To: Christina Taylor
From: Eli Case
Date: April 25, 2023
Subject: CAAM 420/520 – Homework 5

Problem 2 Table 1 gives the performance statistics for each implementation.

Setup:	Block	Wrapped	Line
Elements/Thread	1	N	`n_rows`
Blocks/SM (thread-based)	2	$\frac{64}{N}$	64
Global Memory Transactions (per Block)	32	N	1
Global Memory Transactions (Entire Grid)	32(`N_rows`)(`N_cols`)	(`n_rows`)(`N_cols`)	(`n_rows`)(`N_cols`)

Table 1. Performance Statistics for Each Implementation

Problem 3 We have 64 kB per SM. Since a float is 4 bytes of data, we can process $16,000$ floats per SM.

(a) 7 blocks/SM

(b) The most blocks/SM is 166 for $N = 1$. The least blocks/SM is 1 for $N = 249$.

(c) The largest value is `n_rows` $= 249$.

Problem 4 (a) The kernel would use 400 thread blocks.

(b) The least number of thread blocks the kernel would use is 10 for $N = 40$. The most number of thread blocks is 12,800 for $N = 1$.

(c) The kernel would use 12,800 thread blocks.

(d) If two kernels need to be launched concurrently, the Line setup and Wrapped Line setup are most likely to allow two kernels to be run simultaneously. This is because these setups require less SMs than the Block setup because they have grids with larger blocks that request less SMs.

(e) If the GPU was upgraded to have more SMs, the Block setup and the Wrapped Line setup can achieve the biggest improvement in concurrency and runtime. This is because these setups have grids that can be configured to have smaller block sizes that can request more SMs.

(f) If the Wrapped Line setup is implemented with kernels invoked simultaneously, the factors necessary to consider for choosing an appropriate value of N would be as follows.

i. How small N can be so that there are enough SMs. If we choose an N that is too small, there will be too large of a number of blocks launched to invoke kernels concurrently. With too many thread locks, there will be a shortage of SMs for the parallelism to be efficient or possible for concurrent kernels. Also, if we launch too many thread blocks, there may be a sufficient amount of SMs for efficient parallelism, but too many reads/write to global memory per block for efficient parallelism. Thus, we have a restriction on the minimum value of N so that enough SMs will be available for simultaneous kernel invocation.

ii. How large N is so that the block is sized appropriately for a SM. If we choose an N too large, the block size may be too large to process one block per SM. This restriction largely has to do with the amount of threads available per SM. Additionally, block sizes that are too large can leave idle workers if SMs are available but the problem was not properly divided to utilize all of the SMs because of a larger than optimal block size. This, we would like a N that is not too large so that all the SMs may be utilized in the case of a simultaneous kernel invocation.