	Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current	529 - simulate_k	(8192, 1, 1)x(256, 1, 1)	718.87 ms	981,259,420	0 - NVIDIA GeForce RTX 2080 Ti	1.36 Ghz	[1406631] headless	

Summary | Details | Source | Context | Comments | Raw | Session — Compare | Tools | View | Export

GPU Speed Of Light Throughput

GPU Throughput Chart

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 presents a roofline chart.

Compute (SM) Throughput [%]	68.42	Duration [ms]	718.87
Memory Throughput [%]	54.43	Elapsed Cycles [cycle]	981,259,420
L1/TEX Cache Throughput [%]	96.16	SM Active Cycles [cycle]	976,200,051.60
L2 Cache Throughput [%]	0.00	SM Frequency [Ghz]	1.36
DRAM Throughput [%]	0.00	DRAM Frequency [Ghz]	6.79

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.

Key Performance Indicators

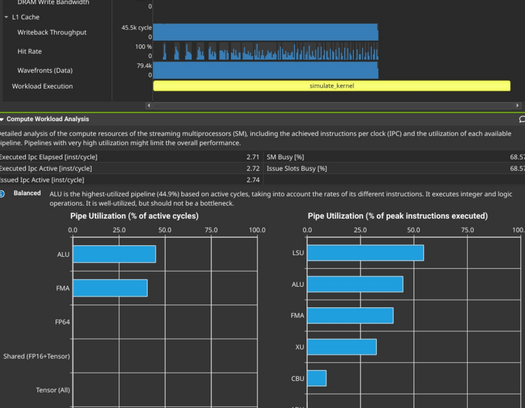
Metric Name	Value	Guidance
sm_throughput.avg.pct_of_peak_sustained_elapsed	54.4341	68.419 - 54.434 >= 10.000

Roofline Analysis — The ratio of peak float (FP32) to double (FP64) performance on this device is 32:1. The workload achieved 15% of this device's FP32 peak performance and 0% of its FP64 peak performance. See the Profiling Guide for more details on roofline analysis.

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]	33.69	Dropped Samples [sample]	0

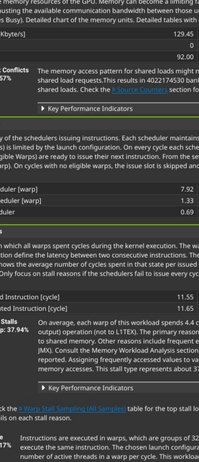


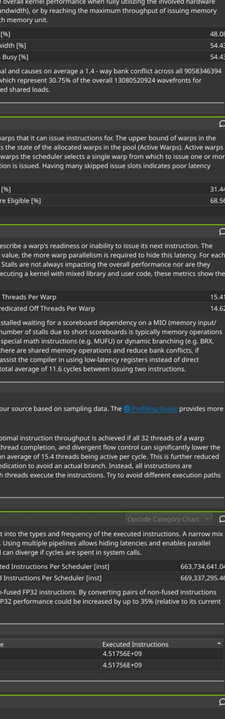
Compute Workload Analysis

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]	2.71	SM Busy [%]	68.57
Executed Ipc Active [inst/cycle]	2.72	Issue Slots Busy [%]	68.57
Issued Ipc Active [inst/cycle]	2.74		

Balanced — ALU is the highest-utilized pipeline (44.9%) based on active cycles, taking into account the rates of its different instructions. It executes integer and logic operations. It is well-utilized, but should not be a bottleneck.





Memory Workload Analysis

Memory Chart

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 [Mbytes/s]	129.45	Mem Busy [%]	48.08
L1/TEX Hit Rate [%]	0	Max Bandwidth [%]	54.43
L2 Hit Rate [%]	92.00	Mem Pipes Busy [%]	54.43

Shared Load Bank Conflicts — The memory access pattern for shared loads might not be optimal and causes on average a 1.4 - way bank conflict across all 9058346394 shared load requests.This results in 4022174530 bank conflicts, which represent 30.75% of the overall 13080520924 wavefronts for shared loads. Check the Source Counters section for uncoalesced shared loads.

