	Result	Size	Time	Cycles	GPU	SM Frequency	Process	Attributes
Current	525 - transposeKernelInt	(32, 32, 1)x(32, 32, 1)	98.05 us	99,844	0 - NVIDIA GeForce GTX 1650 with Max-Q Design	1.02 Ghz	[18271] main	

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

Compare | Tools | View | Export

This table shows all results in the report. Use the column headers to sort the results in this report. Double-click a result to see detailed metrics. Double-click on demangled names to rename it.

ID	Estimated Speedup	Function Name	Demangled Name	Duration (1.77677e+08)	Runtime Improvement (1.70831e+07)	Compute Throughput	Memory Throughput	# Registers	Grid Size
0	61.31	checkSortedRowWi...	checkSortedRowWi...	0.03	0.02	8.37	38.69	16	1024,
1	0.62	sortRowsKernelInt	sortRowsKernelInt(...	50.52	0.31	99.38	99.38	45	1024,
2	22.96	transposeKernelInt	transposeKernelInt(...	0.10	0.02	24.61	77.04	16	32,
3	61.87	checkSortedRowWi...	checkSortedRowWi...	0.03	0.02	9.13	37.23	16	1024,
4	1.18	sortRowsKernelInt	sortRowsKernelInt(...	47.45	0.56	98.82	98.82	45	1024,
5	24.52	transposeKernelInt	transposeKernelInt(...	0.10	0.02	23.96	75.48	16	32,
6	60.26	checkSortedRowWi...	checkSortedRowWi...	0.03	0.02	11.99	36.61	16	1024,
7	3.69	sortRowsKernelInt	sortRowsKernelInt(...	14.30	0.53	96.31	96.31	45	1024,
8	25.74	transposeKernelInt	transposeKernelInt(...	0.10	0.02	24.77	74.26	16	32,
9	59.13	checkSortedRowWi...	checkSortedRowWi...	0.03	0.02	12.71	37.35	16	1024,
10	2.53	sortRowsKernelInt	sortRowsKernelInt(...	13.03	0.33	97.47	97.47	45	1024,

The following performance optimization opportunities were discovered for this result. Follow the rule links to see more context on the Details page.
Note: *Speedup estimates provide upper bounds for the optimization potential of a kernel assuming its overall algorithmic structure is kept unchanged.*

Long Scoreboard Stalls
Est. Speedup: 22.96%

On average, each warp of this kernel spends 23.4 cycles being stalled waiting for a scoreboard dependency on a L1TEX (local, global, surface, texture) operation. Find the instruction producing the data being waited upon to identify the culprit. To reduce the number of cycles waiting on L1TEX data accesses verify the memory access patterns are optimal for the target architecture, attempt to increase cache hit rates by increasing data locality (coalescing), or by changing the cache configuration. Consider moving frequently used data to shared memory. This stall type represents about 59.0% of the total average of 39.6 cycles between issuing two instructions.