	Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current	574 - mmul	(32, 8, 1)x(32, 32, 1)	374.53 us	578,559	0 - NVIDIA GeForce RTX 4060 Laptop GPU	1.54 Ghz	[47102] mmul_shared_mem	

Summary
Details
Source
Context
Comments
Raw
Session
Compare
Tools
View
Export

GPU Speed Of Light Throughput

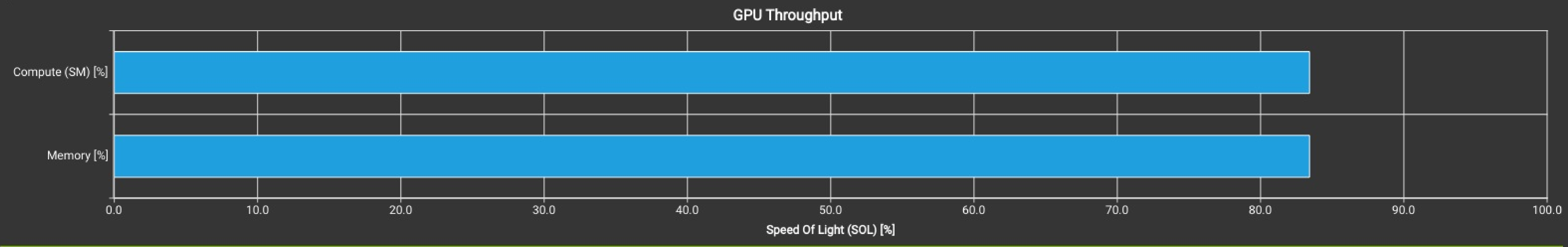
GPU Throughput Chart

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 presented as a roofline chart.

Compute (SM) Throughput [%]	83.42	Duration [us]	374.53
Memory Throughput [%]	83.42	Elapsed Cycles [cycle]	578,559
L1/TEX Cache Throughput [%]	86.68	SM Active Cycles [cycle]	556,833.96
L2 Cache Throughput [%]	6.25	SM Frequency [Ghz]	1.54
DRAM Throughput [%]	2.74	DRAM Frequency [Ghz]	7.99

High Throughput This workload is utilizing greater than 80.0% of the available compute or memory performance of this 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.

Roofline Analysis The ratio of peak float (FP32) to double (FP64) performance on this device is 64:1. The workload achieved 0% 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 [us]	1	# Pass Groups	3
Maximum Buffer Size [Mbyte]	8.39	-	-

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]	0.83	SM Busy [%]	29.14
Executed Ipc Active [inst/cycle]	0.86	Issue Slots Busy [%]	20.72
Issued Ipc Active [inst/cycle]	0.86		

Low Utilization
Est. Local Speedup: 91.61%
All compute pipelines are under-utilized. Either this workload is very small or it doesn't issue enough warps per scheduler. Check the Launch Statistics and Scheduler Statistics sections for further details.

Key Performance Indicators

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]	7.01	Mem Busy [%]	49.98
L1/TEX Hit Rate [%]	1.65	Max Bandwidth [%]	83.42
L2 Hit Rate [%]	92.28	Mem Pipes Busy [%]	83.42
L2 Compression Input Sectors [sector]	0	Local Memory Spilling Requests	0
L2 Compression Ratio	0	Shared Memory Spilling Requests	0
L2 Compression Success Rate [%]	0	Local Memory Spilling Request Overhead [%]	0
L2 Persisting Size [Mbyte]	6.29	Shared Memory Spilling Request Overhead [%]	0
L2 Sector Promotion Misses [%]	0	-	-

Shared Store Bank Conflicts
Est. Speedup: 19.23%
The memory access pattern for shared stores might not be optimal and causes on average a 1.3 - way bank conflict across all 262144 shared store requests.This results in 74734 bank conflicts, which represent 22.18% of the overall 336878 wavefronts for shared stores. Check the Source Counters section for uncoalesced shared stores.

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]	7.99	No Eligible [%]	78.47
Eligible Warps Per Scheduler [warp]	0.98	One or More Eligible [%]	21.53
Issued Warp Per Scheduler	0.22		

Issue Slot Utilization
Est. Local Speedup: 16.58%
Every scheduler is capable of issuing one instruction per cycle, but for this workload each scheduler only issues an instruction every 4.6 cycles. This might leave hardware resources underutilized and may lead to less optimal performance. Out of the maximum of 12 warps per scheduler, this workload allocates an average of 7.99 active warps per scheduler, but only an average of 0.98 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.

Key Performance Indicators

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]	37.10	Avg. Active Threads Per Warp	32
Warp Cycles Per Executed Instruction [cycle]	37.12	Avg. Not Predicated Off Threads Per Warp	31.93

Mio Throttle Stalls
Est. Speedup: 16.58%
On average, each warp of this workload spends 23.2 cycles being stalled waiting for the MIO (memory input/output) instruction queue to be not full. This stall reason is high in cases of extreme 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. This stall type represents about 62.5% of the total average of 37.1 cycles between issuing two instructions.

Key Performance Indicators

Warp Stall Check the Warp 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.

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]	11,501,568	Avg. Executed Instructions Per Scheduler [inst]	119,808
Issued Instructions [inst]	11,506,722	Avg. Issued Instructions Per Scheduler [inst]	119,861.69
Local Memory Spilling Requests [inst]	0	Shared Memory Spilling Requests [inst]	0

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	256	Function Cache Configuration	CachePreferNone
Registers Per Thread [register/thread]	36	Static Shared Memory Per Block [Kbyte/block]	8.19
Block Size	1,024	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	262,144	Driver Shared Memory Per Block [Kbyte/block]	1.02
Waves Per SM	10.67	Shared Memory Configuration Size [Kbyte]	16.38
Uses Green Context	0	Stack Size	1,024
# SMs [SM]	24	# TPCs	12
Enabled TPC IDs	all	-	-

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 [%]	66.67	Block Limit Registers [block]	1
Theoretical Active Warps per SM [warp]	32	Block Limit Shared Mem [block]	1
Achieved Occupancy [%]	66.72	Block Limit Warps [block]	1
Achieved Active Warps Per SM [warp]	32.02	Block Limit SM [block]	24

Theoretical Occupancy
Est. Speedup: 16.58%
The 8.00 theoretical warps per scheduler this kernel can issue according to its occupancy are below the hardware maximum of 12. This kernel's theoretical occupancy (66.7%) is limited by the number of required registers, and the number of warps within each block.

Key Performance Indicators

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]	556,833.96	Average L1 Active Cycles [cycle]	556,833.96
Average L2 Active Cycles [cycle]	196,724.19	Average SMSP Active Cycles [cycle]	556,726.08
Average DRAM Active Cycles [cycle]	82,052	Total SM Elapsed Cycles [cycle]	13,885,336
Total L1 Elapsed Cycles [cycle]	13,885,336	Total L2 Elapsed Cycles [cycle]	8,537,920
Total SMSP Elapsed Cycles [cycle]	55,541,344	Total DRAM Elapsed Cycles [cycle]	11,972,608

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]	155,648	Branch Efficiency [%]	100
Branch Instructions Ratio [%]	0.01	Avg. Divergent Branches [branches]	0

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.