	Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current	1373 - gemv_nvfp4_kernel	(4096, 8, 1)x(32, 1, 1)	296.74 us	339,579	0 - NVIDIA B200	1.12 Ghz	[317650] python3.10	

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 [%]	72.88	Duration [us]	296.74
Memory Throughput [%]	63.69	Elapsed Cycles [cycle]	339,579
L1/TEX Cache Throughput [%]	65.37	SM Active Cycles [cycle]	324,295.19
L2 Cache Throughput [%]	6.55	SM Frequency [Ghz]	1.12
DRAM Throughput [%]	6.90	DRAM Frequency [Ghz]	3.99

Balanced Throughput Compute and Memory are well-balanced: To reduce runtime, both computation and memory traffic must be reduced. Check both the Compute Workload Analysis and Memory Workload Analysis sections.

Roofline Analysis The ratio of peak float (FP32) to double (FP64) performance on this device is 2:1. The workload achieved 7% 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	1
Maximum Buffer Size [Mbyte]	60.95	-	-

Compute Workload Analysis

Pipe Utilization (Elapsed Cycles)

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]	2.92	SM Busy [%]	72.88
Executed Ipc Active [inst/cycle]	2.99	Issue Slots Busy [%]	72.88
Issued Ipc Active [inst/cycle]	2.99		

High Utilization ALU is the highest-utilized pipeline (68.8%) based on elapsed cycles in the workload, taking into account the rates of its different instructions. It executes integer and logic operations. The pipeline is well-utilized, but might become a bottleneck if more work is added. Based on the number of executed instructions, the highest utilized pipeline (68.8%) is ALU. It executes integer and logic operations. Comparing the two, the overall pipeline utilization appears to be caused by frequent, low-latency instructions. See the 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 for this workload. Check the Warp State Statistics section for which reasons cause warps to stall.

Key Performance Indicators

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 [Gbyte/s]	458.96	Mem Busy [%]	63.69
L1/TEX Hit Rate [%]	96.36	Max Bandwidth [%]	32.99
L2 Hit Rate [%]	7.51	Mem Pipes Busy [%]	32.99
L2 Compression Input Sectors [sector]	34,599	Local Memory Spilling Requests	0
L2 Compression Ratio [%]	0	Local Memory Spilling Request Overhead [%]	0
L2 Compression Success Rate [%]	0	L2 Persisting Size [Mbyte]	24.87

Low Compression Rate
Est. Speedup: 5.78%
Out of the 1107168.0 bytes sent to the L2 Compression unit only 0.00% were successfully compressed. To increase this success rate, consider marking only those memory regions as compressible that contain the most zero values and/or expose the most homogeneous values.

Key Performance Indicators

L1TEX Global Load Access Pattern
Est. Speedup: 55.30%
The memory access pattern for global loads from L1TEX might not be optimal. On average, only 4.2 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.

Key Performance Indicators

L1TEX Global Store Access Pattern
Est. Speedup: 59.71%
The memory access pattern for global stores to L1TEX might not be optimal. On average, only 2.0 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.

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]	7.54	No Eligible [%]	24.64
Eligible Warps Per Scheduler [warp]	2.08	One or More Eligible [%]	75.36
Issued Warp Per Scheduler	0.75		

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]	10.00	Avg. Active Threads Per Warp	31.91
Warp Cycles Per Executed Instruction [cycle]	10.00	Avg. Not Predicated Off Threads Per Warp	30.23

Long Scoreboard Stalls
Est. Local Speedup: 34.84%
On average, each warp of this workload spends 3.5 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 34.8% of the total average of 10.0 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

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]	143,622,144	Avg. Executed Instructions Per Scheduler [inst]	242,604.97
Issued Instructions [inst]	143,622,144	Avg. Issued Instructions Per Scheduler [inst]	242,604.97

FP32 Non-Fused Instructions
Est. Speedup: 8.26%
This kernel executes 7340032 fused and 14843904 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 33% (relative to its current performance).

