CSE746 - programming exercise for Lecture 2

Pawel Pomorski

Jan. 14, 2015

The files needed for this exercise can be found in
/home/ppomorsk/CSE746_lec2
The access to directories with solved problems is disabled for now. Use monk node mon54 to compile and run.

1 SAXPY

A number of example implementations of SAXPY have been provided in directory saxpy. Experiment with compiling and running these for different vector sizes. Practice timing the code. Estimate how much time is spent transferring data to/from CPU vs the actual computation time on the GPU.

Modify the timed version of the saxpy code to use cudaMallocHost instead of malloc to allocate host memory (uses page-locked memory for higher performance). Use timing to determine the improvement in performance this gives.

Instructions for compiling are given in the initial comments of the program files. The CUDA and CUBLAS code provided here are good starting templates for problem 2.

2 Visualizing the Julia set

The serial CPU program julia_cpu.cu generates a visualization of the Julia set. Using this program as a starting point, write a CUDA program which accomplishes the same thing.

The Julia set is obtained by generating a sequence starting from a complex number Z_0

$$Z_{n+1} = Z_n^2 + C \tag{1}$$

where C is a complex parameter. If this sequence converges then the starting point Z_0 is in the Julia set. The set can be visualized in the complex plane, producing interesting patterns. The starting point is set up to produce an output file which can be viewed in gnuplot.

Performance is not an issue in this exercise, so you do not need to time this code. As this may be your first GPU code and the aim is to keep things simple, you may choose to use block size 1 in this exercise.