CUDA gpuMeans Kernel Execution Time Prediction Report

This report applies the CUDA kernel performance prediction template to the kernel:
gpuMeans, 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 gpuMeans(const float * vectsA, size_t na,
                         const float * vectsB, size_t nb,
                         size_t dim, float * means, float * numPairs)
{
    size_t offset, stride,
           bx = blockIdx.x, by = blockIdx.y, tx = threadIdx.x;
    float a, b;

    __shared__ float
        threadSumsA[NUMTHREADS], threadSumsB[NUMTHREADS],
        count[NUMTHREADS];

    if((bx >= na) || (by >= nb))
        return;

    threadSumsA[tx] = 0.f;
    threadSumsB[tx] = 0.f;
    count[tx] = 0.f;

    for(offset = tx; offset < dim; offset += NUMTHREADS) {
        a = vectsA[bx * dim + offset];
        b = vectsB[by * dim + offset];
        if(!(isnan(a) || isnan(b))) {
            threadSumsA[tx] += a;
            threadSumsB[tx] += b;
            count[tx] += 1.f;
        }
    }
    __syncthreads();
    ...
}
```

1. Effective Parallelism

Launch configuration:
- Grid = (5,5) → 25 blocks
- Block = (32,32) → 1024 threads/block
- NUMTHREADS = 16 (used for per-block reductions)
- na = nb = 5, dim = 100

The kernel uses only threadIdx.x in [0, NUMTHREADS-1] for work and shared memory: threadSumsA[NUMTHREADS], threadSumsB[NUMTHREADS], count[NUMTHREADS].

Logically, each block has 16 active worker threads (tx = 0..15) operating over the dimension dim = 100, and all 25 blocks with bx = 0..4, by = 0..4 are active (because bx < na and by < nb).

2. Per-block Work and FLOP Count

Loop over dim:
for(offset = tx; offset < dim; offset += NUMTHREADS)
Each of the 16 threads covers distinct indices of [0, dim). For dim = 100 and NUMTHREADS = 16, there are in total 100 iterations of the body per block (each element processed once).

Inside the loop body, assuming no NaNs:
• 2 float loads (a, b).
• 3 additions (sums for A, B, and count).
• Some extra operations from isnan checks.

We approximate ≈ 5 FLOPs per iteration (adds + comparison-equivalents).

Per block:
F_block,loop ≈ 100 × 5 = 500 FLOPs.

Reduction:
A tree reduction over NUMTHREADS = 16 entries adds on the order of 45 float additions per block (for threadSumsA, threadSumsB, and count).

Final step (tx == 0):
Two divisions for means, plus a few stores, ~4 FLOPs.

Total FLOPs per block:
F_block ≈ 500 + 45 + 4 ≈ 549 ≈ 5.5×10^2 FLOPs.

With 25 active blocks:
F_total ≈ 25 × 549 ≈ 1.38 × 10■ FLOPs.

3. Memory Traffic

Per loop iteration (no NaNs):
• Load a = vectsA[...]: 4 bytes
• Load b = vectsB[...]: 4 bytes
→ 8 bytes per iteration.

There are 100 such iterations per block:
B_block,loop = 100 × 8 = 800 bytes.

Final writes (per block):
• means[2 values]: 2 floats → 8 bytes
• numPairs[1 value]: 4 bytes
→ 12 bytes.

Total per block:
B_block ≈ 800 + 12 = 812 bytes.

With 25 blocks:

B_total ≈ 25 × 812 ≈ 20,300 bytes ≈ 2.03 × 10■ bytes (~0.02 MiB).

Given how small this is, the kernel is overwhelmingly dominated by launch overhead.

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 ≈ 1.38 × 10■ FLOPs
• B_total ≈ 2.03 × 10■ 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 (CUDA kernel launch overhead).

GPU	t_compute (µs)	t_mem (µs)	t_body (µs)	t_total (µs)
GTX TITAN Black	0.0027	0.0604	0.0604	≈ 5.0604
GTX TITAN X	0.0022	0.0603	0.0603	≈ 5.0603
TITAN V	0.0009	0.0311	0.0311	≈ 5.0311
RTX 2080 Ti	0.0010	0.0330	0.0330	≈ 5.0330
RTX 4070	0.0005	0.0403	0.0403	≈ 5.0403

6. Conclusion

The gpuMeans kernel performs only on the order of 1.4 × 10■ FLOPs and moves about 0.02 MiB of data per launch. Both the compute and memory times are far below 0.1 µs on all GPUs considered.

Therefore the runtime is dominated almost entirely by the fixed CUDA kernel launch overhead (~5 µs).

Predicted per-launch times:
• GTX TITAN Black: ≈ 5.06 µs
• GTX TITAN X: ≈ 5.06 µs
• TITAN V: ≈ 5.03 µs
• RTX 2080 Ti: ≈ 5.03 µs

- RTX 4070: ≈ 5.04 µs

These values are approximate but consistent with expectations for a very small, reduction-style kernel.