Reduction

CPU Reduction

- Execution time increases steadily as array size grows. The relationship is roughly linear with small timing increments due to efficient memory and computation handling on the CPU.
- The CPU performs well for smaller array sizes, but as the size exceeds 1,000,000, the execution time becomes significantly slower (0.004814 seconds at 1,000,000).
- For very large arrays, CPU reduction becomes inefficient compared to GPU methods.

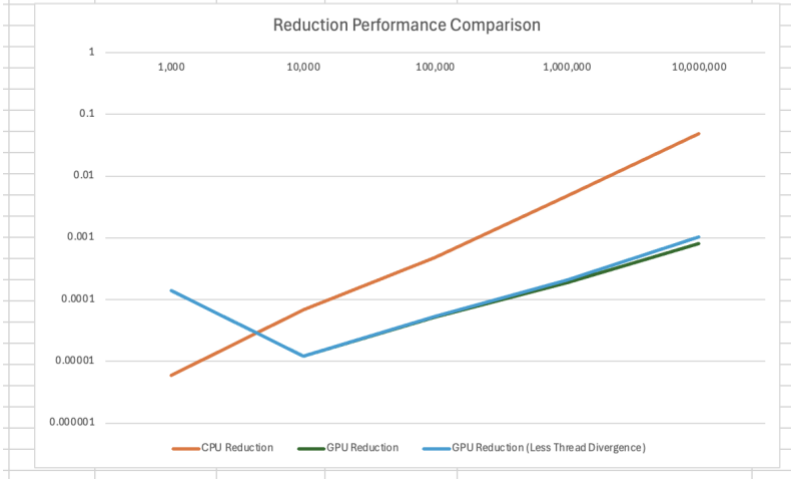
GPU Reduction

- The GPU shows minimal overhead, especially for larger array sizes. It outperforms the CPU for all array sizes, including small arrays like 10,000 elements (0.000012 seconds).
- GPU Reduction becomes practical starting from 100,000 elements compared to the CPU, where the GPU demonstrates consistent scalability with minimal execution time.

GPU Reduction with Less Thread Divergence

- The Less Thread Divergent GPU method has similar performance to the standard GPU reduction for smaller sizes but is slightly slower for larger arrays (e.g., 10,000,000 elements take 0.001038 seconds compared to 0.000817 seconds for the standard GPU).
- The Less Thread Divergent optimization does not show significant benefits for this dataset because thread divergence likely isn't a major issue for this workload. Instead, its extra logic introduces slight overhead.

Array Size	CPU Reduction	GPU Reduction	GPU Reduction (Less Thread Divergence)
1,000	0.000006	0.000141	0.000139
10,000	0.000068	0.000012	0.000012
100,000	0.000487	0.000052	0.000054
1,000,000	0.004814	0.000187	0.000207
10,000,000	0.049066	0.000817	0.001038



Histogram

CPU Histogram

- Execution time scales linearly with array size. It handles small arrays very efficiently (e.g., 0.000006 seconds for 1,000 elements) but becomes slower for larger sizes (0.043062 seconds for 10,000,000 elements).
- The CPU method is effective for smaller datasets (<1,000,000 elements) but becomes impractical for larger sizes as its time grows proportionally with size.

2. GPU Histogram (Non-Strided)

- The Non-Strided GPU Histogram is highly efficient, with performance increasing exponentially with array size. For 10,000,000 elements, it executes in just 0.004320 seconds compared to the CPU's 0.043062 seconds.
- The Non-Strided GPU method starts outperforming the CPU for arrays larger than 10,000 elements, with the gap widening as sizes grow.

GPU Histogram (Strided)

- The Strided GPU Histogram shows nearly identical performance to the Non-Strided method for all tested sizes. For larger datasets (e.g., 10,000,000 elements), the Strided method takes 0.004328 seconds, slightly slower than the Non-Strided method due to increased memory access complexity.
- The Strided optimization doesn't provide clear benefits for this dataset.

Array Size	CPU Histogram	GPU Histogram (Non-Strided)	GPU Histogram (Strided)
1,000	0.000006	0.00015	0.000139
10,000	0.000047	0.000013	0.000013
100,000	0.000448	0.000086	0.000084
1,000,000	0.004306	0.000538	0.000545
10,000,000	0.043062	0.00432	0.004328