

ICS-4020 Programming Parallel Computers

Week 4

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

May 8, 2015

1 Introduction

I took the old code from `cp1`, and changed it to work on GPU, using CUDA. Basically, the only thing I did was to move the matrix multiplication part of the task to GPU. This is implemented in `matrix_multiply` -function, and called from the same `correlate` function as where it was done in `cp1`. `correlate` -function copies the matrix to be multiplied from the host memory in to the device memory, do the multiplication, and copy the results from the device to the host.

Then benchmarks were run on the classroom computer 'Kiwi', and different block sizes (arbitrarily chosen to be powers of two) were tested. Benchmarks for are shown in the table 1, and the corresponding visualizations in figure 1.

Fastest running time, 0.847s, was achieved on 8x8 block size.

2 Results

ny	nx	1*1	2*2	4*4	8*8	16*16	32*32
1	1	0.044	0.041	0.051	0.053	0.043	0.092
1	10	0.036	0.035	0.038	0.047	0.042	0.035
1	100	0.038	0.045	0.039	0.037	0.043	0.046
1	500	0.038	0.037	0.040	0.042	0.040	0.040
1	1000	0.033	0.040	0.038	0.042	0.037	0.041
1	1500	0.036	0.040	0.036	0.045	0.040	0.040
10	1	0.035	0.037	0.038	0.042	0.036	0.042
10	10	0.036	0.040	0.034	0.044	0.039	0.043
10	100	0.035	0.041	0.036	0.045	0.041	0.039
10	500	0.043	0.038	0.038	0.039	0.039	0.039
10	1000	0.033	0.038	0.036	0.041	0.040	0.039
10	1500	0.038	0.034	0.037	0.043	0.042	0.042
100	1	0.034	0.035	0.037	0.042	0.038	0.033
100	10	0.037	0.037	0.039	0.039	0.039	0.036
100	100	0.036	0.038	0.036	0.044	0.039	0.038
100	500	0.051	0.042	0.041	0.044	0.051	0.041
100	1000	0.065	0.050	0.050	0.055	0.055	0.054
100	1500	0.085	0.055	0.045	0.049	0.057	0.055
500	1	0.041	0.037	0.040	0.047	0.039	0.039
500	10	0.045	0.036	0.038	0.038	0.042	0.040
500	100	0.108	0.053	0.045	0.053	0.056	0.057
500	500	0.397	0.139	0.081	0.084	0.092	0.116
500	1000	0.746	0.236	0.124	0.111	0.128	0.199
500	1500	1.096	0.329	0.164	0.140	0.180	0.271
1000	1	0.050	0.042	0.037	0.038	0.038	0.038
1000	10	0.074	0.046	0.039	0.040	0.046	0.044
1000	100	0.328	0.117	0.071	0.057	0.081	0.101
1000	500	1.450	0.426	0.198	0.159	0.210	0.334
1000	1000	2.847	0.807	0.349	0.287	0.393	0.632
1000	1500	4.240	1.181	0.510	0.414	0.559	0.920
1500	1	0.074	0.051	0.041	0.041	0.046	0.038
1500	10	0.123	0.060	0.041	0.055	0.045	0.047
1500	100	0.696	0.213	0.110	0.097	0.126	0.174
1500	500	3.204	0.887	0.388	0.304	0.417	0.695
1500	1000	6.349	1.730	0.731	0.574	0.805	1.345
1500	1500	9.472	2.562	1.075	0.847	1.189	1.995

Table 1: Results for cp8. nx and ny represent the problem size, and x*y present the block size used. Powers of two for the block sides was chosen arbitrarily.

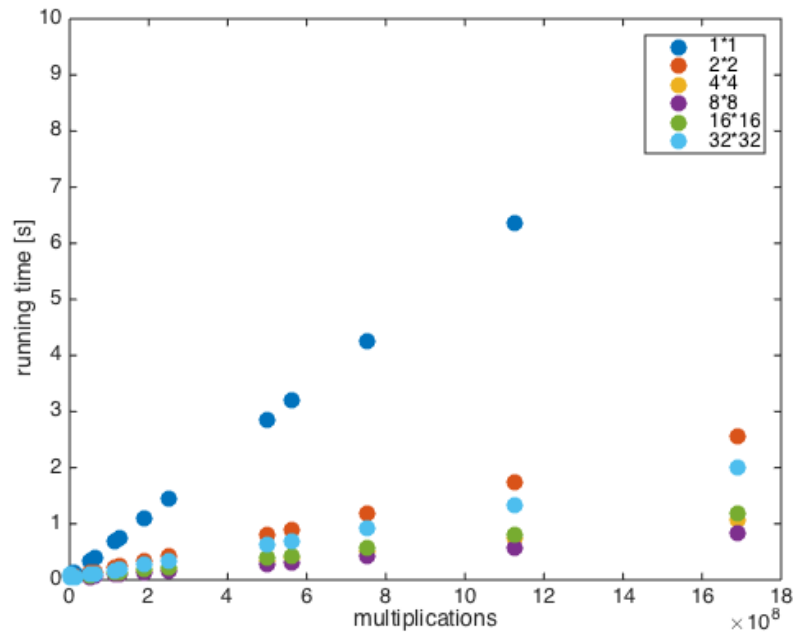


Figure 1: Results for cp8: Multiplications vs. Running time in seconds