ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С. П. КОРОЛЕВА»

Отчет по лабораторной работе №2

Модификация программы из л/р №1 для параллельной работы по технологии OpenMP.

Выполнили:

Мантров И.А.

гр. 6313-10.05.03D

Принял:

Минаев Е.Ю.

Самара 2025

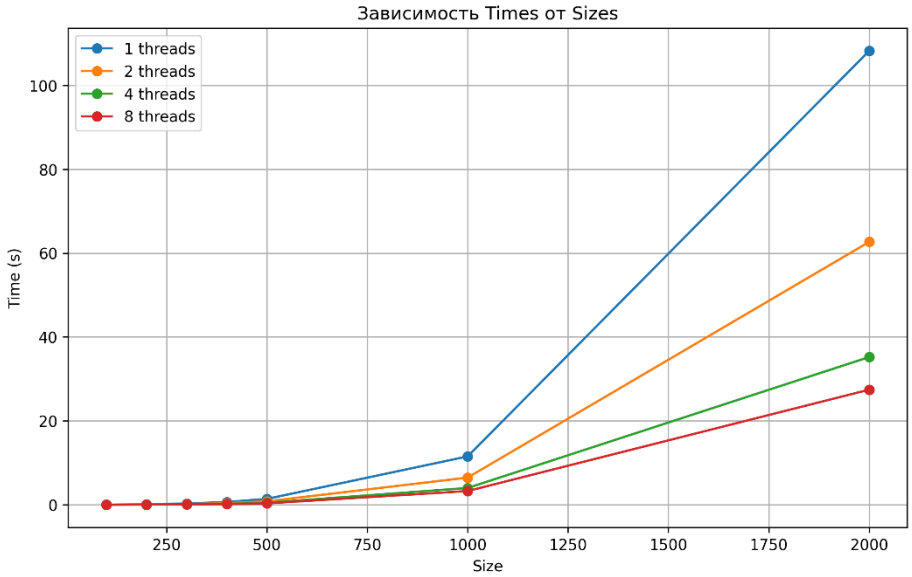
Структура лабораторной работы:

- `matrix.h, matrix.cc` – класс матриц `Matrix`, в котором реализован метод перемножения матриц (перегрузка оператора *)
- `random_generator.h, random_generator.cc` – класс `RandomGenerator`, в котором реализована генерация матриц фиксированного размера с случайными элементами
- `stat.h, stat.cc` – класс ExecutionTimer для замера времени выполнения.
- `openmp.cc` - основная программа для выполнения параллельного матричного умножения с использованием OpenMP.
- `Statistic2.py` – модуль для верификации перемножения матриц с помощью вспомогательного средства numpy и для построения статистического графика.

В файле result_for_lab2 расположены исходные матрицы и результаты их перемножения, а также в файле statistic.txt находится статистика времени перемножения матриц.

В файле compare_result2.txt выведены результаты проверки правильности перемножения (верификация результатов вычислений производилась с помощью модуля numpy, реализация находится в файле statistic2.py)

В файле stat2.png вывод статистики в виде графика.



Вывод: на основании анализа графика производительности (stat2.png) и данных из файла statistic.txt можно сделать вывод, что применение технологии OpenMP для параллельного перемножения матриц значительно сократило время выполнения операции по сравнению с последовательной версией из лабораторной работы №1. График демонстрирует, что с увеличением количества потоков время вычислений уменьшается, особенно заметно для матриц больших размеров. Это подтверждает эффективность распараллеливания алгоритма с помощью OpenMP.

Результаты верификации, приведенные в файле compare_result2.txt, показывают полное соответствие с эталонными значениями, полученными через модуль numpy. Это свидетельствует о корректности реализации параллельного алгоритма без потери точности вычислений. Таким образом, модификация программы с использованием OpenMP успешно достигла поставленной цели — ускорения операции матричного умножения при сохранении точности результатов.