CENTRAL WASHINGTON UNIVERSITY
CS 473 Parallel Computing
Instructor: Dr. Andonie

Final Project

JuneYeob Lee

June 4, 2020

Contents

1 Problem Description

You are given an array of n records, each containing the x and y coordinates of a house. You are also given the x and y coordinates of a railroad station. Design a parallel algorithm to find the house closest to the railroad station.

2 Design

I used MPI to solve the problems. Create n size of vector, each vector index has x and y values pair. Each vector's component represent the location of the house. Also, has one independent x and y pair for the railroad station.

2.1 Partitioning

Assume that having n houses and p processors. If n is divisible by p. Each processor will calculate n/p times.If n is not divisible by p, the beginning of r processors calculate (n/p) +1 times.So, each processor get task with cyclic distribution.

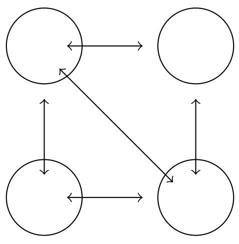
Array of house

1	2	3	1+p	2+p	3+p	...	n-1	n

$$\begin{aligned}
P(1) &= CalculateDistance\ to\ 1, 1+p, 1+2p, 1+3p...\\
P(2) &= CalculateDistance\ to, 2+p, 2+2p, 2+3p...\\
P(3) &= CalculateDistance\ to\ 3, 3+p, 3+2p, 3+3p...\\
&....\\
P(n-1) &= CalculateDistance\ to\ n-1, (n-1)+p, (n-1)+2p, (n-1)+3p...\\
P(n) &= CalculateDistance\ to\ n, n+p, n+2p, n+3p...
\end{aligned}$$

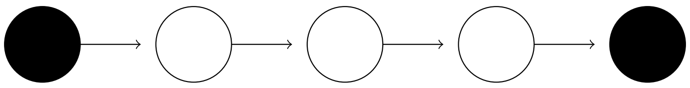
2.2 Communication

Each processor has minimum distance from the rail road to the houses in its part. To find Global minimum value, processor should communicate all the other processors and comparing the local minimum values. Meaning all processors have to send n-1 times and receive n-1 times to get global minimal value.



2.3 Agglomeration and Mapping

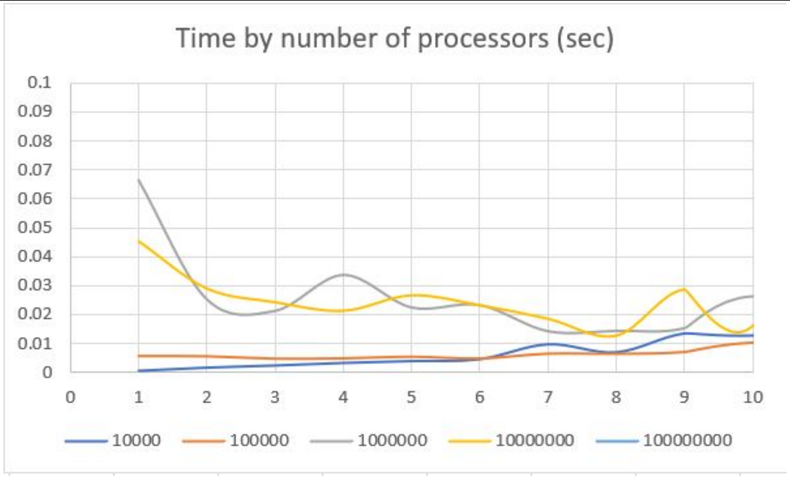
To reduce the amount of communication, change the strategy of communication. So, each Processor send and receive only once. Through this strategy, n-1 times of sending and receiving total to get the global minimum distance. Below Graph shows circular communication. So, the first processor will send local minimum value to the next processor. Then next processor receive the minimum value and compare own minimum then smaller value. Finally, The root processor will receive the Global minimum value. In below graph, black circle represents root processor.



3 Bench marking

Below chart represents the execution time(sec) by different number of houses and number of processors. x and y coordinator's range is between -50000 to 50000. Rail road position is set as (0,0).

Processors|Houses	10000	100000	1000000	10000000	100000000
1	0.000614	0.005852	0.066438	0.04543	0.48525
2	0.001705	0.005762	0.025226	0.0290441	0.281197
3	0.002457	0.004936	0.021206	0.024309	0.203713
4	0.003334	0.00505	0.03362	0.0213216	0.198085
5	0.003988	0.005604	0.02238	0.0266825	0.220511
6	0.004621	0.005016	0.023212	0.0232674	0.204014
7	0.009808	0.006652	0.014175	0.0185903	0.21473
8	0.007036	0.006617	0.014313	0.0127189	0.187104
9	0.013557	0.007234	0.015217	0.0286756	0.190679
10	0.01285	0.010456	0.026215	0.0161053	0.173274
20	0.022032	0.018786	0.025133	0.274093	
30	0.030464	0.026666	0.032115	3.49142	
40	0.036435	0.041174	0.061378	9.94482	



4 Efficiency analysis

According to the graph and chart, sequential algorithm is more efficient if the number of house is smaller. However, the number of houses are increasing, sequential programming will takes more time than parallel programming. Execution time decreases by number of processor but there is certain point that the time increase because of communication between processors. In conclusion, we should figure out the number of processors can perform the best. Time complexity for the sequential Algorithm O(N). Time complexity for the parallel algorithm O(n/p +c).

5 Discussion

How to figure out the best number of processors? (instead of running all the cases)
Is there a better way to communication in my case?
which value can affect more on execution time ? number of object we have to calculate or number of communication between processors.
Assuming that we have the best hardware to solve the problem, which programming language can perform the best among CUDA, MPI, and OpenMP?