Page: Details | Launch: 0 - 516 - stream_version_kernel | Add Baseline | Apply Rules | Save as PDF

	Launch	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	516 - stream_version_kernel (8, 1024, 1)x(1024, 1, 1)	899,55 usecond	1.229.299	28	NVIDIA GeForce GTX 1650	1,37 cycle/nsecond	7.5	[12351] res

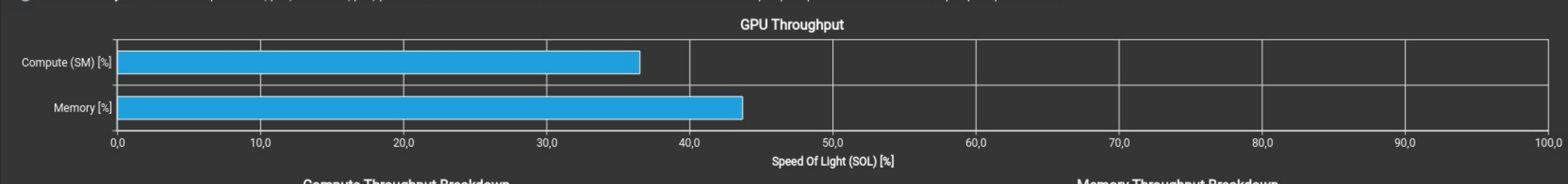
GPU Speed Of Light Throughput

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 [%]	36,51	Duration [usecond]	899,55
Memory Throughput [%]	43,69	Elapsed Cycles [cycle]	1.229.299
L1/TEX Cache Throughput [%]	41,35	SM Active Cycles [cycle]	1.137.670,79
L2 Cache Throughput [%]	43,69	SM Frequency [cycle/nsecond]	1,37
DRAM Throughput [%]	38,60	DRAM Frequency [cycle/nsecond]	5,99

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 6% of this device's fp32 peak performance and 0% of its fp64 peak performance.

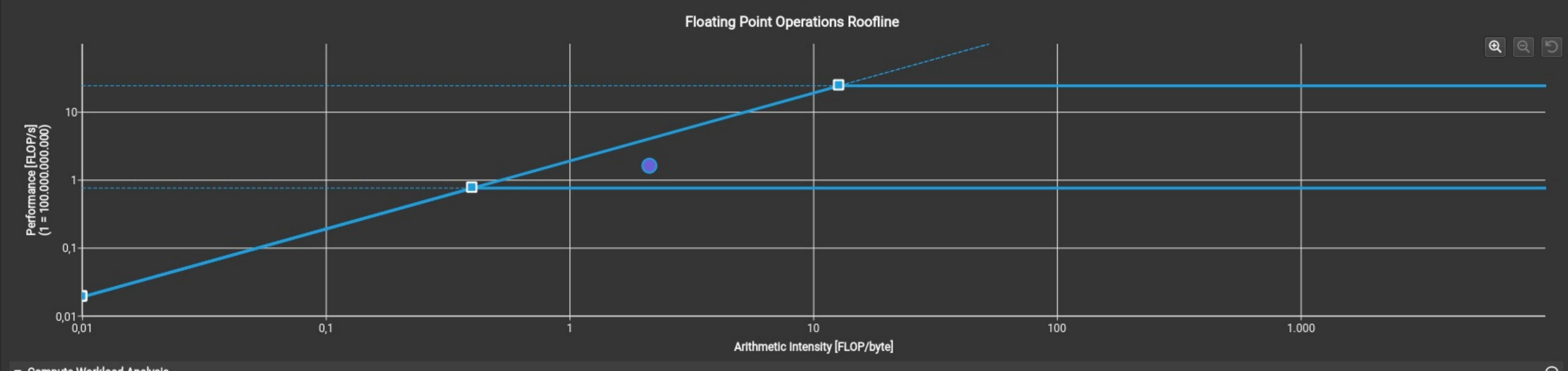


Compute Throughput Breakdown

SM: Inst Executed Pipe Lsu [%]	36,51
SM: Issue Active [%]	29,68
SM: Inst Executed [%]	29,68
SM: Pipe Alu Cycles Active [%]	19,03
SM: Pipe Fma Cycles Active [%]	18,26
SM: Mio Inst Issued [%]	14,20
SM: Mio2rf Writeback Active [%]	10,34
SM: Inst Executed Pipe Xu [%]	9,13
SM: Inst Executed Pipe Cbu Pred On Any [%]	6,20
SM: Inst Executed Pipe Adu [%]	6,10
SM: Inst Executed Pipe Uniform [%]	4,57
SM: Mio Pq Read Cycles Active [%]	1,52
SM: Mio Pq Write Cycles Active [%]	1,52
SM: Pipe Tensor Cycles Active [%]	0
SM: Pipe Shared Cycles Active [%]	0
SM: Pipe Fp64 Cycles Active [%]	0
IDC: Request Cycles Active [%]	0
SM: Inst Executed Pipe Tex [%]	0
SM: Inst Executed Pipe Ipa [%]	0
SM: Inst Executed Pipe Fp16 [%]	0

