Page: Details | Launch: 0 - 508 - constant_shared_mem_ker | Add Baseline | Apply Bases | Save as PDF

	Launch	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	508 - constant_shared_mem_kernel (256, 256, 1)x(32, 32, 1)	7.71 msecond	10.572.917	26	NVIDIA GeForce GTX 1650	1,37 cycle/nsecond	7.5	7125271 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 [%]	62,33	Duration [msecond]	7,71
Memory Throughput [%]	62,33	Elapsed Cycles [cycle]	10.572.917
L1/TEX Cache Throughput [%]	66,91	SM Active Cycles [cycle]	9.848.611,43
L2 Cache Throughput [%]	26,41	SM Frequency [cycle/nsecond]	1,37
DRAM Throughput [%]	37,43	DRAM Frequency [cycle/nsecond]	5,99

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 report sections.

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.

GPU Throughput

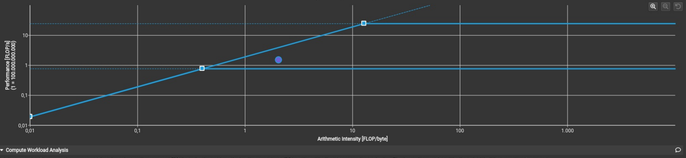
Compute Throughput Breakdown

SM: Inst Executed Pipe Lsu [%]	62,33
SM: Issue Active [%]	25,50
SM: Inst Executed [%]	25,50
SM: Mio Inst Issued [%]	23,61
SM: Pipe Alu Cycles Active [%]	17,00
SM: Pipe Fma Cycles Active [%]	14,87
SM: Mio2rf Writeback Active [%]	11,37
SM: Inst Executed Pipe Adu [%]	8,50
SM: Mio Pq Write Cycles Active [%]	5,74
SM: Mio Pq Read Cycles Active [%]	5,74
SM: Inst Executed Pipe Cbu Pred On Any [%]	1,49
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 Xu [%]	0
SM: Inst Executed Pipe Uniform [%]	0
SM: Inst Executed Pipe Tex [%]	0
SM: Inst Executed Pipe Ipa [%]	0
SM: Inst Executed Pipe Fp16 [%]	0

Memory Throughput Breakdown

L1: Lsuin Requests [%]	62,33
DRAM: Cycles Active [%]	37,43
L1: Data Pipe Lsu Wavefronts [%]	31,39
L2: T Sectors [%]	26,41
DRAM: Dram Sectors [%]	21,26
L1: Lsu Writeback Active [%]	19,91
L2: Xbar2to Cycles Active [%]	17,90
L2: Lts2xbar Cycles Active [%]	16,10
L2: D Sectors [%]	11,92
L2: D Sectors Fill Device [%]	10,96
L2: T Tag Requests [%]	10,46
L1: M1:Tex2ldst Req Cycles Active [%]	9,79
L1: M Xbar2l1tex Read Sectors [%]	8,84
L1: Data Bank Reads [%]	7,88
L1: Data Bank Writes [%]	2,61
L1: Tex0:Sm2tex Req Cycles Active [%]	0,04
L1: F Wavefronts [%]	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

Floating Point Operations Roofline



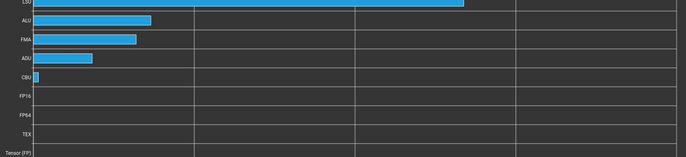
Compute Workload Analysis

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]	1,02	SM Busy [%]	27,37
Executed Ipc Active [inst/cycle]	1,09	Issue Slots Busy [%]	27,37
Issued Ipc Active [inst/cycle]	1,09		

High Utilization LSU is the highest-utilized pipeline (66.9%). It executes load/store memory operations. The pipeline is well-utilized and might become a bottleneck if more work is added.

Pipe Utilization

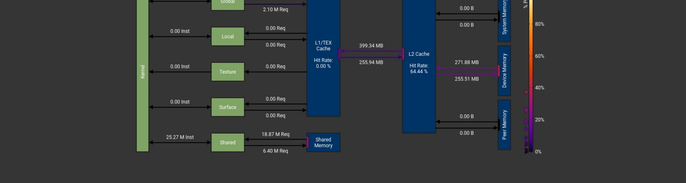


Memory Workload Analysis

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]	31,77	Mem Busy [%]	31,30
L1/TEX Hit Rate [%]	0,00	Max Bandwidth [%]	62,33
L2 Hit Rate [%]	64,44	Mem Pipes Busy [%]	62,33

Memory Chart



Shared Memory

	Instructions	Requests	Wavefronts	% Peak	Bank Conflicts
Shared Load	18.869.760	18.869.760	18.869.760	12,75	0
Shared Load Matrix	0	0	0	0	0
Shared Store	6.404.100	6.404.100	6.404.100	4,33	0
Shared Atomic	0	0	0	0	0
Other	-	-	12.418.040	8,39	0
Total	25.273.860	25.273.860	37.691.900	25,46	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	13.085.638	8,84	6.404.100	4,33
Global Load	6.404.100	6.404.100	6.404.100	4,33	13.085.700	2,04	0	418.742.400	-	-	-	-
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	2.096.640	2.096.640	2.096.640	1,42	8.386.560	4	0	268.369.920	-	-	-	-
Local Store	0	0	0	0	0	0	0	0	8.386.560	5,67	-	-
Surface Store	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	6.404.100	6.404.100	6.404.100	4,33	13.085.700	2,04	0	418.742.400	13.085.638	8,84	6.404.100	4,33
Stores	2.096.640	2.096.640	2.096.640	1,42	8.386.560	4	0	268.369.920	8.386.560	5,67	-	-
Total	8.500.740	8.500.740	8.500.740	5,74	21.472.260	2,53	0	687.112.320	21.472.198	14,51	6.404.100	4,33

