	Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current	578 - parallel_merge_kernel	(524288, 1, 1)x(256, 1, 1)	14,82 ms	18.112.354	0 - NVIDIA GeForce RTX 3050 Ti Laptop GPU	1,22 Ghz	[7960] merge_sort.exe	

Summary | Details | Source | Context | Comments | Raw | Session

Compare | Tools | View | Export

GPU Speed Of Light Throughput

GPU Throughput Chart

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 [%]	89,31	Duration [ms]	14,82
Memory Throughput [%]	51,07	Elapsed Cycles [cycle]	18.112.354
L1/TEX Cache Throughput [%]	51,07	SM Active Cycles [cycle]	18.108.795,50
L2 Cache Throughput [%]	22,07	SM Frequency [Ghz]	1,22
DRAM Throughput [%]	37,82	DRAM Frequency [Ghz]	5,99

High Throughput The kernel is utilizing greater than 80.0% of the available compute or memory performance of the device. To further improve performance, work will likely need to be shifted from the most utilized to another unit. Start by analyzing workloads in the Compute Workload Analysis section.

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	524.288	Function Cache Configuration	CachePreferNone
Registers Per Thread [register/thread]	21	Static Shared Memory Per Block [byte/block]	0
Block Size	256	Dynamic Shared Memory Per Block [byte/block]	0
Threads [thread]	134.217.728	Driver Shared Memory Per Block [Kbyte/block]	1,02
Waves Per SM	4.369,07	Shared Memory Configuration Size [Kbyte]	8,19
Uses Green Context	0	# SMs [SM]	20

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]	10
Theoretical Active Warps per SM [warp]	48	Block Limit Shared Mem [block]	8
Achieved Occupancy [%]	89,62	Block Limit Warps [block]	6
Achieved Active Warps Per SM [warp]	43,02	Block Limit SM [block]	16

Achieved Occupancy
Est. Local Speedup: 10.38%

The difference between calculated theoretical (100.0%) and measured achieved occupancy (89.6%) 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. See the CUDA Best Practices Guide for more details on optimizing occupancy.

GPU and Memory Workload Distribution

Analysis of workload distribution in active cycles of SM, SMP, SMSP, L1 & L2 caches, and DRAM

Average SM Active Cycles [cycle]	18.108.795,50	Average L1 Active Cycles [cycle]	18.108.795,50
Average L2 Active Cycles [cycle]	17.261.959,38	Average SMSP Active Cycles [cycle]	18.109.485,69
Average DRAM Active Cycles [cycle]	33.558.032	Total SM Elapsed Cycles [cycle]	362.240.150
Total L1 Elapsed Cycles [cycle]	362.240.150	Total L2 Elapsed Cycles [cycle]	277.340.064
Total SMSP Elapsed Cycles [cycle]	1.448.960.600	Total DRAM Elapsed Cycles [cycle]	354.947.072

Missing Roofline and Memory Charts? Profile again with the *detailed* metric set to collect all necessary metrics.
Also consider collecting all available sections with the *full* metric set.