Memory Throughput Breakdown

L2: T Sectors [%]	43,69
DRAM: Cycles Active [%]	38,60
L1: Lsuin Requests [%]	36,51
L2: Lts2xbar Cycles Active [%]	32,67
L2: Xbar2lts Cycles Active [%]	22,91
DRAM: Dram Sectors [%]	21,97
L1: Data Pipe Lsu Wavefronts [%]	20,67
L1: Lsu Writeback Active [%]	18,22
L1: M Xbar2l1tex Read Sectors [%]	17,99
L2: D Sectors [%]	16,43
L2: T Tag Requests [%]	16,08
L2: D Sectors Fill Device [%]	11,09
L1: M L1tex2xbar Req Cycles Active [%]	10,88
L1: Data Bank Reads [%]	7,60
L1: Data Bank Writes [%]	3,01
L1: F Wavefronts [%]	0,00
L1: Texin Sm2tex Req Cycles Active [%]	0,00
L1: Data Pipe Tex Wavefronts [%]	0
L1: Tex Writeback Active [%]	0
L2: D Atomic Input Cycles Active [%]	0
L2: D Sectors Fill Sysmem [%]	0

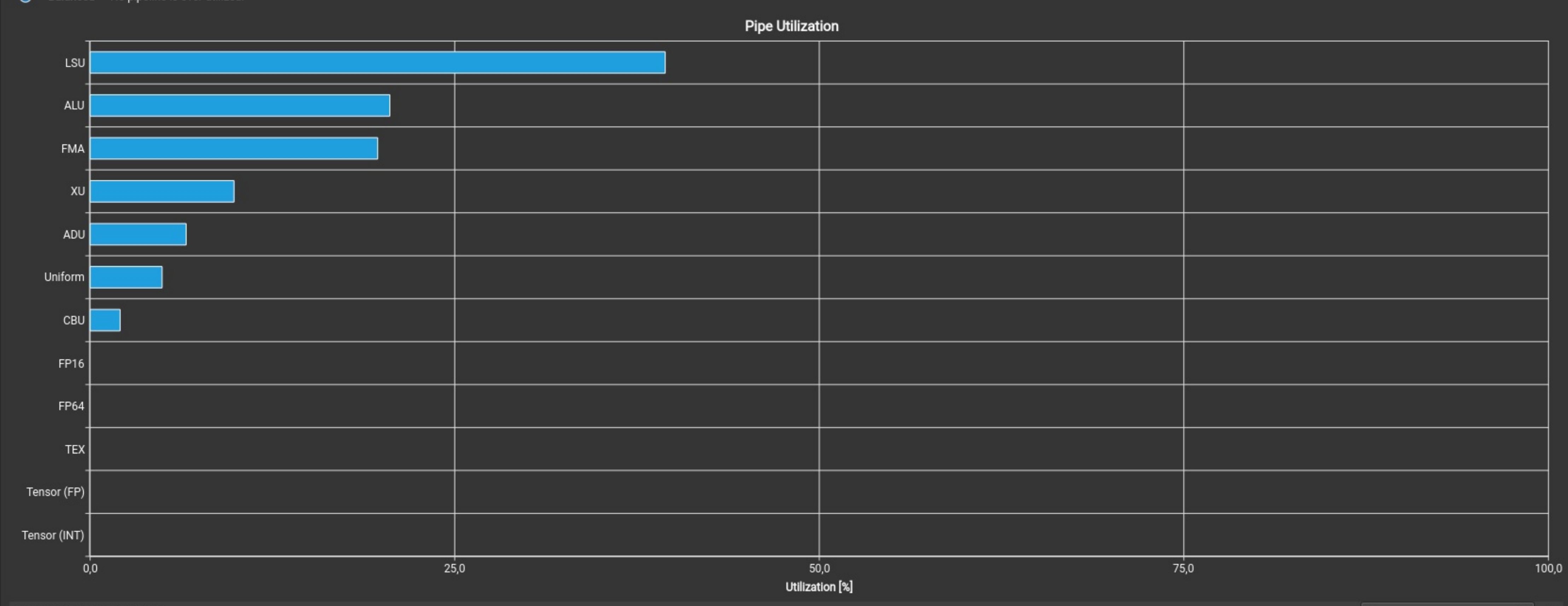


Compute Workload Analysis

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]	1,19	SM Busy [%]	32,06
Executed Ipc Active [inst/cycle]	1,28	Issue Slots Busy [%]	32,06
Issued Ipc Active [inst/cycle]	1,28		

