Lab4 Report

Ke Xu 604761427

Briefly explain the parallel strategies you applied in this lab. How is it parallelized? What is the expected communication overhead?

I implement multiple thread blocks (work groups) with multiple threads (work items) in each of them. Every thread deals with different sets of data and does parallel computing. I implement enough parallelism to hide the latency of fetching data from global to local.
I just partition the very outer loop to each different work groups and each item compute one row of the outcome. The communication overhead includes fetching data which are weight, Cin and bias from the global memory and write Cout back to it.

Evaluate your program in terms of the execution time, and make a comparison with your lab-4 submission. Does the GPU perform better for this task? If yes, explain why you think it does.

The execution time is 2.33s approximately, which includes the kernel computing time and the setup time that is more than 1 second. You mean lab-3? It is faster than lab-3 because of the more parallelism GPU can support. The GPU has much more work items than CPU and can highly hide the memory latency.

Discuss how much memory is used at private, local, and global levels of your best configuration.

Local: I try my best to use it as much as possible. I use 96*96*4=37kB.
Private: I implement 5*5*256*4(weight)+228*228*4(Cin) =232.5kB.
Global: It has to store all the data and turns out to be 228*228*4*256+256*256*5*5*4+256*4=55MB.

Challenges you encountered and how you solved them.

1. Sometimes when I terminate the AWS at the end of a day, I forgot to download the code I wrote so I waste one day.
2. I had some errors caused by synchronization. So I implement barrier(CLK_LOCAL_MEM_FENCE) to get the right answer.