ICS-4020 Programming Parallel Computers
Week 5

Kristian Hartikainen (222956)
kristian.hartikainen@aalto.fi

May 17, 2015

1 Introduction

I extended the last week's cp8 solution to speed up the matrix dot product calculation. In outline, the fuctionality is quite the same as in last week's solution. The speed ups are achieved by efficiently utilizing the shared cache of the GPU's streaming multiprocessors, and the threads' local caches.

The matrix multiplication is done by dividing the output matrix into squares of size mk * mk, which presents the calculations done in each cuda block. These mk * mk squares are again divided into k * k squares which correspond to each cuda thread. Thus, each thread has to fetch 2*k data points to the cuda block cache, and it calculates k*k points for the result matrix. With this hardware, values m=25 and k=5 turned out to be succifiently fast. These were determined by first trying out different k values such that the stored values still fit into the thread cache on this hardware. After that, m was incremented until the block cache ran out.

In my solution, I pad the normalized matrix such that it's x and y dimensions are both divisible my m*k. This way I don't have to do any conditional operations in the kernels, and thus reduce the run time.

The benchmarks were run on the classroom computer 'kiwi'. The benchmarks and the plot for are shown in the results section.

The fastest running time for 4000*4000 matrix was 0.795s.

2 Results

ny	nx	time
1	1	0.386
1	1	0.000
1	10	0.388
1	10	0.000
1	100	0.393
1	100	0.001
1	1000	0.392
1	1000	0.001
1	2000	0.404
1	2000	0.003
1	4000	0.390
1	4000	0.004
10	1	0.395
10	1	0.000
10	10	0.403
10	10	0.001
10	100	0.394
10	100	0.001
10	1000	0.389
10	1000	0.002
10	2000	0.391
10	2000	0.003
10	4000	0.391
10	4000	0.005
100	1	0.392
100	1	0.001
100	10	0.388
100	10	0.001
100	100	0.393
100	100	0.001
100	1000	0.398
100	1000	0.007
100	2000	0.401
100	2000	0.008
100	4000	0.402
100	4000	0.012
1000	1	0.397
1000	1	0.011
1000	10	0.401
1000	10	0.011
1000	100	0.392
1000	100	0.012
1000	1000	0.425
1000	1000	0.040
1000	2000	0.458

ny	nx	time
1000	2000	0.070
1000	4000	0.524
1000	4000	0.129
2000	1	0.411
2000	1	0.024
2000	10	0.413
2000	10	0.025
2000	100	0.421
2000	100	0.028
2000	1000	0.487
2000	1000	0.094
2000	2000	0.556
2000	2000	0.163
2000	4000	0.687
2000	4000	0.303
4000	1	0.450
4000	1	0.064
4000	10	0.455
4000	10	0.065
4000	100	0.451
4000	100	0.071
4000	1000	0.637
4000	1000	0.246
4000	2000	0.810
4000	2000	0.430
4000	4000	1.176
4000	4000	0.795

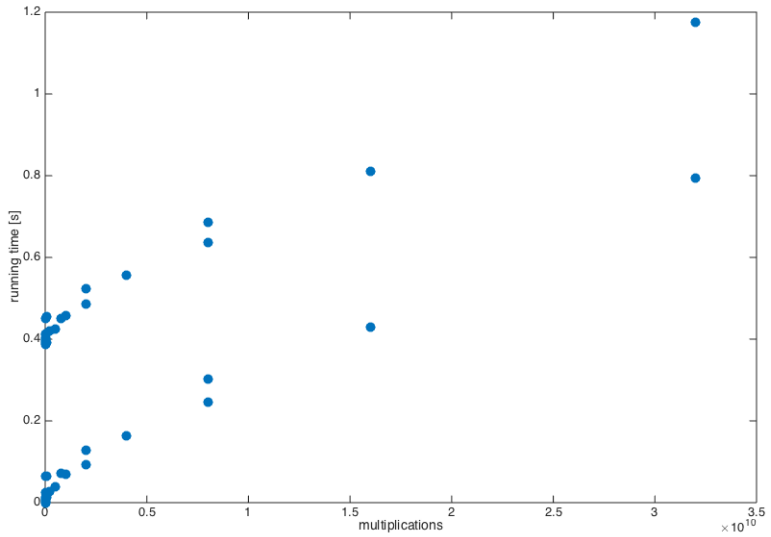

Figure 1: Results for cp9: Multiplications vs. Running time in seconds