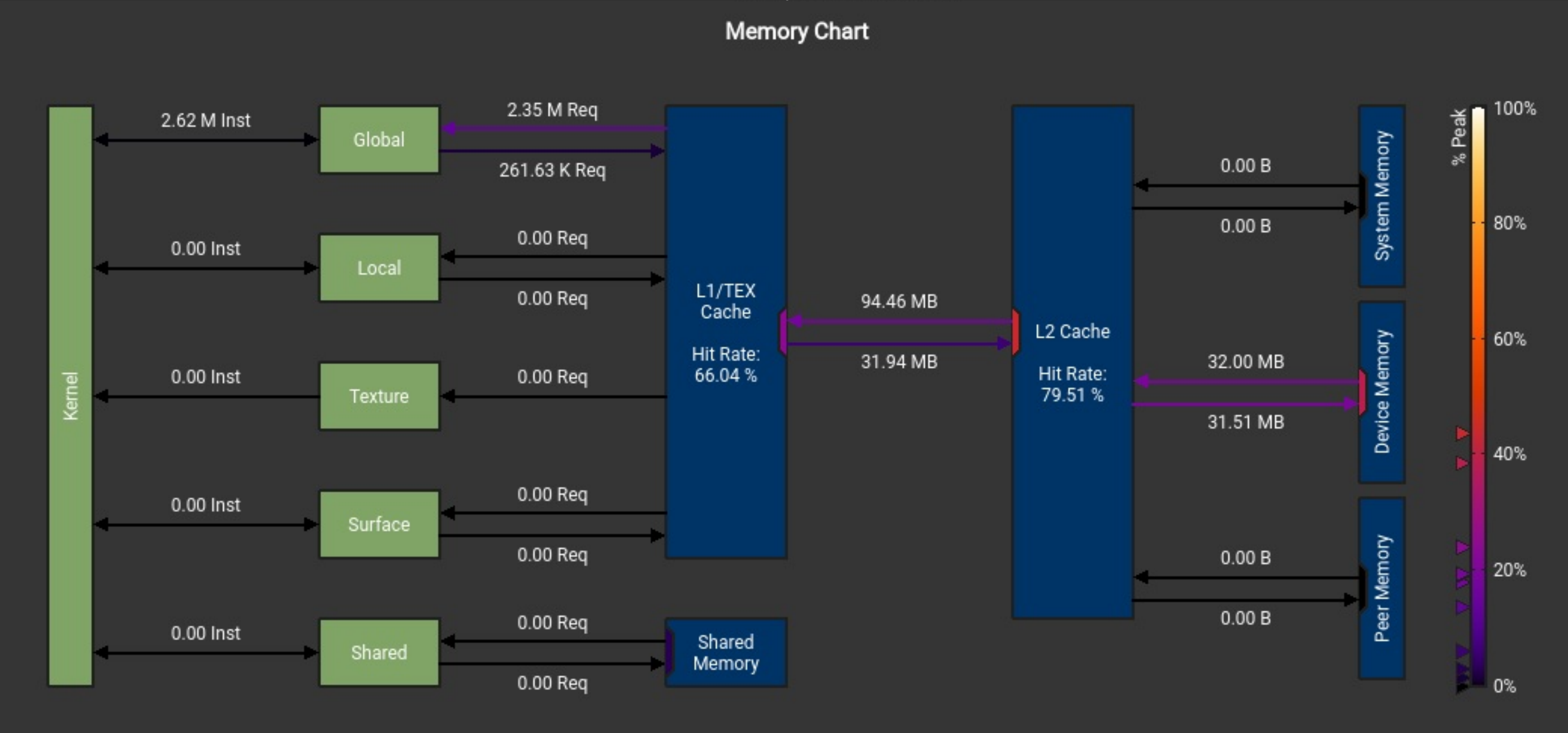
Balanced No pipeline is over-utilized.



