

ME759
High Performance Computing for Engineering Applications
Data Parallel Reduction Assignment

Edit the source files **vector_reduction.cu** and **vector_reduction_kernel.cu** to complete the functionality of the parallel addition reduction on the device (array reduction means summing up all the entries in the array and returning the result). The size of the array is guaranteed to be equal to 1024 elements for this assignment.

There are two modes of operation for the application.

a) No arguments: The application will create a randomly initialized array to process. After the device kernel is invoked, it will compute the correct solution value using the CPU, and compare that solution with the device-computed solution. If it matches (within a certain tolerance), it will print out "**Test PASSED**" to the screen before exiting.

b) One argument: The application will initialize the input array with the values found in the file provided as an argument.

In either case, the program will print out the final result of the CPU and GPU computations, and whether or not the comparison passed.

The **ArrayReduction** folder should contain all the changes and additions you have made to the source code. In addition, it should include a **pdf** file with your answers to the following questions:

1) How many times does your thread block synchronize to reduce the array of 1024 elements to a single value?

Answer:

The reduction is taking place by the factor of 2, so the thread block will synchronize 10 times to reach to a single value.

2) Over the life of the program, how many warps will be adversely affected by warp divergence? Warp divergence happens when threads within a warp take different code paths through the program.

Answer:

The number of warps that would be affected would be after the calculation from $32 = 2^5$. So, the total no. of warps that would be affected are 5.

Grading:

Your submission will be graded using the following scheme:

a) Demo/knowledge: 25%

- Produces correct result output for test inputs

b) Functionality: 40%

- Uses an $O(N)$ operation data-parallel reduction algorithm, where N is the number of elements in the array

c) Report: 35%

- Answer to question 1: 15%, answer to question 2: 20%