Parallel Computing II: Homework II

17. November 2014

This will be your second homework in the exercise parallel computing. Send your solution to Matthias.huy@daimonas.de and to t.grahs@tu-braunschweig.de until November 24th 08.00pm. Prepare a pdf file for your written text and attach the source code of your program to the mail.

Task I (25 points)

Matrix Multiplication

In this task you are going to implement a matrix-matrix multiplication. Use a decent memory mapping of the two dimensional matrix to the one dimension array and divide the matrix into tiles. Do

1. a cpu/host version of the matrix-matrix multiplication
2. a gpu/device kernel using global memory
3. a gpu/device kernel using shared memory

Use a timer (f.i. `Timer` class from `cuda_utils.h`) to measure the performance of your code i.e compare the different version of the matrix-matrix multiplication with each other. Try different kernel execution parameters, i.e. grid and thread block sizes, and different matrix sizes gridsize, blocksize and size of the matrix. Plot your results in graphs.

Document your approaches and results **in detail** (description. graphs, etc.)!

Usage of `Timer` class

```
Timer timer;
initTimer(&timer);

... code_to_measure ...

cudaThreadSyncronize();
double duration=getTimer(&timer);
```

(Include `cuda_utils.h` in your code.
Put the `cuda_utils.h` in the same directory like your .cu-file and compile with option `-I.`