

Задание 1

Анализ влияния кэша на операцию матричного умножения

Отчёт

Савельев К.М.

2022

1. Постановка задачи

Реализовать последовательный алгоритм матричного умножения и оценить влияние кэша на время выполнения программы. В задании рассматривались квадратные целочисленные матрицы одинакового размера N . Элемент произведения матриц A и B определяется как:

$$c_{ij} = \sum_{k=1}^N a_{ik} \cdot b_{kj}, \quad i, j \in 1, 2, \dots, N. \quad (1)$$

В программе реализованы шесть режимов итерирования, соответствующих всевозможным перестановкам порядка суммирования по индексам i, j и k в формуле 1. Необходимо построить графики зависимости времени выполнения программы от порядка индексов суммирования для трёх размеров матриц – 300×300 , 500×500 и 1000×1000 .

2. Формат командной строки

На вход программе подаются бинарные файлы, содержащие размерности матрицы N (`int32_t`) и двумерные массивы, содержащие элементы матриц (`int32_t`).

`./out <имя м-цы A> <имя м-цы B> <имя м-цы C> <режим>`

Наличие файла с матрицей C необязательно, при его отсутствии он будет создан автоматически. Элементы матрицы C имеют тип `int64_t`. Номера режимов соответствуют следующему порядку индексов суммирования:

- | | |
|------------|------------|
| 1. ijk ; | 4. jik ; |
| 2. ikj ; | 5. jki ; |
| 3. kij ; | 6. kji . |

3. Спецификация системы

Процессор: Apple M1 Pro CPU @ 2.06GHz

Число вычислительных ядер: 8

4. Результаты выполнения

Для каждого режима было проведено 5 экспериментов. На рисунках 1 и 2 представлены графики зависимости времени выполнения от режима суммирования для разных размерностей матриц.

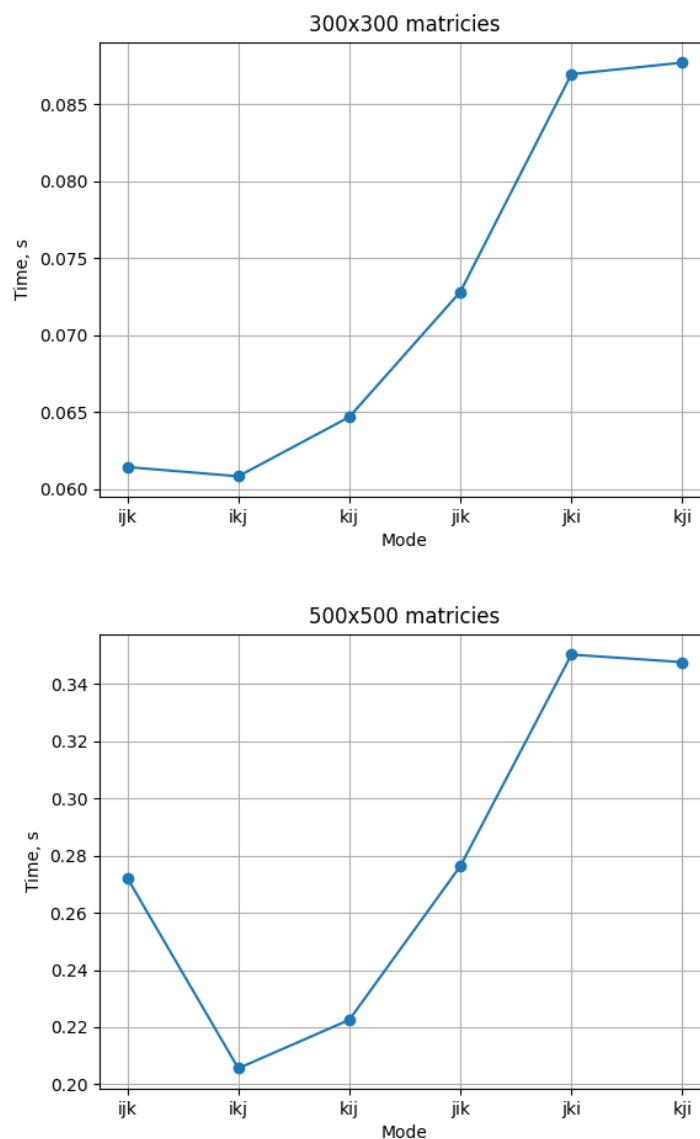


Рисунок 1 — График зависимости времени выполнения от режима суммирования для матриц 300×300 и 500×500 соответственно.

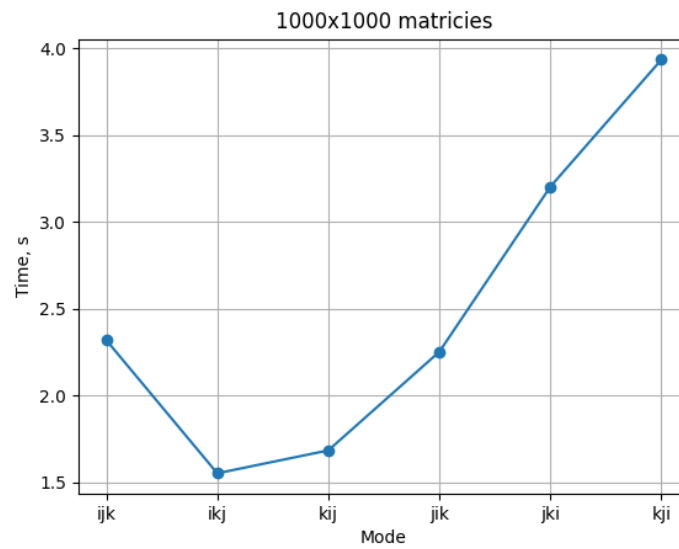


Рисунок 2 — График зависимости времени выполнения от режима суммирования для матрицы 1000×1000 .