CS 5334 Spring 2021
Lab 3 Assignment – 50 points
Accelerator Programming – Heat Equation
Assigned February 24, 2021
Due by 11:59pm on Sunday, March 7, 2021

March 10, 2021

For this lab, you will use shared memory and GPU parallel programming to accelerate execution of a program that solves the 2D heat equation. You should implement parallel versions for the multicore x86_64 CPU and NVIDIA GPU architectures. You should compile and run your parallel programs on a Bridges-2 node and report the speedups that you obtain. You may use your choice of shared memory parallel programming model for the AMD EPYC multicore CPU and your choice of GPU programming model for the NVIDIA V100 GPU. Prizes will be given for maximum speedup obtained on the CPU and on the GPU.

Serial/Parallel ~ Speedup

A serial code is provided for you in the labs/lab3/heat2d.c file in the github repository at https://github.com/mooresv/labs/lab3 . The code solves the steady state heat equation on a rectangular region. Given an approximate solution of the equation, a better approximation is obtained by replacing the solution value at each interior point by the average of its four neighbors. This iterative process is repeated until the difference between successive approximations is below the tolerance **epsilon**. epsilon is hard-coded to be 1e-3, but you may change this value if needed to illustrate speedup. The serial version needs approximately 18/epsilon iterations to complete. You must parallelize this code – i.e., you cannot find another parallel program for the 2D heat equation on the Web and turn that in instead of doing your own parallelization.

You should turn in both your code and a writeup describing your choice(s) of parallel programming models and your results. You may discuss the lab with other class members and ask questions about the lab in class. You are encouraged to discuss the lab on the Lab 3 collaboration page on Teams. However, you should do your own original work on your code and writeup.

The lab will be graded as follows:

1. (10 points) Description of your strategy for the shared memory parallelization and optimization for the CPU. Please describe what programming model you chose and how you applied it. Please also describe the steps you went through in optimizing the performance of your program.
2. (10 points) Description of your strategy for the GPU parallelization and optimization. Please describe what programming model you chose and how you applied it. Please also describe the steps you went through in optimizing the performance of your program.
3. (10 points) Correct operation of the CPU and GPU versions.
4. (10 points) Acceptable speedup on the CPU
5. (10 points) Acceptable speedup on the GPU