Page: Details | Process: All | Launch: 0 - 113 - cuColorToGray | Add Baseline | Apply Baules | Save as PDF

Current 113 - cuColorToGray (3200, 2464, 1) Time: 628.45 usecond Cycles: 859,988 Regs: 11 GPU: GeForce GTX 1050 SM Frequency: 1.35 Ghz CC: 6.1 Process: [29236] ColorToGrayImage.exe

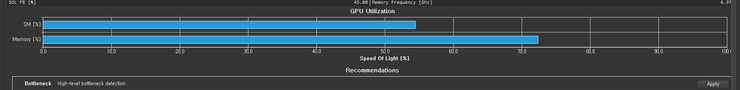
GPU Speed Of Light

All

High-level overview of the utilization for compute and memory resources of the GPU. For each unit, the Speed Of Light (SOL) reports the achieved percentage of utilization with respect to the theoretical maximum.

SOL SM [%]	56.44	Duration [usecond]	628.45
SOL Memory [%]	72.37	Elapsed Cycles [cycle]	859,988
SOL TEX [%]	47.30	SM Active Cycles [cycle]	852,499.24
SOL L2 [%]	72.37	SM Frequency [Ghz]	1.35
SOL FB [%]	45.80	Memory Frequency [Ghz]	6.91

GPU Utilization



Speed Of Light [%]

Recommendations

Bottleneck High-level bottleneck detection — Apply

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]	2.24	SM Busy [%]	56.44
Executed Ipc Active [inst/cycle]	2.25	Issue Slots Busy [%]	37.46
Issued Ipc Active [inst/cycle]	2.25		

Recommendations

Slow Pipe Limiter Slow pipe limiting compute utilization — Apply

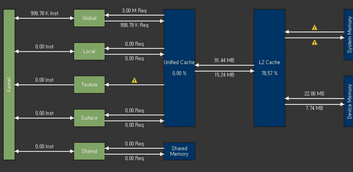
Memory Workload Analysis

All

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/second]	51.86	Mem Busy [%]	72.37
L1 Hit Rate [%]	0	Mem Bandwidth [%]	91.41
L2 Hit Rate [%]	78.57	Mem Pipes Busy [%]	23.82

Memory Chart



Shared Memory

	Instructions	Requests	% Peak	Bank Conflicts
Shared Load	0	0	0	0
Shared Store	0	0	0	0
Shared Atomic	0			
Total	0	0	0	0

First-Level (Unified) Cache

	Instructions	SM<>TEX Requests	% Peak	Hit Rate	TEX<>L2 Requests	% Peak	L2<>TEX Returns	% Peak	TEX<>SM Returns	% Peak
Global Load Cached	749,088	-	-	-	-	-	0	0		
Global Load Uncached	-	2,996,352	70.65	-	-	-	2,996,352	35.14		
Local Load Cached	0	0	0	0	-	-	0	0	2,996,352	35.14
Local Load Uncached	0	0	0	-	-	-	0	0		
Surface Load	0	0	0	-	-	-	0	0		
Texture Load	0	-	-	-	-	-	-	-		
Global Store	249,696	998,784	23.55	-	499,392	5.86	-	-	-	-
Local Store	0	0	0	-	0	0	-	-	-	-
Surface Store	0	0	0	-	0	0	-	-	-	-
Global Reduction	0	0	0	-	0	0	-	-	-	-
Surface Reduction	0	0	0	-	0	0	-	-	-	-
Global Atomic	0	0	0	-	0	0	0	0	0	0
Global Atomic Cas	0	0	0	-	0	0	0	0	0	0
Surface Atomic	0	0	0	-	0	0	0	0	0	0
Surface Atomic Cas	0	0	0	-	0	0	0	0	0	0
Loads	749,088	2,996,352	70.65	0	-	-	2,996,352	35.14	2,996,352	35.14
Stores	249,696	998,784	23.55	-	499,392	5.86	-	-	-	-
Total	998,784	3,995,136	94.20	0	499,392	5.86	2,996,352	35.14	2,996,352	35.14

Second-Level (L2) Cache

	TEX<>L2 Requests	% Peak	L2<>TEX Returns	% Peak	Total Bytes	Total Throughput
Global Load Cached	-	-	0	0	0	0
Global Load Uncached	-	-	2,996,352	45.70	95,883,264	152,571,515,661.30
Local Load Cached	-	-	0	0	0	0
Local Load Uncached	-	-	0	0	0	0
Surface Load	-	-	0	0	0	0
Texture Load	-	-	-	-	-	-
Global Store	499,392	15.23	-	-	15,980,544	25,428,585,976.88
Local Store	0	0	-	-	0	0
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	0	0	0	0	0	0
Surface Atomic	0	0	0	0	0	0
Surface Atomic Cas	0	0	0	0	0	0
Loads	0	0	2,996,352	45.70	95,883,264	152,571,515,661.30
Stores	499,392	15.23	-	-	15,980,544	25,428,585,976.88
Total	499,392	15.23	2,996,352	45.70	111,863,808	178,000,101,638.18

Device Memory (FB)

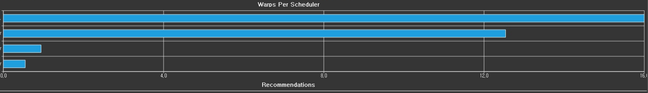
	L2<>FB Sectors	% Peak	Bytes	Throughput
Load	748,162	34.22	23,973,184	38,148,648,977.95
Store	253,635	11.58	8,116,320	12,914,863,282.24
Total	1,002,797	45.80	32,089,504	51,063,512,260.20

