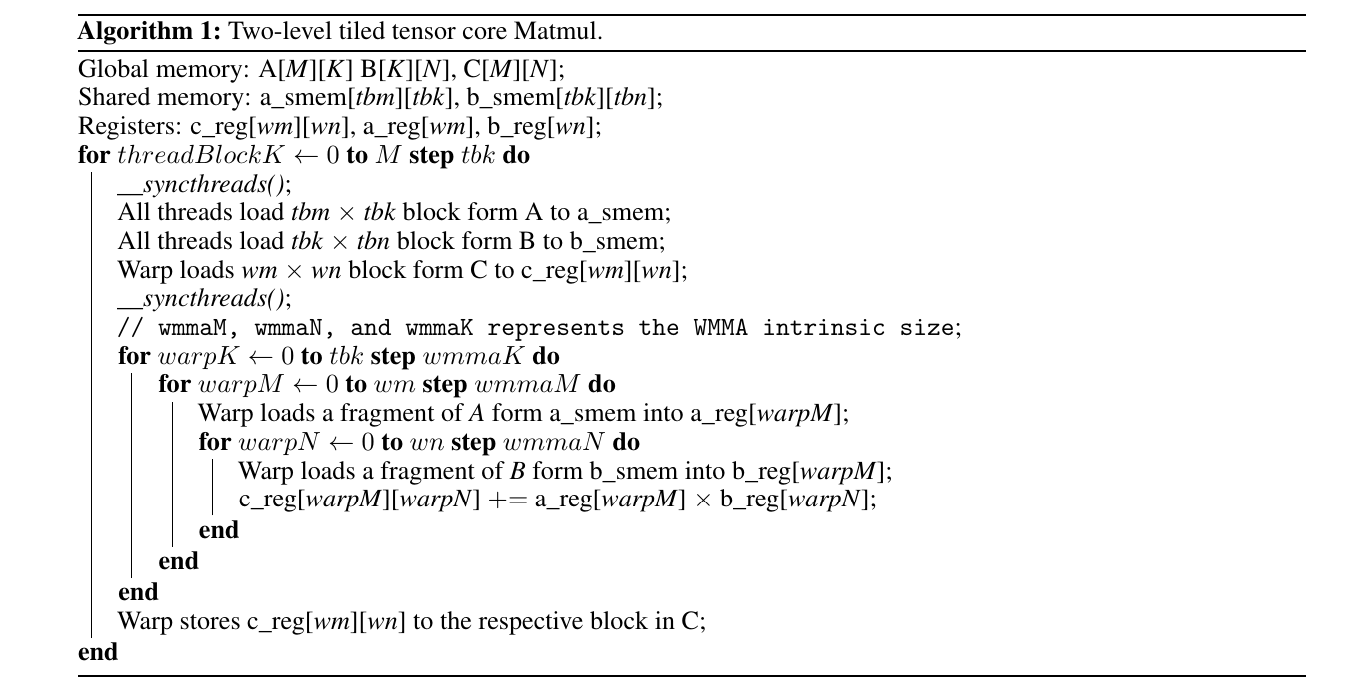


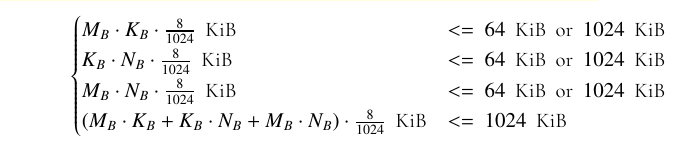
2088 is divisible by 2,3,4,6,8,9,12, etc.

Using fp32 (4 bytes) to match the width of Shared Memory bank since no two threads of the same warp should be accessing different 4 byte words in the same bank which causes bank conflicts.



Tiling Math:

On my GTX1650 GPU I have 1024 KiB L2 cache and 64 KiB of shared memory (ShaMem) per Streaming Multiprocessor (SM) so the math is



so starting points for tile sizes are:

M\_B = 464, N\_B = 512 and K\_B = 16

Which gets me fairly close to the limits of shared memory and L2 cache.

Tile of matrix A: 464 \* 16 \* 4 bytes = 29696 bytes

Tile of matrix B: 8 \* 512 \* 4 bytes = 32768 bytes

Combined = 62464 < 65536 bytes

Tile of matrix C: 464 \* 512 \* 4 bytes = 950272 bytes, /1024 b = 928 KiB which is < 1024 KiB

62464 / 65536 = 95.31%

928 KiB / 1024 KiB = 90.63%

But using the 464 dimension does not divide 2088 evenly so I need to pad the M dimension by 232 elements so the matrix dimensions are now:

2320 \* 2048 matmul 2048 \* 2048 = 2320 \* 2048

This will be the starting point from which I am going to optimize Cuda C++ code (if I have time) and the MLIR.