CS 4444/6444 Spring 2017
Assignment 3: Heated Plate with MPI
Due by 5:00 PM on Tuesday, March 14

Introduction

Your assignment is to develop a program, "halo" that performs the classic heated plate problem with a slight twist. The twist is that we want to use the program to measure the performance effects of different decompositions, thickness of boundary layers, and amount of computation per cell. Your program should be called "halo", and take six parameters:

- **x_dim** – the number of chunks in the X dimension.
- **iters_per_cell** – the number of inner loop iterations to "compute" per cell formed from the chunk divisions. This is used to increase the granularity of the code. The minimum value is "1".
- **iterations_per_snapshot** – every "iterations_per_snapsot" output the matrix to snapshot.X where "X" is the iteration number. The format of the file should be PPM format. That way you can use some program like xv to view the sequence of files as a "movie". To keep the image file from growing too large, scale the snapshot so that it is at most 1,000X1,000. Scale by averaging down.
- **iterations** – how many iterations to perform
- **boundary_thickness** – how many ghost cell layers to send at a time – and how many internal iterations to perform per communication.

The interior cells should all initialized to the same temperature (50 degrees), and the border cells fixed at a specific temperature (0 degrees along the top and left sides and 100 degrees along the bottom and right sides). In each time step of the simulation, the temperature of each cell is computed by averaging the temperatures of the four neighboring cells in the previous time step. To spice it up this year we will add a single internal condition in which a single cell is held at 1000 degrees. The single cell is at position row=4500, column = 6500. If those coordinates are outside the specified plate size then it is ignored.

You are to capture performance data 10,000X10,000 problem size, and do 10,000 iterations for a 1D decomposition.

Your submitted tar file will be called *<userid>.hw3.tar*. For example, my homework would be *ag8t.hw3.tar*. The tar file MUST include all sources and a make file. We will call "make", so make sure it makes.

Your write-up should include performance tables and plots (i.e. execution time and speedup) for one sequential execution and the following parameter values:

• Iters_per_cell of 1, 2, 4
• Boundary thickness of 1, 2, 4
• 20, 100, 200 processors

This will result in 27 parallel executions.

Hint: The attached reading from Chapter 13 Section 4 of Quinn's *Parallel Programming* book is a good starting point for this assignment. Suggestion: Develop your parallel code on much smaller problems sizes and a small number of processors, e.g., 4 or 8. Make sure it is robust before beginning the large runs.

Files needed

To start, download from the course Collab site the following files:

•heated_plate.c: this is the file that you will need to parallelize.

• ground_truth.ppm: a snapshot generated by the sequential implementation of a 10,000 x 10,000 heated plate after 10,000 iterations. Use this to verify that your parallel implementation is producing the correct results.

• create_jpegs.pl: run this Perl script to convert PPM snapshots of the heated plate into JPEG images.

Homework point breakdown:

50 points for the write up

50 points for code

- 10 points for comments
- 20 points for working MPI implementation (bare bones)
- 20 points for performance optimizations (e.g. proper exchanges, asynch messaging)

The performance winner will get a "This is SuperComputing" T-shirt and a "Virginia Supercomputing" T-Shirt. The runner up will get a "Virginia Supercomputing" T-Shirt.