UCS 645 : Parallel & Distributed Computing

Lab Exercise #4

Learning Outcomes:

- **How to query CUDA device properties.**
- **Understanding how to observe the difference between theoretical and measure memory bandwidth.**

Problem 1: Write a CUDA program for very basic implementation of vector addition kernel. Compute the profiling of object using nvprof Complete the following changes

1.1 Modify the example to use statically defined global variables (i.e. where the size is declared at compile time and you do not need to use cudaMalloc). Note: A device symbol (statically defined CUDA memory) is not the same as a device address in the host code. Passing a symbol as an argument to the kernel launch will cause invalid memory accesses in the kernel.

1.2 Modify the code to record timing data of the kernel execution. Print this data to the console.

1.3 We would like to query the device properties so that we can calculate the theoretical memory bandwidth of the device. The formal for theoretical bandwidth is given by; theoreticalBW = memoryClockRate ∗ memoryBusWidth

Using cudaDeviceProp query the two values from the first cudaDevice available and multiply these by two (as DDR memory is double pumped, hence the name) to calculate the theoretical bandwidth. Print the theoretical bandwidth to the console in GB\s (Giga Bytes per second). Note that the above will calculate the result in kilobits\second (as memoryClockRate is measured in kilohertz and memoryBusWidth is measured in bits). You will need to convert to the memory clock rate to Gb\s (Gigabits per second) and then convert this to GB\s.

1.4 Theoretical bandwidth is the maximum bandwidth we could achieve in ideal conditions. We will learn more about improving bandwidth in later lectures. For now we would like to calculate the measure bandwidth of the vectorAdd kernel. Measure bandwidth is given by;

measuredBW = (RBytes + WBytes)/t

Where RBytes is the number of bytes read and WBytes is the number of bytes written by the kernel. You can calculate these values by considering how many bytes the kernel reads and writes and multiplying it by the number of threads that are launched. The value t is given by your timing data in ms you will need to convert this to seconds to give the bandwidth in GB\s. Print the value to the console so that you can compare it with the theoretical bandwidth. Note: Don’t forget to switch to Release mode to profile your code execution times.