CUDA gpuKendall Kernel Execution Time Prediction Report

This report applies the CUDA kernel performance prediction template to the kernel: gpuKendall, launched with Grid = (5,5) and Block = (32,32).

We estimate the expected per-launch execution time on five NVIDIA GPUs:
• GeForce GTX TITAN Black
• GeForce GTX TITAN X
• NVIDIA TITAN V
• GeForce RTX 2080 Ti
• GeForce RTX 4070

```
__global__ void gpuKendall(const float * a, size_t na, const float * b, size_t nb,
                           size_t sampleSize, double * results)
{
    size_t i, j, tests,
           tx = threadIdx.x, ty = threadIdx.y,
           bx = blockIdx.x,  by = blockIdx.y,
           rowa = bx * sampleSize, rowb = by * sampleSize;
    float discordant, concordant = 0.f, numer, denom;
    __shared__ float threadSums[NUMTHREADS*NUMTHREADS];

    for(i = tx; i < sampleSize; i += NUMTHREADS) {
        for(j = i+1+ty; j < sampleSize; j += NUMTHREADS) {
            tests = ((a[rowa+j] >  a[rowa+i]) && (b[rowb+j] >  b[rowb+i]))
                  + ((a[rowa+j] <  a[rowa+i]) && (b[rowb+j] <  b[rowb+i]))
                  + ((a[rowa+j] == a[rowa+i]) && (b[rowb+j] == b[rowb+i]));
            concordant = concordant + (float)tests;
        }
    }
    threadSums[tx*NUMTHREADS+ty] = concordant;
    __syncthreads();
    // 2D reduction in shared memory...
}
```

1. Workload Analysis

Launch configuration:
• Grid = (5,5) → 25 blocks
• Block = (32,32) → 1024 threads/block
• NUMTHREADS = 16 (used for loop strides and shared memory)
• na = nb = 5, sampleSize = 100

Logically, the computation is a pairwise comparison over all distinct index pairs (i,j) with $0 \le i < j <$ sampleSize for each block (bx,by).

Total pairs per block:
C(sampleSize, 2) = sampleSize × (sampleSize - 1) / 2
= 100 × 99 / 2 = 4,950 pairs

With 25 blocks (all used, since grid = (5,5) and na = nb = 5):
Total pair evaluations across the grid:
N_pairs_total = 4,950 × 25 = 123,750

2. FLOP Count (Approximate)

For each pair, the code evaluates three conjunctions of float comparisons and accumulates an integer count into a float:

tests = (condition1) + (condition2) + (condition3);
concordant += (float) tests;

We approximate this as ≈ 10 FLOP-equivalents per pair (comparisons, boolean logic, and the final float add).

Total FLOPs:
F_total ≈ 123,750 × 10 = 1,237,500 ≈ 1.24 × 10^6 FLOPs.

3. Memory Traffic

For each pair (i,j) and a given block (bx,by), the kernel reads:
• a[rowa + i], a[rowa + j]
• b[rowb + i], b[rowb + j]

Assuming these four float values are loaded once each and reused in the three conditions, we get 4 float loads per pair → 16 bytes per pair.

Total bytes per block:
B_block = 4,950 pairs × 16 bytes ≈ 79,200 bytes

Across 25 blocks:
B_total ≈ 79,200 × 25 = 1,980,000 bytes ≈ 1.98 × 10^6 bytes (~1.89 MiB).

The final write of one double per block (results[by*na + bx]) adds only 25 × 8 = 200 bytes and is negligible in comparison.

Thus, the kernel is predominantly memory- and launch-limited.

4. GPU Specifications Used

Approximate FP32 peak performance and memory bandwidth:

GPU	Peak FP32 (FLOPs/s)	Bandwidth (bytes/s)
GTX TITAN Black	5.12e12	3.36e11
GTX TITAN X	6.14e12	3.365e11
TITAN V	1.49e13	6.528e11
RTX 2080 Ti	1.345e13	6.16e11
RTX 4070	2.9e13	5.04e11

5. Time Estimates

Using:
• F_total ≈ 1.2375 × 10^6 FLOPs

• B_total ≈ 1.98 × 10^6 bytes

We compute:
t_compute = F_total / Peak_FP32
t_mem = B_total / Bandwidth
t_body = max(t_compute, t_mem)
t_total ≈ t_body + 5 µs (for kernel launch overhead).

GPU	t_compute (µs)	t_mem (µs)	t_body (µs)	t_total (µs)
GTX TITAN Black	0.24	5.89	5.89	≈ 10.89
GTX TITAN X	0.20	5.88	5.88	≈ 10.88
TITAN V	0.08	3.03	3.03	≈ 8.03
RTX 2080 Ti	0.09	3.21	3.21	≈ 8.21
RTX 4070	0.04	3.93	3.93	≈ 8.93

6. Conclusion

The gpuKendall kernel performs roughly 1.24M FLOPs and reads about 1.98MB of data per launch, spread across 25 blocks and 16×16 logical threads per block.

Because the memory and compute loads are modest, the kernel body completes in a few microseconds, and total runtime per launch is dominated by the fixed CUDA kernel launch overhead (~5 µs).

Predicted per-launch times:
• GTX TITAN Black / TITAN X: ≈ 10.9 µs
• TITAN V: ≈ 8.0 µs
• RTX 2080 Ti: ≈ 8.2 µs
• RTX 4070: ≈ 8.9 µs

These values are approximate but consistent with typical behavior of small, pairwise-comparison kernels with limited memory traffic.