

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики

Звіт

Лабораторна робота №9

Тема: «MPI»

з дисципліни "Паралельні та розподілені обчислення"

Виконав студент групи ПМі-31
Яцуляк Андрій

Львів 2023 р.

Мета: Паралельне множення матриць використовуючи MPI.

Теоретичний матеріал

Матриця - це прямокутний масив чи таблиця чисел, символів або виразів, упорядкованих у рядки та стовпці, які використовуються для представлення математичного об'єкта або властивості такого об'єкта.

Основні операції з матрицями: додавання матриць однакових розмірів, множення матриці на число, множення матриць, транспонування матриці.

Хід роботи

Завдання виконав мовою програмування C++ у середовищі Visual Studio 2022.

Написав клас Matrix, який використовував для множення (вся матриця ініціалізується одним масивом довжиною $n*n$). Програму потрібно викликати через консоль, вказуючи кількість потоків та розміри матриці. Якщо розміри матриці менші або рівні 10, то я виводжу кожну матрицю в консолі, також результат множення послідовно і паралельно.

```

#pragma once
#ifndef MATRIX_H_
#define MATRIX_H_

#include <vector>
#include <complex>

using std::vector;
using std::complex;

template <typename T>
class Matrix
{
private:
    size_t rows, cols;
    vector<T> elements;
public:
    Matrix(size_t numRows, size_t numCols);
    Matrix(size_t numRows, size_t numCols, T* data);
    size_t getRows();
    size_t getCols();
    T operator()(size_t row, size_t col) const;
    T& operator()(size_t row, size_t col);
    T* data();
    const vector<T>& getElements();
};

#endif

```

Спочатку я перевіряв правильність роботи на матрицях 5 на 5 (значення елементів матриць ініціалізується випадковими числами від 0 до 99):

Для початку, перевіряв звичайне множення матриці 4x4 на 4 потоках:

```
D:\Parallel computing\Lab9and10\x64\Debug>mpiexec -n 4 Lab9and10.exe 4
The matrices are: 4x4
Total mpi time = 0.0013035
Sequential multiplication time = 2.4e-06 seconds
Matrix A:
55 55 2 22
2 33 13 49
36 15 21 10
77 9 45 78
Matrix B:
36 84 51 62
72 87 23 39
58 77 68 21
36 94 2 5
Matrix C(mpi):
6848 11627 4250 5707
4966 8646 1843 1929
3954 6886 3629 3308
8838 18048 7350 6460
Matrix C(seq):
6848 11627 4250 5707
4966 8646 1843 1929
3954 6886 3629 3308
8838 18048 7350 6460
```

Тут на екран виводиться загальний час множення, використовуючи MPI, час послідовного множення. Отже, все працює правильно

Далі я експериментую з матрицями різних розмірностей для різних потоків, ось кілька прикладів:

```
D:\Parallel computing\Lab9and10\x64\Debug>mpiexec -n 4 Lab9and10.exe 200
The matrices are: 200x200
Total mpi time = 0.0738674
Sequential multiplication time = 0.188726 seconds

D:\Parallel computing\Lab9and10\x64\Debug>mpiexec -n 10 Lab9and10.exe 200
The matrices are: 200x200
Total mpi time = 0.039341
Sequential multiplication time = 0.189922 seconds

D:\Parallel computing\Lab9and10\x64\Debug>mpiexec -n 10 Lab9and10.exe 1000
The matrices are: 1000x1000
Total mpi time = 4.68296
Sequential multiplication time = 24.2035 seconds

D:\Parallel computing\Lab9and10\x64\Debug>mpiexec -n 4 Lab9and10.exe 1000
The matrices are: 1000x1000
Total mpi time = 8.97738
Sequential multiplication time = 23.9342 seconds

D:\Parallel computing\Lab9and10\x64\Debug>mpiexec -n 16 Lab9and10.exe 1000
The matrices are: 1000x1000
Total mpi time = 4.60655
Sequential multiplication time = 23.9801 seconds

D:\Parallel computing\Lab9and10\x64\Debug>mpiexec -n 32 Lab9and10.exe 1000
The matrices are: 1000x1000
Total mpi time = 4.43237
Sequential multiplication time = 23.9145 seconds
```

Отже, множення матриць за допомогою MPI дає помітне прискорення, проте час все ще довший, якщо порівнювати з CUDA.

Висновок. Під час виконання лабораторної роботи я написав програму для паралельного множення матриць використовуючи MPI, протестував роботу на матрицях різної розмірності та для різної кількості потоків.