CSE 625 Parallel Programming Term Project

November 8, 2022

100 points (30% of the final grade)

Due: December 7 (Wed) midnight (Submit your project report and any related source code to the Blackboard.)

1 (30 points)

In the CodeBlocks project, All\_Pair\_distance, it implements three functions using C++ multi-threads to compute the pair-wise distance matrix of MNIST train images (loaded from train-images.bin). These three methods are:

1 block\_all\_pairs (C++ multi-threads - block work distribution)

2 block\_ cyclic\_all\_pairs (C++ multi-threads - block cyclic work distribution)

3 dynamic\_all\_pairs (C++ multi-threads - dynamic work distribution)

Use OpenMP to re-implement these three methods.

1.1 Implement block work distribution in OpenMP and compare its computing time with   
 that of C++ multi-threads block\_all\_pairs implementation. List your OpenMP   
 implementation in the report and put the computing time results in the following   
 table.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Matrix Size | 400 | 800 | 10,000 | 20,000 | 30,000 | 60,000 |
| C++ Block 12 threads |  |  |  |  |  |  |
| OpenMP block  12 threads |  |  |  |  |  |  |

1.2 Implement block-cyclic work distribution in OpenMP and compare its computing   
 time with that of C++ multi-threads block\_cyclic\_all\_pairs implementation. List   
 your OpenMP implementation in the report and put the computing time results in the   
 following table.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Matrix Size | 400 | 800 | 10,000 | 20,000 | 30,000 | 60,000 |
| C++ block-cyclic 12 threads  Chunk size 2 |  |  |  |  |  |  |
| OpenMP block-cyclic  12 threads  Chunk size 2 |  |  |  |  |  |  |

1.3 Implement dynamic work distribution in OpenMP and compare its computing time   
 with that of C++ multi-threads dynamic\_all\_pairs implementation. List your   
 OpenMP implementation in the report and put the computing time results in the   
 following table.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Matrix Size | 400 | 800 | 10,000 | 20,000 | 30,000 | 60,000 |
| C++ dynamic 12 threads  Chunk size 2 |  |  |  |  |  |  |
| OpenMP dynamic  12 threads  Chunk size 2 |  |  |  |  |  |  |

2 (70 points) Individual term project

The topic of the term project must relate to solving computationally intensive problems   
using any parallel computing platforms such as C++ multi-threading, OpenMP, CUDA, python libraries (e.g, PyTorch), etc.

Select your term project from one of the follwig list of project topics:

1 **sp500 stock price Similarity Computation Efficency and Analysis**

The goal of this topic is a study of computation of similarity matrix of time series   
 (based on various similarity measurements) and their applications.

Code example:

sp500.zip (available on the Blackboard)

References:

[1] Measuring Financial Time Series Similarity With a View to Identifying Profitable  
 Stock Market Opportunities  
 (<https://arxiv.org/pdf/2107.03926.pdf>)

[2] A similarity measurement for time series and its application to the stock   
 market   
 (<https://www.sciencedirect.com/science/article/pii/S0957417421006503>)

[3] A Fuzzy Approach for Similarity Measurement in Time Series, Case Study for   
 Stocks

(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7274751/)

2 **Celeb Faces Dataset analysis**

The goal of this topic is a study of computational efficiency of large image dataset.

Code example:

celebA.zip (available on the Blackboard)

The example uses the Large-scale CelebFaces Attributes (CelebA) Dataset, which can be downloaded here: https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200K celebrity images, each with 40 attribute annotations. The images in this dataset cover large pose variations and background clutter. CelebA has large diversities, large quantities, and rich annotations, including

* 10,177 number of identities,
* 202,599 number of color (R, G, B) face images, and
* 5 landmark locations, 40 binary attributes annotations per image.

3 **MNIST written digit classifiers Implementation and Analysis (C++)**  
  
 3.1 OneNN classifier and

3.2 NN\_Softmax classifier

(Download link: https://louisville.box.com/s/9hpbooqc7n8xr5jk740byr2gaih94tv9)

The goal of this topic is a study of classifier computation efficiency ans analyze and compare the accuracy of the classfiers and the misclassified digit images.

4 **CUDA applications using CodeBlocks gcc compiler**

5 **PyTorch (CPU and/or GPU) applications of reasonably large-scale computing**   
 (e.g., computing all-pair distance matrix using various distance metric)

Project Report Outline

1 Title page

CSE 625 Term Project Report

<Project Title>

<Name>

mm-dd-yyyy

2 Project statement and objective  
  
3 General description of the approach (including required platforms)

4 Implementation details (showing important source code snippets in the report)

5 Contributions

6 References