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	6.404.100	13.085.700	2,04	16,09	38,02	418.742.400	54.345.090.514,93	8.908.800	0	0
L1/TEX Store	2.096.640	8.386.560	4	10,32	100	268.369.920	34.829.498.025,24	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	8.500.740	21.472.260	2,53	26,41	60,83	687.112.320	89.174.588.540,17	8.908.800	0	0
GPU Total	8.501.272	21.472.751	2,53	26,41	60,83	687.128.032	89.176.627.669,87	8.909.871	0	0

Device Memory

	Sectors	% Peak	Bytes	Throughput
Load	8.908.919	19,30	285.085.408	36.998.862.074,26
Store	8.372.620	18,14	267.923.840	34.771.405.015,18
Total	17.281.539	37,43	553.009.248	71.770.267.089,44

L1/TEX Cache (Deprecated)

	Instructions	L1/TEX Requests	% Peak	Hit Rate	L2 Requests	% Peak	L2 Returns	% Peak	L1/TEX Returns	% Peak
Global Load Cached	6.404.100	6.404.100	4,33	0	-	-	-	-	-	-
Global Load Uncached	-	-	-	-	-	-	13.085.638	8,84	6.404.100	4,33
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	2.096.640	2.096.640	1,42	0	8.386.560	5,67	-	-	-	-
Local Store	0	0	0	0	-	-	-	-	-	-
Surface Store	0	0	0	0	0	0	-	-	-	-
Surface 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	6.404.100	6.404.100	4,33	0	-	-	13.085.638	8,84	6.404.100	4,33
Stores	2.096.640	2.096.640	1,42	0	8.386.560	5,67	-	-	-	-
Total	8.500.740	8.500.740	5,74	0	8.386.560	5,67	13.085.638	8,84	6.404.100	4,33

L2 Cache (Deprecated)

	L2 Requests	% Peak	L2 Returns	% Peak	Total Bytes	Total Throughput
Global Load Cached	-	-	-	-	418.740.416	54.344.833.028,09
Global Load Uncached	-	-	13.085.638	8,84	-	-
Local Load Cached	-	-	-	-	-	-
Local Load Uncached	-	-	-	-	-	-
Surface Load	0	0	0	0	0	0
Texture Load	0	0	0	0	0	0
Global Store	8.386.560	5,67	-	-	268.369.920	34.829.498.025,24
Local Store	-	-	-	-	-	-
Surface Store	0	0	-	-	0	0
Surface 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	-	-	13.085.638	8,84	418.740.416	54.344.833.028,09
Stores	8.386.560	5,67	-	-	268.369.920	34.829.498.025,24
Total	8.386.560	5,67	13.085.638	8,84	687.110.336	89.174.331.053,33

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]	7,76	No Eligible [%]	72,31
Eligible Warps Per Scheduler [warp]	1,05	One or More Eligible [%]	27,69
Issued Warp Per Scheduler	0,28		

Issue Slot Utilization Every scheduler is capable of issuing one instruction per cycle, but for this kernel each scheduler only issues an instruction every 3.6 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 7.76 active warps per scheduler, but only an average of 1.05 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 warps 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]	28,01	Avg. Active Threads Per Warp	31,99
Warp Cycles Per Executed Instruction [cycle]	28,01	Avg. Not Predicated Off Threads Per Warp	23,80

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 kernel achieves an average of 32.0 threads being active per cycle. This is further reduced to 23.8 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. In addition, ensure your kernel makes use of Independent Thread Scheduling, which allows a warp to reconverge after a data-dependent conditional block by explicitly calling __syncwarp().

Instruction Statistics

Statistics of the executed low-level assembly instructions (SASS). The instruction mix provides insight into the types and frequency of 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]	150.994.272	Avg. Executed Instructions Per Scheduler [inst]	2 695,826,29
Issued Instructions [inst]	150.969.079	Avg. Issued Instructions Per Scheduler [inst]	2 695,876,41

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	65.536	Registers Per Thread [register/thread]	26
Block Size	1.024	Static Shared Memory Per Block [Kbyte/block]	4,62
Threads [thread]	67.108.864	Dynamic Shared Memory Per Block [Byte/block]	0
Waves Per SM	4.681,14	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 indicate highly imbalanced workloads.

Theoretical Occupancy [%]	100	Block Limit Registers [block]	2
Theoretical Active Warps per SM [warp]	32	Block Limit Shared Mem [block]	13
Achieved Occupancy [%]	96,52	Block Limit Warps [block]	1
Achieved Active Warps Per SM [warp]	30,89	Block Limit SM [block]	16

Occupancy Limiters This kernel's theoretical occupancy is not impacted by any block limit.

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]	4.193.792	Branch Efficiency [%]	0
Branch Instructions Ratio [%]	0,03	Avg. Divergent Branches	0