P. Koumoutsakos, S. Martin
ETH Zentrum, CLT E 13
CH-8092 Zürich

HW 6: GP-GPU Programming with CUDA

Issued: May 13, 2019
Due Date: May 31, 2019 10:00am

Warm-Up & Information

This homework requires the use of multiple computational nodes. Feel free to use either Euler or Piz Daint for your solution. Along with this pdf, you should download the code containing:

- *task1/* The skeleton code for Task 1.
- *task2/* The skeleton code for Task 2.

Grading

- CSE students: Max 150 points to be converted linearly to a 1-to-6 scale.
- Engineering students: Max 100 points to be converted linearly to a 1-to-6 scale.
- Bonus: A correct answer to task **2b** will grant 0.5 grade points to improve the grade of this or another homework, up to 6 points.

Guidelines for Moodle submission:

- Upload all your .cpp code files to Moodle before the deadline.
- Your report in PDF format. Do not forget to answer every question.

Task 1: (100 Points) Advanced MPI

By means of office gossiping, you learn that the chassis manufactoring company's board of investors is considering you as their new CTO (Chief Technical Officer), a coveted position with a significant pay increase. To convince the board that you are the perfect candidate for a CTO, you plan to demonstrate that the heat2D model can be made into a distributed application, capable of executing in a hundred of nodes simultaneously. This step will place your company as industry leader and solidify your career for years to come.

The goal of this task is to use MPI to completely parallelize the execution of the heat2D model. That is, every kernel in the solver should be parallel. To achieve this goal, proceed in the following order:

a) Working with a single grid (gridCount = 1), use MPI to parallelize the *applyJacobi* Kernel.[1]

b) Still working with a single grid, use MPI to parallelize the residual and *calculateL2Norm* kernels.

c) Set the number of grids to 2 and use MPI to parallelize the *applyRestriction* and *applyProlongation* kernels.

d) Make sure your solution works for gridCount = 4 number of grids.

For each one of these items, provide a brief explanation of the challenges you faced and the solutions you implemented. The conditions for a correct answer are:

1. To verify correctness, make sure the resulting residual L2Norm is the same as the base (sequential) algorithm.
2. Use MPI's cartesian grid-based communicators to establish the problem's geometry and each rank's neighbors.
3. Use MPI's custom datatypes to communicate both contiguous and non-contiguous boundaries.

Run the new model on 1, 2, and 4 full nodes of the Euler or Piz Daint supercomputers.[2] Analyze your timing results regarding strong scaling speed-up and efficiency both at a whole solver and a per-kernel level.

[1] Hint: Make sure you communicate all necessary boundary information to neighboring ranks before and/or after the jacobi step.

[2] You are allowed (and even encouraged) to reduce the problem size during development to speedup your work. However, please report your results using the grid size provided in the skeleton code.

Task 2: (50 Points) Communication-Tolerant Programming

»Note: Mandatory for CSE Students, Optional for Engineers«

In this exercise, you will apply a threading model to a given MPI application to reduce the cost of intra-node communication. Use the threading model of your choice (e.g., openMP, Pthreads).

a) Implement the approach described in the lecture slide "Hybrid Model: Approach I"[3].

b) (Optional - 0.5pt Bonus) Implement the approach described in the lecture slide "Hybrid Model: Approach II"[4].

For your solution(s) to be correct, the resulting L2Norm should be exactly the same as the base code in all runs. If your code produces the correct result only in a subset of runs, meaning it may contain concurrency problems, it will not be considered correct.

Compare the performance of your intra-node communication-tolerant version(s) with the provided base version on 1, 2, and 4 full Euler or Piz Daint nodes. What is the observed effect of these approaches on communication?

Find a way to differentiate the effect of your optimization on intra-node communication vs. network communication. Had your optimization any positive effect on reducing network communication?

[3]Slide 16 of the (13.05.2019) Communication-Tolerant Programming lecture

[4]Slide 17 of the (13.05.2019) Communication-Tolerant Programming lecture