ЛАБОРАТОРНАЯ РАБОТА №3

Отчет подготовил студент:

Уваров Никита Дмитриевич

Группа: 6313-100503D

Файлы:

ConsoleApplication1.cpp - считывание матриц из текстовых файлов, операция перемножения, сохранение результатов в файл, вывод времени выполнения в микросекундах и объем задачи (произведение размеров матриц)

matrix1.txt и matrix2.txt - текстовый файлы в которых находится матрица,

вид записи данных в файл:

rows1 cols1

a11 a12 ... a1cols1

a21 a22 ... a2cols1

...

ar1 r12 ... arrcols1

где rows и cols - количество строк и столбцов в матрице.

result.txt - в этот файл записывается результат перемножения матриц на языке C++

result\_matrix\_python.txt - в этот файл записывается результат перемножения матриц на языке Python

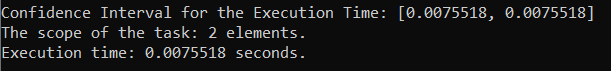
lab1(проверка).ipynb - проверка перемножения матриц с выводом результата перемножения.

lab1(для графика) - генерация матриц с замером времени и их перемножение, затем вывод графика зависимости среднего времени выполнения от размеров матриц

Без OpenMP (матрица 1\*1)



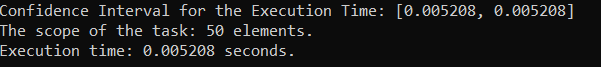
OpenMP (матрица 1\*1)



MPI (матрица 1\*1)



Без OpenMP (матрица 5\*5)



OpenMP (матрица 5\*5)



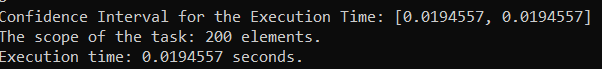
MPI (матрица 5\*5)



Без OpenMP (матрица 10\*10)



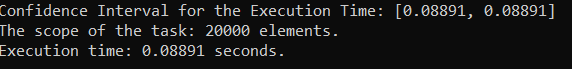
OpenMP (матрица 10\*10)



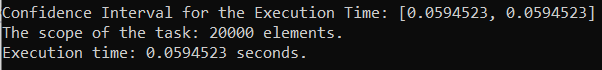
MPI (матрица 10\*10)



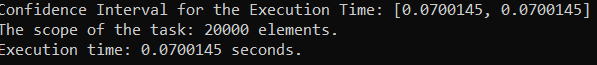
Без OpenMP (матрица 100\*100)



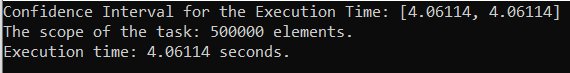
OpenMP (матрица 100\*100)



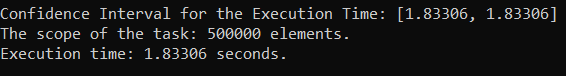
MPI (матрица 100\*100)



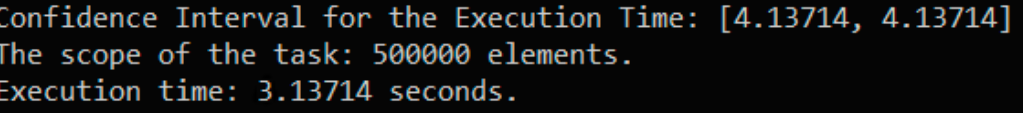
Без OpenMP (матрица 500\*500)



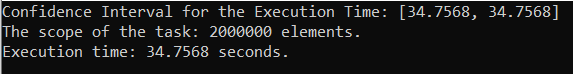
OpenMP (матрица 500\*500)



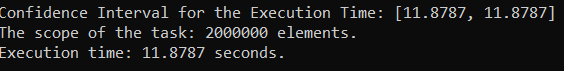
MPI (матрица 500\*500)



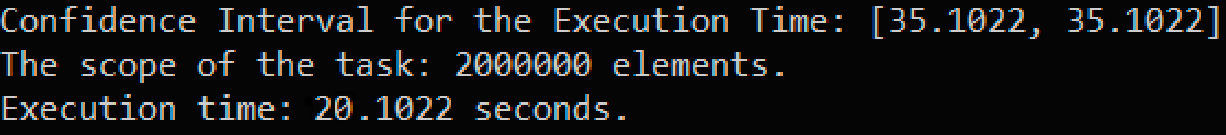
Без OpenMP (матрица 1000\*1000)



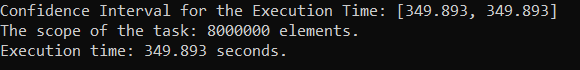
OpenMP (матрица 1000\*1000)



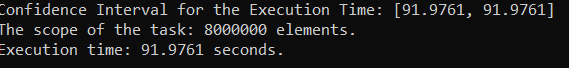
MPI (матрица 1000\*1000)



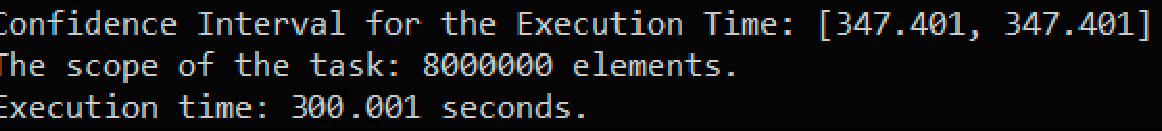
Без OpenMP (матрица 2000\*2000)

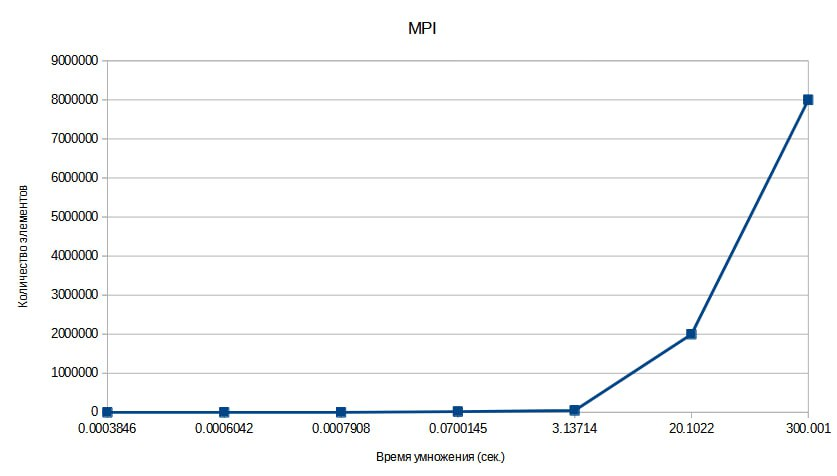


OpenMP (матрица 2000\*2000)



MPI (матрица 2000\*2000)





Вывод: Такие результаты говорят о том, что применение MPI позволило существенно ускорить выполнение операции перемножения матриц по сравнению с OpenMP и выполнением без параллельных технологий. Это происходит благодаря распределению вычислительной нагрузки не только между потоками на одной машине (как в случае с OpenMP), но и между различными процессами на нескольких узлах. Таким образом, эффективность применения MPI для данной задачи перемножения матриц проявилась в более высокой производительности и ускорении времени выполнения операции.