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 one from which to issue an instruction (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/cycle]	12.54	Instructions Per Active Issue Slot [inst/issue]	1.80
Eligible Warps Per Scheduler [warp/cycle]	0.96	No Eligible [%]	45.14
Issued Warp Per Scheduler [issue/cycle]	0.55	One or More Eligible [%]	54.75

Warps Per Scheduler



Recommendations

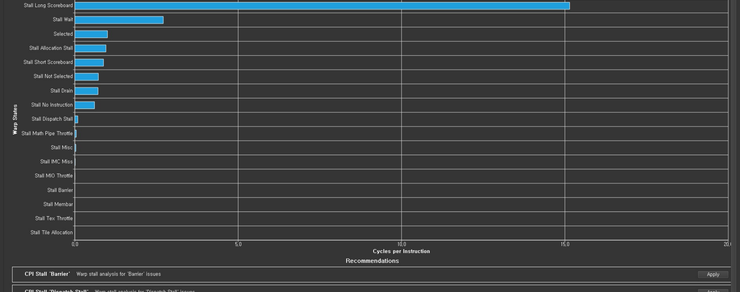
Issue Slot Utilization Scheduler instruction issue analysis — Apply

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.

Warp Cycles Per Issued Instruction [cycle/inst]	23.00	Avg. Active Threads Per Warp [thread/inst]	32
Warp Cycles Per Issue Active [cycle/issue]	22.18	Avg. Not Predicated Off Threads Per Warp [thread/inst]	31.36
Warp Cycles Per Executed Instruction [cycle/inst]	31.08		

Warp State (All Cycles)



Recommendations

CPI Stall 'Barrier' Warp stall analysis for 'Barrier' issues — Apply

CPI Stall 'Dispatch Stall' Warp stall analysis for 'Dispatch Stall' issues — Apply

CPI Stall 'Drain' Warp stall analysis for 'Drain' issues — Apply

CPI Stall 'IMC Miss' Warp stall analysis for 'Immediate constant cache (IMC)' issues — Apply

CPI Stall 'LG Throttle' Warp stall analysis for 'LG Throttle' issues — Apply

CPI Stall 'Long Scoreboard' Warp stall analysis for 'Long Scoreboard' issues — Apply

CPI Stall 'Math Pipe Throttle' Warp stall analysis for 'Math Pipe Throttle' issues — Apply

CPI Stall 'Membar' Warp stall analysis for 'Membar' issues — Apply

CPI Stall 'MIO Throttle' Warp stall analysis for 'MIO Throttle' issues — Apply

CPI Stall 'Misc' Warp stall analysis for 'Misc' issues — Apply

CPI Stall 'No Instructions' Warp stall analysis for 'No Instructions' issues — Apply

CPI Stall 'Not Selected' Warp stall analysis for 'Not Selected' issues — Apply

CPI Stall 'Short Scoreboard' Warp stall analysis for 'Short Scoreboard' issues — Apply

CPI Stall 'Sleeping' Warp stall analysis for 'Sleeping' issues — Apply

CPI Stall 'TEX Throttle' Warp stall analysis for 'TEX Throttle' issues — Apply

CPI Stall 'Wait' Warp stall analysis for 'Wait' issues — Apply

Thread Divergence Warp and thread control flow analysis — Apply

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]	9,533,440	Avg. Executed Instructions Per Scheduler [inst]	476,578.46
Issued Instructions [inst]	9,532,480	Avg. Issued Instructions Per Scheduler [inst]	476,634.26

Executed Instruction Mix

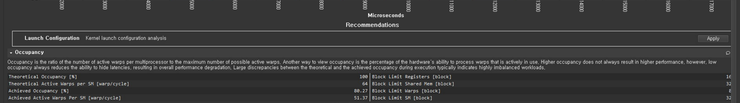
Launch Statistics

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	31,570	Registers Per Thread [register/thread]	11
Block Size	256	Static Shared Memory Per Block [byte/block]	0
Threads [thread]	8,081,920	Dynamic Shared Memory Per Block [byte/block]	0
Waves Per SM	789.25	Shared Memory Configuration Size [byte]	0

Block Durations

Warp Durations



Recommendations

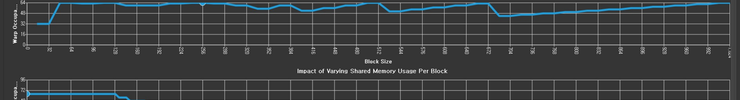
Launch Configuration Kernel launch configuration analysis — Apply

Occupancy

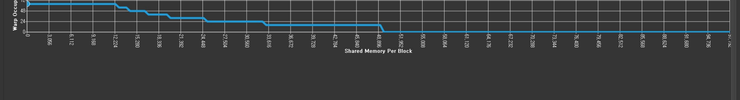
Occupancy is the ratio of the number of active warps per multiprocessor to the maximum number of possible active warps. Another way to view occupancy is the percentage of the hardware's ability to process warps that is actively in use. Higher occupancy does not always result in higher performance, however, 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]	16
Theoretical Active Warps per SM [warp/cycle]	64	Block Limit Shared Mem [block]	32
Achieved Occupancy [%]	89.27	Block Limit Warps [block]	8
Achieved Active Warps Per SM [warp/cycle]	57.13	Block Limit SM [block]	32

Impact of Varying Register Count Per Thread



Impact of Varying Block Size



Impact of Varying Shared Memory Usage Per Block