Задание 4

Умножение матриц.

Минь Синьжань

1. Постановка задачи

С помощью интринсик используемой целевой архитектуры (AVX для Intel, AMD; NEON для ARM) реализовать векторизованную версию матричного умножения (A\*B = C). Можно предполагать, что матрицы A, B - квадратные. Обязательное требование - хранение всех матриц предполагается в едином порядке (все в row-major либо все в col-major). Тип элементов матрицы - float для 128-битных, double для 256-битных векторных расширений. Сравнить результаты и время выполнения векторизованного алгоритма с его последовательной версией. (N = 512, 1024, 2048).

2. Формат командной строки

Для Apple M3: g++ -O3 -std=c++11 -o file\_name vector\_neon.cpp

Для Intel i7: g++ -O2 -mavx -mfma vector\_avx.cpp -o file\_name

3. Спецификация системы

Процессор: Apple M3/Intel CORE i7-1065G7 CPU@1.30GHz

вычислительных ядер: 8/8

4. Результаты выполнения

Для Apple M3 я использовала NEON, для Intel i7 я использовала AVX. Для каждого значения N, для Sequential\_time, Vectorized\_time и Max\_error я выполнил три запуска и взял среднее значение, которое затем занёс в таблицу. И вот результаты:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Процессор | N\_size | Sequential\_time | Vectorized\_time | Max\_error |
| Apple M3 | 512 | 0.134544 | 0.0265929 | 0 |
| Intel i7 | 512 | 0.177067 | 0.0934039 | 0 |
| Apple M3 | 1024 | 0.957385 | 0.234705 | 0 |
| Intel i7 | 1024 | 1.38977 | 0.715686 | 0 |
| Apple M3 | 2048 | 16.006 | 4.00603 | 0 |
| Intel i7 | 2048 | 40.3847 | 19.7624 | 0 |

**Apple M3 процессор:**

Использование набора инструкций NEON значительно повысило производительность при всех значениях N\_size, особенно при больших N\_size (таких как 1024 и 2048), где прирост производительности еще более заметен. Для N\_size 512 и 1024 время векторизованного выполнения значительно ниже, чем последовательного, что показывает высокую эффективность NEON на Apple M3.

**Intel i7 процессор:**

Использование набора инструкций AVX также повысило производительность, но не так значительно, как на Apple M3. Особенно при меньших N\_size (таких как 512 и 1024), время векторизованного выполнения все еще значительно выше, чем результаты Apple M3. При N\_size 2048 улучшение производительности заметно, но все же остается значительное отставание по сравнению с Apple M3.

**Точность:**

Максимальная ошибка (Max\_error) во всех тестах равна 0, что свидетельствует о том, что точность векторных вычислений на обоих процессорах стабильна и одинакова в данных тестах.