Page: Details | Result: 12 - 621 - kernelSortBins | Add Baseline | Apply Rules | Occupancy Calculator | Save as PDF

	Result	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	621 - kernelSortBins (21, 1, 1)x(256, 1, 1)	58.90 msecond	90,002,017	16	0 - NVIDIA GeForce RTX 2080	1.52 cycle/nsecond	7.5	[74512] render

GPU Speed Of Light Throughput

All

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.

Compute (SM) Throughput [%]	5.41	Duration [msecond]	58.90
Memory Throughput [%]	31.22	Elapsed Cycles [cycle]	90,002,017
L1/TEX Cache Throughput [%]	52.73	SM Active Cycles [cycle]	34,883,783.96
L2 Cache Throughput [%]	31.22	SM Frequency [cycle/nsecond]	1.52
DRAM Throughput [%]	5.17	DRAM Frequency [cycle/nsecond]	6.78

Small Grid This kernel grid is too small to fill the available resources on this device, resulting in only 0.1 full waves across all SMs. Look at Launch Statistics for more details.

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	21	Registers Per Thread [register/thread]	16
Block Size	256	Static Shared Memory Per Block [byte/block]	0
Threads [thread]	5,376	Dynamic Shared Memory Per Block [byte/block]	0
Waves Per SM	0.11	Driver Shared Memory Per Block [byte/block]	0
Function Cache Configuration	cudaFuncCachePreferNone	Shared Memory Configuration Size [Kbyte]	32.77

Small Grid The grid for this launch is configured to execute only 21 blocks, which is less than the GPU's 46 multiprocessors. This can underutilize some multiprocessors. If you do not intend to execute this kernel concurrently with other workloads, consider reducing the block size to have at least one block per multiprocessor or increase the size of the grid to fully utilize the available hardware resources. See the Hardware Model description for more details on launch configurations.

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]	32	Block Limit Shared Mem [block]	16
Achieved Occupancy [%]	23.88	Block Limit Warps [block]	4
Achieved Active Warps Per SM [warp]	7.64	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 (23.9%) 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. See the CUDA Best Practices Guide for more details on optimizing occupancy.