

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 [ms] (6.45 ms)	Runtime Improvement [ms] (4.27 ms)	Compute Throughput [%]	Memory Throughput [%]	# Reg
3	53.06	gradient_kernel	gradient_kernel(flo...	2.83	1.50	87.15	7.58	
4	63.03	edge_kernel	edge_kernel.char..	0.18	0.11	28.70	36.97	
5	48.34	thresholding kernel	thresholding kernel..	0.09	0.04	32.00	51.66	

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: 63.03%

On average, each warp of this workload spends 30.0 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 81.5% of the total average of 36.8 cycles between issuing two instructions.
- [Uncoalesced Global Accesses](#)  
Est. Speedup: 39.40%

This kernel has uncoalesced global accesses resulting in a total of 909824 excessive sectors (41% of the total 2241606 sectors). Check the L2 Theoretical Sectors Global Excessive table for the primary source locations. The [CUDA Programming Guide](#) has additional information on reducing uncoalesced device memory accesses.
- [L1TEX Global Store Access Pattern](#)  
Est. Speedup: 38.09%

The memory access pattern for global stores to L1TEX might not be optimal. On average, only 7.8 of the 32 bytes transmitted per sector are utilized by each thread. This could possibly be caused by a stride between threads. Check the [Source Counters](#) section for uncoalesced global stores.