	216,510	Total SM Elapsed Cycles [cycle]	50,507,000
Total L1 Elapsed Cycles [cycle]	50,507,000	Total L2 Elapsed Cycles [cycle]	59,081,792
Total SMSP Elapsed Cycles [cycle]	202,028,000	Total DRAM Elapsed Cycles [cycle]	86,345,728

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]	5,619,712	Branch Efficiency [%]	96.54
Branch Instructions Ratio [%]	0.08	Avg. Divergent Branches	1,024

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.