Wing Lee
CSC 656
12/08/24

# Step 10: Analysis Questions

1. What is the MFLOP/s performance gain going from the CPU-only code to the final version of your CUDA code (the one with the cudaMemPrefetchAsync() call)? Please report this gain in terms of a multiplier, e.g., 2.5x, rather than with an absolute number or a percentage. Show your work on how you compute this result.

   Comparing the CPU only MFLOPS to the final version of cuda we would do

   CPU only = 553.66
   Final version = 462.751

   $462.751/ 533.66 \ = \ 0.8358$

   So that means that the MFLOP/s performance gain would be 0.8358x

2. What is the memory bandwidth performance gain (or loss) going from the CPU-only code to the final version of your CUDA code (the one with the cudaMemPrefetchAsync() call)? Please report this gain in terms of a multiplier, e.g., 2.5x, rather than with an absolute number or a percentage. Show your work on how you compute this result.

   5.83/ 6.96 = 0.837

   There is performance loss of 0.837x

3. For the final version of your CUDA code (the one with the cudaMemPrefetchAsync() call), what is the total number of concurrent threads being run? Show your work on how you arrive at this result.

So we declared that the block size is 256, there a kernel with 1 block because of add<<<1, 256>>>(N, x, y);

So it would be number of blocks * threads in a block so it would be 1 * 256 = 256. Therefore the final cuda code has 256 concurrent threads

Table

| A | B | C | D |
|---|---|---|---|
| tion | Elapsed time (ms) | MFLOP/s | memory bandwidth |
| Addition | 882.14 | 553.66 | 6.96 |
| or Addition: 1 thread, 1 thread block | 51,829.42 | 9.4209 | 0.118 |
| or Addition: 256 threads, 1 thread block | 1,684.88 | 289.801 | 3.65 |
| or Addition: 256 threads/block, many blocks | 1,675.01 | 291.509 | 3.67 |
| or Addition: 256 threads/block, many blocks, explicit data movement | 1,055.17 | 462.751 | 5.83 |
| | | (ms)/1000 = seconds | |
| | | (# arithmetic ops/ seconds) / 1024*1024 | |