CSCE 435 Fall 2021

Assignment 3: Bitonic Sort using CUDA

Instructions to compile and execute the bitonic sort CUDA code:

1. Upload both the files (**bitonic_sort.cu** and **bitonic.grace_job**) to your scratch directory after logging into grace portal.
2. Open the current directory in the terminal using the "Open in terminal" option on the top.
3. Authenticate using your net id's password and duo-2 factor authentication.
4. Load the CUDA module using the following command:
   module load CUDA
5. Compile the source file using the following command:
   nvcc bitonic_sort.cu -o bitonic
6. Run the batch file using the following command:
   sbatch bitonic.grace_job
7. After a job is complete, you'll be able to see the output in the output file corresponding to your jobid in the same directory as the source code.
   (You'll be able to find out whether a recent job has been completed or not by going to: grace dashboard > jobs > active jobs)

Important: In the code, the number of values that are to be sorted (NUM_VALS) = number of threads (THREADS) x number of blocks (BLOCKS) (line #16). They are initialized to 2^9 threads and 2^15 blocks which is equal to 2^24 numbers. So, if you change the number of threads, make sure you also change the number of blocks to keep the total number of values constant.

Assignment:

- Measure the time taken by the cudaMemCpy function for transferring data (both, host to device, and device to host) and plot **time taken vs number of threads** (64, 128, 512, and 1024) for 3 sizes of NUM_VALS (2^16, 2^20, and 2^24). **[35 points]**
  - You can plot both of the memcpy times (host to device, and device to host) on the same graph. There will be 3 plots for 3 different NUM_VALS.
  - Make sure you also change BLOCKS when you're changing THREADS so as to keep NUM_VALS constant.

- Measure the time taken by the kernel (bitonic_sort_step) to execute (Hint: use cudaDeviceSynchronize()) and plot **time taken vs number of threads** (64, 128, 512, and 1024) for 3 sizes of NUM_VALS (2^16, 2^20, and 2^24). **[35 points]**
  - There will be 3 plots in total for 3 different NUM_VALS.

- Calculate the effective bandwidth of the kernel (in GB/s) for different numbers of threads (64, 128, 512, and 1024) keeping NUM_VALS = 2^24. **[30 points]**

For reference: https://developer.nvidia.com/blog/how-implement-performance-metrics-cuda-cc/

Submission:

Upload a pdf on canvas containing your answers.