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. Deprecated UI elements for backwards compatibility.

Memory Throughput [Gbyte/second]	74,04	Mem Busy [%]	43,69
L1/TEX Hit Rate [%]	66,04	Max Bandwidth [%]	38,60
L2 Hit Rate [%]	79,51	Mem Pipes Busy [%]	36,51



Shared Memory

	Instructions	Requests	Wavefronts	% Peak	Bank Conflicts
Shared Load	0	0	0	0	0
Shared Load Matrix	0	0	0	0	0
Shared Store	0	0	0	0	0
Shared Atomic	0	0	0	0	0
Other	-	-	524.288	3,05	0
Total	0	0	524.288	3,05	0

L1/TEX Cache

	Instructions	Requests	Wavefronts	% Peak	Sectors	Sectors/Req	Hit Rate	Bytes	Sector Misses to L2	% Peak to L2	Returns to SM	% Peak to SM
Local Load	0	0	0	0	0	0	0	0	3.100.644	18,02	2.608.911	15,16
Global Load	2.354.688	2.354.688	2.458.932	14,29	10.896.403	4,63	72,39	348.684.896				
Surface Load	0	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	261.632	261.632	261.632	1,52	1.046.528	4	0	33.488.896	1.046.528	6,08	-	-
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	-	-
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	see above
Global Atomic CAS	0											
Surface Atomic ALU	0	0	0	0	0	0	0	0	0	0	see above	see above
Surface Atomic CAS	0											
Loads	2.354.688	2.354.688	2.458.932	14,29	10.896.403	4,63	72,39	348.684.896	3.100.644	18,02	2.608.911	15,16
Stores	261.632	261.632	261.632	1,52	1.046.528	4	0	33.488.896	1.046.528	6,08	-	-
Total	2.616.320	2.616.320	2.720.564	15,81	11.942.931	4,56	66,05	382.173.792	4.147.172	24,11	2.608.911	15,16

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	1.263.370	3.094.212	2,45	32,67	68,09	99.014.784	110.071.217.672,80	1.048.576	0	0
L1/TEX Store	261.632	1.046.528	4	11,05	100	33.488.896	37.228.415.922,59	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	1.525.107	4.137.366	2,71	43,69	75,84	132.395.712	147.179.609.405,57	1.048.576	0	0
GPU Total	1.525.145	4.137.664	2,71	43,69	75,83	132.405.248	147.190.210.237,99	1.048.585	0	0

Device Memory

	Sectors	% Peak	Bytes	Throughput
Load	1.048.691	19,45	33.558.112	37.305.360.890,75
Store	1.032.522	19,15	33.040.704	36.730.176.799,12
Total	2.081.213	38,60	66.598.816	74.035.537.689,87

L1/TEX Cache (Deprecated)

	Instructions	L1/TEX Requests	% Peak	Hit Rate	L2 Requests	% Peak	L2 Returns	% Peak	L1/TEX Returns	% Peak
Global Load Cached	2.354.688	2.354.688	13,69	72,39	-	-	3.100.644	18,02	2.608.911	15,16
Global Load Uncached	-	-	-	-	-	-				
Local Load Cached	0	0	0	0	-	-				
Local Load Uncached	-	-	-	-	-	-				
Surface Load	0	0	0	0	-	-	0	0	0	0
Texture Load	0	0	0	0	-	-	0	0		
Global Store	261.632	261.632	1,52	0	1.046.528	6,08	-	-	-	-
Local Store	0	0	0	0			-	-	-	-
Surface Store	0	0	0	0	0	0	-	-	-	-
Global Reduction	0	0	0	0	0	0	-	-	-	-
Surface Reduction	0	0	0	0	0	0	-	-	-	-
Global Atomic	0	0	0	0	0	0	0	0	see above	see above
Global Atomic Cas	0									
Surface Atomic	0	0	0	0	0	0	0	0	see above	see above
Surface Atomic Cas	0									
Loads	2.354.688	2.354.688	13,69	72,39	-	-	3.100.644	18,02	2.608.911	15,16
Stores	261.632	261.632	1,52	0	1.046.528	6,08	-	-	-	-
Total	2.616.320	2.616.320	15,21	66,05	1.046.528	6,08	3.100.644	18,02	2.608.911	15,16

L2 Cache (Deprecated)

