MP4 Solution:

1. (Fall 2021) Consider a 2D convolution tiling strategy in which each thread loads one value from input N from global memory to shared memory (Strategy 2). Consider 10x10 output tiles and a 7x7 convolution kernel (mask). In an internal tile (i.e., not near any boundary):

a. How many float multiplications are needed for each tile?

b. How many threads are turned off during calculation?

Answer a: (1pt)

Number of float multiplications: 10*10*7*7 = 4900.

Answer b: (1pt)

The size of the corresponding input tile is: (10 + 7 - 1) x (10 + 7 - 1) = 16 x 16.

Number of threads turned off during calculation: 16*16 - 10*10 = 156.

2. Consider a 3D convolution tiling strategy in which each thread loads one value from input N. Assume a 16x16x16 output tile and a 3x3x3 mask. For an internal tile (not near any boundary), what is the average reuse for each element loaded into shared memory?

Answer: 16 * 16 * 16 * 3 * 3 * 3 / (16 + 3 - 1) ^ 3 = 512/27 = 18.96296 (1pt)

3. Consider a 2D convolution tiling strategy in which each thread loads one value from the input matrix (Strategy 2). Output tiles are squares of size OUTPUT_WIDTH. Consider OUTPUT_WIDTH and MASK_WIDTH to be 16 and 9 respectively.

a. Assume that the GPU supports up to 2048 threads per SM. How many thread blocks can execute simultaneously on each SM? Please explain your answer.

b. How many bytes of shared memory are used on each SM? Assume that the input is an array of floats?

c. What is the average reuse for each value loaded to shared memory in an internal tile (not near any boundary)? Please explain your answer.

d. Consider the previous setup (same block size and MASK_WIDTH, but different input size and output size) where each thread loads 4 values (i.e. tiles of 2x2) into shared memory. With the number of threads in each thread block unchanged, in an internal tile, what is the new average reuse for each value loaded into the shared memory? Please explain your answer. (Hint: you can start from computing the new input tile size.)

Answer a: (1pt)

Each block is of size 24 * 24. Therefore, floor(2048 / (24 * 24)) = 3 blocks.

Answer b: (1pt)

Shared memory bytes = 24 * 24 * 3 * 4 = 6912 B.

Answer c: (1pt)

Reuse = (16 * 16 * 9 * 9) / (24 * 24) = 36

Answer d: (1pt)

First, we compute the new input and output tile sizes. The new shared memory size is the same as the new input tile size, which is 24*2 * 24*2 = 48 * 48. Therefore, the new output tile size is (48-9+1) * (48-9+1) = 40 * 40.

Reuse = (40 * 40 * 9 * 9) / (48 * 48) = 56.25

4. Consider 1D tiled convolution with mask width of 5 and output tile width of 32. Assume we use Strategy 1 (we use multiple steps to load input tile into shared memory). The use of shared memory reduces the number of global memory accesses.

What is the average reduction fraction in global memory accesses for thread block 0?

(A) 5

(B) 4

(C) 157/34

(D) 77/18

Answer: (C) (1pt)

Thread block 0 has 2 ghost elements.

Loading into shared memory requires 0 (ghost) + 32 (output tile elements) + 2 (halo) = 34 accesses. Without shared memory, total accesses would be 32*5 - 3 = 157. Thus, effective reduction is 157/34.

5. In a tiled 2D convolution using strategy 2 (each thread loads one element into shared memory, and during calculation we turn off some threads) with 8x8 output tiles and 9x9 mask. With warp size = 32, how many warps in each thread block have control divergence? We are only asking about thread blocks that don't deal with ghost elements.

(A) 16

(B) 6

(C) 4

(D) 2

Answer: (C). (1pt)

Each tile has 16x16 threads, and each tile has 8 warps. During calculation, only the 8x8 threads at the center is active, all the threads in the first 2 warps and in the last 2 warps are turned off. So, the 4 warps in the middle have control divergence.

6. For a tiled 3D convolution, assume that we load an entire input tile, including the halo elements into the shared memory when calculating an output tile. Further assume that the tiles are internal and thus do not involve any ghost elements. Assume input tile is a cube with width larger than 7 on each side, if the mask is a cube with 7 elements on each side, what is the trend

of the average number of times each input element will be accessed from the shared memory during the calculation as a function of the input tile width?

(A) Increases with input tile width with a limit of 7

(B) Increases with input tile width with a limit of 343

(C) Decreases with input tile width with a limit of 7

(D) Decreases with input tile width with a limit of 343

Answer: (B). (1pt)

Let input tile width = t, and the average number of times each input element will be accessed = f(t). Then f(t) = (t-(7-1))^3/(t^3) * 7^3 = ((t-6)/t)^3 * 343. Clearly f(t) increases as t increases, and f(t) < 343.