CMSC 180
Introduction to Parallel Computing
Second Semester AY 2023-2024

Laboratory Research Problem 03
Core-affine Threaded Computation of the Pearson Correlation Coefficient

Introduction

Rewrite the program that you wrote in Laboratory Research Problem 02 (LRP02) so that the threads that you created each must be ran specifically unto one specific core only. If your computing machine has four or eight threads, assign your threads to three or seven cores, respectively.

Research Activity 1: How to do it?

1. Write the main program `lab03` (just rewrite `lab02`) that includes the following:
   (1) Read n and t as user inputs (maybe from a command line or as a data stream), where n is the size of the square matrix, t is the number of threads to create, and $n >> t$;
   (2) Create a non-zero $n \times n$ square matrix **X** whose elements are assigned with random integer;
   (3) Create an $n \times 1$ vector $\boldsymbol{y}$ and a $1 \times n$ vector $\boldsymbol{r}$ as in `lab01` and `lab02`;
   (4) Divide your **X** into t submatrices of size $n \times n/t$ each, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t$;
   (5) Take note of the system time `time_before`;
   (6) Create t threads, where for the ith thread call `pearson_cor`($\boldsymbol{x}_i$, $\boldsymbol{y}$, n, n/t);← very important
   (7) Assign the thread to a core;← very important, this is the additional step in this exercise
   (8) Recreate the correct $\boldsymbol{r}$ from the output of each threaded `pearson_cor()`;← very important
   (9) Take note of the system time `time_after`;
   (10) Obtain the elapsed time `time_elapsed:=time_after - time_before`;
   (11) Output `time_elapsed`;

2. Fill in the following table with your time readings:

n	t	Time Elapsed (seconds)			Average Runtime (seconds)
		Run 1	Run 2	Run 3	
25,000	1*				
25,000	2				
25,000	4				
25,000	8				
25,000	16				
25,000	32				
25,000	64				

Research Question 1: What is the difference of the average time that you obtained for $t = 1$ compared to the average that was obtained in LRP01 and LRP02?

Research Activity 2: Repeat research activity 1 for $n = 30{,}000$ and $n = 40{,}000$. Do you think you can now achieve $n = 50{,}000$ and even $n = 100{,}000$? Try it to see if you can. If you were able to do so, why do you think you can now do it? If not yet, why do you think you still can not?

3. Using a graphing software for each n, graph t versus **Average** obtained from the Table above. Describe in detail what you have observed. Do you think you can go as far as $t = n$? If not, what about $t = n/2$? Or, $t = n/4$? Or, $t = n/8$?

Research Question 2: Did your average time improve compared to LRP01 and LRP02 for every t? Which t is the average time better? Statistically the same? Slower? Why?

Research Question 3: Discuss in your own words what will happen if instead you divide **X** in step (4) above, still into t submatrices, but in a manner where each submatrix is of size $n/t \times n$?

Solutions to Possible Problems that may Come up

Problem 1: *I do not know how to write core-affine threaded programs*.
Answer 1: *Is that really a problem?*

Problem 2: *The programming language I used for LRP01 and LRP02 does not have a robust library for writing core-affine threaded codes.*
Answer 2: *See Answer 1.*