	L2 Requests	% Peak	L2 Returns	% Peak	Total Bytes	Total Throughput
Global Load Cached	-	-	3.100.644	18,02	99.220.608	110.300.024.901,28
Global Load Uncached	-	-			-	-
Local Load Cached	-	-			-	-
Local Load Uncached	-	-			-	-
Surface Load	-	-	0	0	0	0
Texture Load	-	-	0	0	0	0
Global Store	1.046.528	6,08	-	-	33.488.896	37.228.415.922,59
Local Store			-	-		
Surface Store	0	0	-	-	0	0
Global Reduction	0	0	-	-	0	0
Surface Reduction	0	0	-	-	0	0
Global Atomic	0	0	0	0	0	0
Global Atomic Cas						
Surface Atomic	0	0	0	0	0	0
Surface Atomic Cas						
Loads	-	-	3.100.644	18,02	99.220.608	110.300.024.901,28
Stores	1.046.528	6,08	-	-	33.488.896	37.228.415.922,59
Total	1.046.528	6,08	3.100.644	18,02	132.709.504	147.528.440.823,88

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]	7,12	No Eligible [%]	67,35
Eligible Warps Per Scheduler [warp]	0,95	One or More Eligible [%]	32,65
Issued Warp Per Scheduler	0,33		

Issue Slot Utilization 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 7.12 active warps per scheduler, but only an average of 0.95 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.

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]	21,82	Avg. Active Threads Per Warp	32,00
Warp Cycles Per Executed Instruction [cycle]	21,82	Avg. Not Predicated Off Threads Per Warp	29,95

long_scoreboard On average, each warp of this kernel spends 8.8 cycles being stalled waiting for a scoreboard dependency on a L1TEX (local, global, surface, texture) operation. This represents about 40.3% of the total average of 21.8 cycles between issuing two instructions. 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 or by changing the cache configuration, and consider moving frequently used data to shared memory.

Warp Stall Check the Source Counters section for the top stall locations in your source based on sampling data.

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]	20.423.168	Avg. Executed Instructions Per Scheduler [inst]	364.699,43
Issued Instructions [inst]	20.426.136	Avg. Issued Instructions Per Scheduler [inst]	364.752,43

NVLink

High-level summary of NVLink utilization. It shows the total received and transmitted (sent) memory, as well as the overall link peak utilization. NVLink Topology. Detailed tables with properties for each NVLink.

Physical Links	0	Logical Links	0

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	Registers Per Thread [register/thread]	28
Block Size	1.024	Static Shared Memory Per Block [byte/block]	0
Threads [thread]	8.388.608	Dynamic Shared Memory Per Block [byte/block]	0
Waves Per SM	585,14	Driver Shared Memory Per Block [byte/block]	0
Function Cache Configuration	cudaFuncCachePreferNone	Shared Memory Configuration Size [Kbyte]	32,77

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]	2
Theoretical Active Warps per SM [warp]	32	Block Limit Shared Mem [block]	16
Achieved Occupancy [%]	88,56	Block Limit Warps [block]	1
Achieved Active Warps Per SM [warp]	28,34	Block Limit SM [block]	16

Occupancy Limiters This kernel's theoretical occupancy is not impacted by any block limit. The difference between calculated theoretical (100.0%) and measured achieved occupancy (88.6%) can be the result of warp scheduling overheads or workload imbalances during the kernel execution. Load imbalances can occur between warps within a block as well as across blocks of the same kernel.

Source Counters

Source metrics, including branch efficiency and sampled warp stall reasons. Sampling Data 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]	1.048.064	Branch Efficiency [%]	100
Branch Instructions Ratio [%]	0,05	Avg. Divergent Branches	0

Uncoalesced Global Accesses Uncoalesced global access, expected 1046528 sectors, got 1307138 (1.25x) at PC 0x7f9856f9b350

Uncoalesced Global Accesses Uncoalesced global access, expected 1046528 sectors, got 1307138 (1.25x) at PC 0x7f9856f9b370

Uncoalesced Global Accesses Uncoalesced global access, expected 1046528 sectors, got 1307138 (1.25x) at PC 0x7f9856f9b3a0

Uncoalesced Global Accesses Uncoalesced global access, expected 1046528 sectors, got 1307138 (1.25x) at PC 0x7f9856f9b3d0

Uncoalesced Global Accesses Uncoalesced global access, expected 1046528 sectors, got 1307138 (1.25x) at PC 0x7f9856f9b3e0

Uncoalesced Global Accesses Uncoalesced global access, expected 1046528 sectors, got 1307138 (1.25x) at PC 0x7f9856f9b400