Key Performance Indicators

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]	7.92	No Eligible [%]	31.44
Eligible Warps Per Scheduler [warp]	1.33	One or More Eligible [%]	68.56
Issued Warp Per Scheduler	0.69		

Short Scoreboard Stalls — On average, each warp of this workload spends 4.4 cycles being stalled waiting for a scoreboard dependency on a MIO (memory input/output) operation (not to L1TEX). The primary reason for a high number of stalls due to short scoreboards is typically memory operations to shared memory. Other reasons include frequent execution of special math instructions (e.g. MUFU) or dynamic branching (e.g. BRX, JMX). Consult the Memory Workload Analysis section to verify if there are shared memory operations and reduce bank conflicts, if reported. Assigning frequently accessed values to variables can assist the compiler in using low-latency registers instead of direct memory accesses. This stall type represents about 37.9% of the total average of 11.6 cycles between issuing two instructions.

Key Performance Indicators

Warp Stall — Check the Issue Stall Sampling (All Samples) table for the top stall locations in your source based on sampling data. The Profiling Guide provides more details on each stall reason.

Thread Divergence — 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 workload achieves an average of 15.4 threads being active per cycle. This is further reduced to 14.6 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.

Key Performance Indicators

Instruction Statistics

Opcode Category Chart

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]	180,535,822,364	Avg. Executed Instructions Per Scheduler [inst]	663,734,641.04
Issued Instructions [inst]	182,059,744,365	Avg. Issued Instructions Per Scheduler [inst]	669,337,295.46

FP32 Non-Fused Instructions — This kernel executes 15371176815 fused and 37260660485 non-fused FP32 instructions. By converting pairs of non-fused instructions to their FFMA higher-throughput equivalent, the achieved FP32 performance could be increased by up to 35% (relative to its current performance).

Most frequently executed non-fused FP32 instructions

Location	Opcode	Executed Instructions
photon.cuh, line 40	FADD	4.51756E+09
photon.cuh, line 38	FADD	4.51756E+09

Key Performance Indicators

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	8,192	Function Cache Configuration	CachePreferNone
Registers Per Thread [register/thread]	56	Static Shared Memory Per Block [Kbyte/block]	6.46
Block Size	256	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	2,097,152	Driver Shared Memory Per Block [byte/block]	0
Waves Per SM	30.12	Shared Memory Configuration Size [Kbyte]	32.77
Uses Green Context	0	Stack Size	1,024
# SMs [SM]	68	# TPCs	34
Enabled TPC IDs	all	-	-

Occupancy

% Occupancy Graphs

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]	4
Achieved Occupancy [%]	99.00	Block Limit Warps [block]	4
Achieved Active Warps Per SM [warp]	31.68	Block Limit SM [block]	16

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]	976,200,051.60	Average L1 Active Cycles [cycle]	976,200,051.60
Average L2 Active Cycles [cycle]	43,495.93	Average SMSP Active Cycles [cycle]	976,248,651.65
Average DRAM Active Cycles [cycle]	969.33	Total SM Elapsed Cycles [cycle]	66,523,356,546
Total L1 Elapsed Cycles [cycle]	66,523,356,546	Total L2 Elapsed Cycles [cycle]	40,803,133,756
Total SMSP Elapsed Cycles [cycle]	266,093,426,184	Total DRAM Elapsed Cycles [cycle]	58,584,453,120

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]	28,146,802,683	Branch Efficiency [%]	41.97
Branch Instructions Ratio [%]	0.16	Avg. Divergent Branches	31,711,839.45

Uncoalesced Shared Accesses — This kernel has uncoalesced shared accesses resulting in a total of 3590537398 excessive wavefronts (12% of the total 30731219332 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.

Key Performance Indicators

L1 Wavefronts Shared Excessive

Location	Value	Value (%)
photon.cuh:40 (0x7effbfa33590 in simulate_kernel)	1,795,268,699	50
photon.cuh:38 (0x7effbfa33500 in simulate_kernel)	1,795,268,699	50
tiny_mc.cu:52 (0x7effbfa35e00 in simulate_kernel)	0	0
tiny_mc.cu:51 (0x7effbfa35da0 in simulate_kernel)	0	0
tiny_mc.cu:43 (0x7effbfa35d90 in simulate_kernel)	0	0