Parallel implementation of Submatrix Search

Final project

Course 10324, Parallel and Distributed Computation

2022 Spring Semester

Problem Definition

Picture(N) and **Object**(N) – are square matrices of integers with N rows and N columns. Each member of the matrix represents a "color". The range of possible colors is [1, 100].

Position(I, J) defines a coordinates of the upper left corner of the Object into Picture.

For each pair of overlapping members **p** and **o** of the Picture and Object we will calculate a relative difference

$$diff = abs((\mathbf{p} - \mathbf{o})/\mathbf{p})$$

The total difference is defined as a sum of all relative differences for all overlapping members for given Position(I, J) of the Object into Picture. We will call it **Matching**(I, J).

For example, for the Picture and Object from the Fig.1 the matching at Position(0,0) is equal

Matching(0,0) = abs((10-5)/10) + abs((5-14)/5) + abs((67-9)/67) + abs((23-20)/23) + abs((6-56)/6) +

abs((5-2)/5) + abs((12-6)/12) + abs((10-10)/10) + abs((20-3)/20)

10	5	67	12	8	4
23	6	5	14	9	5
12	10	20	56	2	3
1	2	6	10	3	2
45	3	7	5	5	2
11	43	2	54	1	12

5	14	9
20	56	2
6	10	3

Fig 1. Picture(6) and Object(3)

10	5	67	12	8	4
23	6	5	14	9	5
12	10	20	56	2	3
1	2	6	10	3	2
45	3	7	5	5	2
11	43	2	54	1	12

Fig 2. Ideal Matching of the Object into the Picture at Position (1, 2)

This project deals with a "recognition" if there is a Position(I,J) of the Object into Picture with a Matchin(I,J) less than the given value.

Input data and Output Result of the project

The input file contains in the first line a Matching Value. Next line contains a number of Pictures in the file. For all pictures the first line defines its ID, next line contains the picture size, followed by picture elements. The same order is for objects as well.

Input.txt

Matching value

Number of Pictures

   #For each picture:

Picture ID

Picture dimension(N)

N^2 lines of members of the picture

…

Number of Objects

   #For each object:

Object ID

Object dimension(N)

N^2 lines of members of the object

Output.txt

The output file contains information about result for each Picture - if there was found an Object with a Matching value less than given one. It has to be in the following format:

Picture Id: found Object ID in Position (I,J)   # In case some object was found

Picture Id: No Objects were found   # In case NO objects were found

For example,

Picture 1 found Object 1 in Position(1,2)

Picture 2 found Object 11 in Position(11,20)

Picture 3 No Objects were found

Requirements

- Implement the Simplified "Recognition" algorithm explained above to find an Object into Picture with appropriate Matching value.
- Input file contains a number of Pictures and Objects. For each pair Picture/Object find if there exists an Appropriate Matching. The search for the given Picture is stopped when one of Object from the object set is found.
- The input file **input.txt** initially is known for one machine only. The results must be written to the file **output.txt** on the same machine
- The computation time of the parallel program must be faster than sequential solution.

- Be ready to demonstrate your solution running on VLAB (two computers from different pools

-

- if MPI is used)

- **No code sharing between students is allowed.** Each part of code, if any, which was incorporated to your project must be referenced according to the academic rules.

- Be able to explain each line of the project code, including those that was reused from any source.

- **The project that is not created properly (missing files, build or run errors) will not be accepted**

Grade Policy

- **60 points** for the effective **proper** parallel implementation of the problem with two components: *MPI+OpenMP*   or   *OpenMP+ CUDA*   or   *MPI+CUDA*. The project that produces wrong results will not be accepted.

- **10 points** for implementation of full *MPI+OpenMP+CUDA* configuration.

- **10 points** for the documentation of your solution – clear explanation what and how the problem was parallelized, what is a rational of choosing the specific architecture, complexity evaluation.

- **10 points** for the code quality – modularity, generality, self-explanatory, organization.

- **10 points** for the Load Balancing.

Additional Bonus for the project grade

5 points for implementation with OpenCL instead of CUDA

5 points for use of OpenMP task construct

5 points for use CUDA multistreaming

5 points for implementation a different sophisticated variation of the algorithm (must be approved by lecturer).

5 points for your own proposal (must be approved by lecturer).

הפרויקט יוגדר כמטלת הקורס. הגשת התוכנה והתיעוד רק דרך מערכת **Moodle.**

נדרשת הגנה על הפרויקט בפגישה עם המרצה לפי לוח הגשות שיתפרסם.

יישום והגשת הפרויקט <u>ביחידים בלבד.</u>

בהצלחה