Assignment IV: Advanced CUDA

Aritra Bhakat

December 19, 2023

GitHub repo

Exercise 1: Thread Scheduling and Execution Efficiency

1. **Assume X=800 and Y=600. Assume that we decided to use a grid of 16X16 blocks. That is, each block is organized as a 2D 16X16 array of threads. How many warps will be generated during the execution of the kernel? How many warps will have control divergence?**

   We have a grid size of $\lceil (800, 600)/(16, 16) \rceil = (50, 18)$, adding up to a total of 900 blocks. Each block contains $\frac{16 \cdot 16}{32} = 8$ warps, giving us a total of 7200 generated warps. The last row of blocks (corresponding to pixels `(0, 592) - (800, 608)`) will have divergence within the blocks since half the threads within the block are outside the image. However, since warps are assigned linearly, each warp corresponds to 2 rows (16 + 16 threads). Therefore the warps divide evenly between rows, and there is no control divergence.

2. **Now assume X=600 and Y=800 instead, how many warps will have control divergence?**

   In this case the last column of blocks will diverge, `(592, 0) - (607, 0)`. In each warp within this region, half the threads will be inside the image (`592-599`), and half outside (`600-607`). Since each warp corresponds to 2 rows, there are $800/2 = 400$ warps with control divergence.

3. **Now assume X=600 and Y=799, how many warps will have control divergence?**

   Similarly to the last scenario, the last column of blocks but now also the last row of blocks will have divergence within the block. However, only the last warp of the last row of blocks will have control divergence, since only the last row (pixels `(0, 799) - (599, 799)`) is outside the image. These add up to $\lceil 600/16 \rceil = 38$ warps.

Since the bottom right corner is double counted, the total number of warps with control divergence is $400 + 38 - 1 = 437$.

Exercise 2: CUDA Streams

1. **Compared to the non-streamed vector addition, what performance gain do you get?**
2. **Use nvprof to collect traces and the NVIDIA Visual Profiler (nvvp) to visualize the overlap of communication and computation.**
3. **What is the impact of segment size on performance?**

Exercise 3: Heat Equation with using NVIDIA libraries

1. **Run the program with different dimX values. For each one, approximate the FLOPS (floating-point operation per second) achieved in computing the SMPV (sparse matrix multiplication).**
2. **Run the program with dimX=128 and vary nsteps from 100 to 10000. What do you observe?**
3. **Compare the performance with and without the prefetching in Unified Memory. How is the performance impact?**