Page: Details | Result: 5 - 136 - prefixSumScanKernel | Add Baseline | Apply Rules | Occupancy Calculator | Save as PDF

	Result	Time	Cycles	Regs	GPU	SM Frequency	CC	Process
Current	136 - prefixSumScanKernel (...	30.30 usecond	٤٢٬٠٣٣	١٦	0 - NVIDIA GeForce RTX 3090	1.38 cycle/nsecond	8.6	[258893] task2

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.

Compute (SM) Throughput [%]	20.61	Duration [usecond]	30.30
Memory Throughput [%]	18.13	Elapsed Cycles [cycle]	42,033
L1/TEX Cache Throughput [%]	19.35	SM Active Cycles [cycle]	39,321.55
L2 Cache Throughput [%]	2.26	SM Frequency [cycle/nsecond]	1.38
DRAM Throughput [%]	0.76	DRAM Frequency [cycle/nsecond]	9.44

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.

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	24,000	Function Cache Configuration	cudaFuncCachePreferNone
Registers Per Thread [register/thread]	16	Static Shared Memory Per Block [Kbyte/block]	8.19
Block Size	4	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	96,000	Driver Shared Memory Per Block [Kbyte/block]	1.02
Waves Per SM	26.61	Shared Memory Configuration Size [Kbyte]	102.40

Block Size Threads are executed in groups of 32 threads called warps. This kernel launch is configured to execute 4 threads per block. Consequently, some threads in a warp are masked off and those hardware resources are unused. Try changing the number of threads per block to be a multiple of 32 threads. Between 128 and 256 threads per block is a good initial range for experimentation. Use smaller thread blocks rather than one large thread block per multiprocessor if latency affects performance. This is particularly beneficial to kernels that frequently call __syncthreads(). 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 [%]	22.92	Block Limit Registers [block]	128
Theoretical Active Warps per SM [warp]	11	Block Limit Shared Mem [block]	11
Achieved Occupancy [%]	18.31	Block Limit Warps [block]	48
Achieved Active Warps Per SM [warp]	8.79	Block Limit SM [block]	16

Occupancy Limiters This kernel's theoretical occupancy (22.9%) is limited by the required amount of shared memory See the CUDA Best Practices Guide for more details on optimizing occupancy.