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	32,768	Function Cache Configuration	CachePreferNone
Cluster Size	0	Preferred Cluster Size	0
Registers Per Thread [register/thread]	32	Cluster Scheduling Policy	PolicySpread
Static Shared Memory Per Block [byte/block]	128	Block Size	32
Dynamic Shared Memory Per Block [byte/block]	0	Threads [thread]	1,048,576
Driver Shared Memory Per Block [Kbyte/block]	1.02	Waves Per SM	6.92
Shared Memory Configuration Size [Kbyte]	65.54	Uses Green Context	0
Stack Size	1,024	# SMs [SM]	148
# TPCs	74	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 [%]	50	Block Limit Registers [block]	64
Theoretical Active Warps per SM [warp]	32	Block Limit Shared Mem [block]	56
Achieved Occupancy [%]	46.78	Block Limit Warps [block]	64
Achieved Active Warps Per SM [warp]	29.94	Block Limit SM [block]	32
Cluster Occupancy [%]	0	Block Limit Barriers [block]	32
Max Active Clusters [cluster]	0	Max Cluster Size [block]	8
Overall GPU Occupancy [%]	0		

Theoretical Occupancy
Est. Local Speedup: 50.00%
The 8.00 theoretical warps per scheduler this kernel can issue according to its occupancy are below the hardware maximum of 16. This kernel's theoretical occupancy (50.0%) is limited by the number of blocks that can fit on the SM.

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]	324,295.19	Average L1 Active Cycles [cycle]	324,295.19
Average L2 Active Cycles [cycle]	327,702.77	Average SMSP Active Cycles [cycle]	321,947.52
Average DRAM Active Cycles [cycle]	66,499.62	Total SM Elapsed Cycles [cycle]	49,266,172
Total L1 Elapsed Cycles [cycle]	49,266,172	Total L2 Elapsed Cycles [cycle]	63,045,484
Total SMSP Elapsed Cycles [cycle]	197,064,688	Total DRAM Elapsed Cycles [cycle]	75,869,440

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]	7,962,624	Branch Efficiency [%]	100
Branch Instructions Ratio [%]	0.06	Avg. Divergent Branches [branches]	0

Uncoalesced Global Accesses
Est. Speedup: 83.01%
This kernel has uncoalesced global accesses resulting in a total of 102760448 excessive sectors (87% of the total 118390784 sectors). Check the L2 Theoretical Sectors Global Excessive table for the primary source locations. The CUDA Programming Guide has additional information on reducing uncoalesced device memory accesses.

Key Performance Indicators

L2 Theoretical Sectors Global Excessive

Location	Value	Value (%)
0x7ffcc55ec8a0 in gemv_nvfp4_kernel	3,211,264	3
0x7ffcc55ec890 in gemv_nvfp4_kernel	3,211,264	3
0x7ffcc55ec830 in gemv_nvfp4_kernel	3,211,264	3
0x7ffcc55ec810 in gemv_nvfp4_kernel	3,211,264	3
0x7ffcc55ec750 in gemv_nvfp4_kernel	3,211,264	3

Warp Stall Sampling (All Samples)

Location	Value	Value (%)
0x7ffcc55eb880 in gemv_nvfp4_kernel	1,459	12
0x7ffcc55eb8c0 in gemv_nvfp4_kernel	1,081	9
0x7ffcc55ec710 in gemv_nvfp4_kernel	182	2
0x7ffcc55ec2a0 in gemv_nvfp4_kernel	171	1
0x7ffcc55ec610 in gemv_nvfp4_kernel	164	1

Most Instructions Executed

Location	Value	Value (%)
0x7ffcc55eca00 in gemv_nvfp4_kernel	458,752	0
0x7ffcc55ec9f0 in gemv_nvfp4_kernel	458,752	0
0x7ffcc55ec9e0 in gemv_nvfp4_kernel	458,752	0
0x7ffcc55ec9d0 in gemv_nvfp4_kernel	458,752	0
0x7ffcc55ec9c0 in gemv_nvfp4_kernel	458,752	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.