CUDA Maximum Kernel Execution Time Estimates on Five GPUs

This report applies the generic CUDA kernel time prediction template to the given maximum_kernel and evaluates its approximate execution time on five GPUs: • GeForce GTX TITAN Black (6 GB)
• GeForce GTX TITAN X (12 GB)
• NVIDIA TITAN V (12 GB)
• GeForce RTX 2080 Ti (11 GB)
• GeForce RTX 4070 (12 GB)

The kernel configuration and loop structure are: • GridDim = (5, 5, 1) → 25 blocks
• BlockDim = (32, 32, 1) → 1024 threads per block
• Loop upper bound k = 100
• Only threadIdx.x participates in the loop; threadIdx.y duplicates work.

1. Work Analysis for maximum_kernel

Loop body:
for (size_t offset = threadIdx.x; offset < k; offset += blockDim.x) { float t = abs(vg_a[x * pitch_a + offset] - vg_b[y * pitch_b + offset]); temp[threadIdx.x] = max(temp[threadIdx.x], t); } Approximate per-iteration operations:
• 1 subtraction (vg_a - vg_b)
• 1 absolute value (abs)
• 1 max comparison/update
≈ 3 FLOPs per iteration (counting abs/max as simple FLOP-equivalent ops).

Per block:
• For a fixed threadIdx.y row, the 32 threads along x cooperatively cover k = 100 elements.
That gives exactly 100 iterations per row (sum over all threadIdx.x).
• There are 32 such rows (threadIdx.y = 0..31) doing the same work.
→ Iterations per block = 100 × 32 = 3,200

Across the whole grid:
• Blocks per grid = 5 × 5 = 25
→ Total iterations across the grid:
N_iters_total = 3,200 × 25 = 80,000

Total FLOPs:
FLOPs_total ≈ 80,000 × 3 = 240,000 FLOPs = 2.4×10■

Memory per iteration:
• Load 1 float from vg_a (4 bytes)
• Load 1 float from vg_b (4 bytes)
→ 8 bytes per iteration (ignoring shared memory).

Total bytes:
Bytes_total ≈ 80,000 × 8 = 640,000 bytes ≈ 6.4×10■ bytes (~0.61 MiB)
Final writes to d[...] add only ~3.2 KB and are negligible at this scale.

2. GPU Specifications Used

We use approximate published peak FP32 throughput and memory bandwidth:

• GeForce GTX TITAN Black: 5.12 TFLOPs, 336 GB/s
• GeForce GTX TITAN X: 6.14 TFLOPs, 336.5 GB/s
• NVIDIA TITAN V: 14.9 TFLOPs, 652.8 GB/s
• GeForce RTX 2080 Ti: 13.45 TFLOPs, 616 GB/s
• GeForce RTX 4070: 29 TFLOPs, 504 GB/s

We also assume a typical CUDA kernel launch latency:
t_launch ≈ 5 microseconds (µs) on modern NVIDIA GPUs.

3. Time Estimates per GPU

For each GPU:
• t_compute = FLOPs_total / Peak_FP32
• t_mem = Bytes_total / Bandwidth
• t_body = max(t_compute, t_mem)
• t_total ≈ t_body + t_launch

GPU	t_compute (µs)	t_mem (µs)	t_body (µs)	t_total ≈ (µs)
GeForce GTX TITAN Black	0.047	1.90	1.90	≈ 7.5
GeForce GTX TITAN X	0.039	1.90	1.90	≈ 7.0
NVIDIA TITAN V	0.016	0.98	0.98	≈ 6.0
GeForce RTX 2080 Ti	0.018	1.04	1.04	≈ 6.0
GeForce RTX 4070	0.008	1.27	1.27	≈ 6.0

4. Interpretation

Because the kernel performs only ~2.4×10■ FLOPs and moves ~0.61 MiB of data, the ideal compute and memory times are both on the order of 1–2 microseconds or less. For such a small kernel, the fixed kernel launch overhead (~5 µs) dominates.

As the GPU generation improves (TITAN Black → TITAN X → TITAN V → RTX 2080 Ti → RTX 4070), the memory-bound time t_mem shrinks, but the launch latency remains roughly the same. Therefore the total predicted time per launch stays clustered around 6–8 µs across all five GPUs.

These numbers are approximate, but the methodology is consistent with the provided calculation template and standard GPU performance modeling practice.