**Portland State University – ECE 588/688**

**Winter 2024**

**Homework 3 Notes**

Due Week 6 Sunday, Feb 18, 2024 Before 11:59 PM

Submit to Canvas

Notes are marked in blue color below in this file.

Write a program using pthreads to estimate the temperature of all points on a grid in a similar fashion to the algorithm provided in the parallel computing program:

[https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial##ExamplesHeat](https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial)

You can start out by modifying the sample “sum.c” code, or write your own code from scratch.

You should use the following parameters:

1. Grid size = 1000 x 1000, spanning all points in the square between coordinates (1,1) and (1000,1000).
2. Initial condition: All center points in the region (200, 200) to (800, 800) have a temperature of 500 degrees, and all other points have a temperature of zero. Points outside the grid (i.e., neighbors of points on the boundary) always have a temperature of zero that does not change.
3. At each time step *t*, the temperature of a point at coordinates (x,y) is computed from the temperatures of the neighboring points in the previous time step *(t-1)* according to the following equation:

T(x,y)*(t)* = T(x,y)*(t-1)*

+ Cx \* (T(x+1,y)*(t-1)* + T(x-1,y)*(t-1)* – 2 T(x,y)*(t-1)*)

+ Cy \* (T(x,y+1) *(t-1)* + T(x,y-1) *(t-1)* – 2 T(x,y) *(t-1)*)

Where Cx=0.125 and Cy=0.11

1. Run the program for 4000 time steps. Note that depending on how you split your data, you may need to communicate information to neighboring processors after each time step.
2. After each 200 time steps, you should print the temperatures of the following points: (1, 1), (150,150), (400, 400), (500, 500), (750, 750), and (900,900).

You will get a better score if you do a good job parallelizing your algorithm to get a high parallel speedup.

You should turn in (1) a text file that contains the table of parallel speedups, and (2) a file with your C program. The file name has to be “YourLastname\_homework3.c”. Your program should run one command line parameters: the number of processors. When your program runs, it should produce one line of output for each of the points requested, and a line that displays runtime in this format:

Time = 98570173000 nanoseconds (98.5701733 sec)

**Homework3 Notes:**

HW3 is another problem you need to write a parallel program for, that problem is a bit more complex than HW2. It is a heat transfer for a grid, there is description of that problem in LLNL website.

(Show LLNL “Simple Heat Equation” top graph) We have this grid and this is a 2-D array, the value of each element in the array in the case of temperature of that element, and in each time as time goes by, heat transfers from the hotter elements of the array to the colder elements of the array. The way the heat is transferred is using this equation:

Uxy = Uxy + Cx \* (Ux+1,y + Ux-1,y - 2 \* Uxy) + Cy \* (Ux,y+1 + Yx,y-1 – 2\*Uxy)

The new temperature of a point in the array is equal to the old temperature plus some coefficient for heat transfer along the X-axis multiplied by the two values to the left and right minus 2 of the original value here (point to the blue square in the 2nd chart Ux,y), and the heat transfer across the Y-axsis is this coefficient Cy multiplied by the sum of these 2 temperatures (points to the 2nd chart the 2 pink square in vertical direction Ux,y+1 and Ux,y-1) minus 2 of the original value here (points to the middle blue square Ux,y). This is how heat is transferred across the array.

The way this works is: in the sequential program, what happens is we have a bunch of time steps, for each time step, we compute the temperature of each point in the array based on the temperature of the previous time step for all the 5 points that are participate in this computation equation. As we pass one time step to the next, heat is going to be transferred from the hot parts to the cold parts, so the hot parts will be colder and the cold parts will be a little bit warmer.

So the homework here is asking us to put together a grid which is 1000x1000 in size, the initial condition of that grid is all the points in the center region which is between coordinate of (200,200) to coordinate of (800, 800) have a temperature of 500 degrees; and all the other points having a temperature of 0 degree. We can assume the points on the boundary and points outside the boundary is always 0 degree.

Then the temperature for each time step is computed by using the temperature of the previous time step, basically using that same heat transfer equation. We have some value of coefficient of transferring heat across X axis Cx and across Y axis Cy.

You are required to run this program 4000 time steps. For each time step, you need to compute the temperature for all the points. After each 200 time steps, you should print the temperature of these coordinates listed in e).

You can write a sequential version of the program. The question is how you can parallelize this program to improve speed up.

This homework is about Pthreads, the first thing we need to think about is how we are going to divide the work between different threads. There are problems which we can do functional decomposition, where we have each thread do part of the function that is needed for the problem. This is not one of those problem. For this problem, we do domain decomposition. We divide the data between the different threads and have each thread operates on part of the data.

This homework 3 is trying to compute the heat that is being transferred between neighboring points in a system.