



**Спецкурс: системы и средства параллельного
программирования.**

Отчёт № 1.

**Анализ влияния кэша на операцию матричного
умножения.**

Работу выполнил
Плужников И.Р.

Постановка задачи и формат данных.

Задача: Реализовать последовательный алгоритм матричного умножения и оценить влияние кэша на время выполнения программы.

Формат командной строки: <имя файла матрицы A > <имя файла матрицы B > <имя файла ответа> <порядок индексов>.

Формат файла-матрицы: Матрица представляются в виде бинарного файла следующего формата:

Тип	Значение	Описание
Символ типа char	T – f (float) или d (double)	Тип элементов
Число типа int	N – натуральное число	Число строк и столбцов матрицы
Массив чисел типа T	$N \times N$ элементов	Массив элементов матрицы

Элементы матрицы хранятся построчно.

Описание алгоритма.

Математическая постановка: Алгоритм матричного умножения ($A * B = C$) можно

$$c_{ij} = \sum_k (a_{ik} \cdot b_{kj}),$$

представить в следующем виде: для каждого элемента матрицы C .

Оценка влияния кэша на время выполнения программы осуществляется за счёт перестановки индексов суммирования.

Анализ времени выполнения: Для оценки времени выполнения программы использовалась функция: clock(). Для повышения надёжности экспериментов опыты проводились несколько раз (3).

Верификация: Для проверки корректности работы программы использовались тестовые данные.

Основные функции:

Разбор командной строки. В рамках функции осуществляется анализ и разбор командной строки.

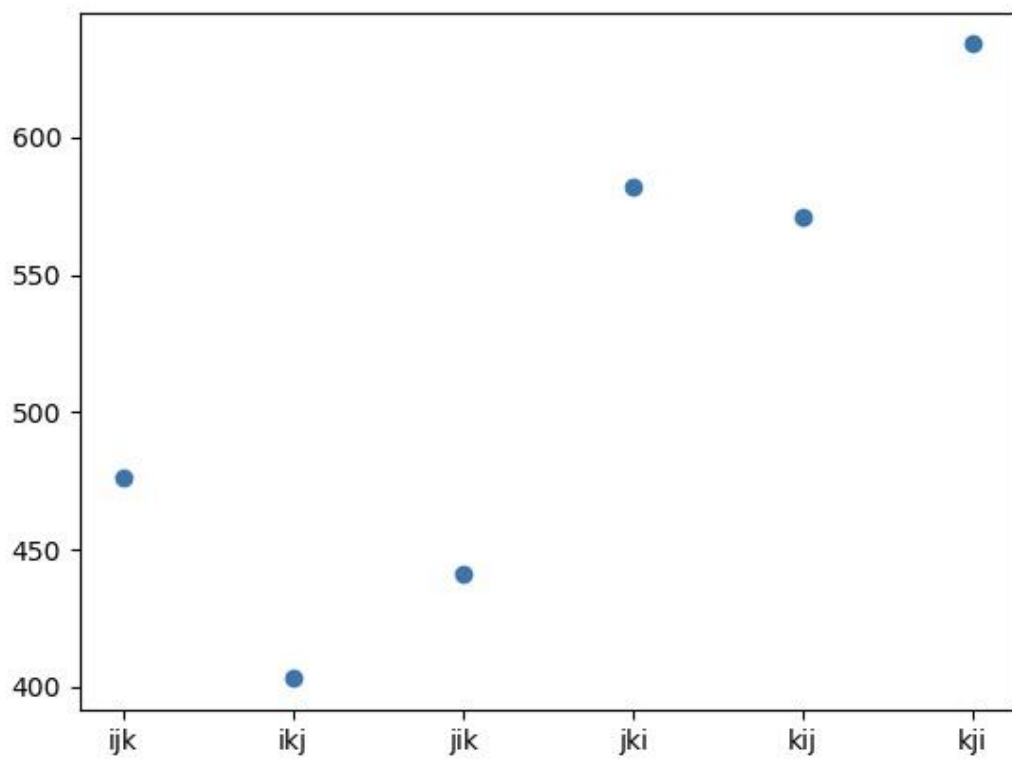
Чтение файлов матриц. В рамках функции осуществляется чтение входных матриц.

Перемножение матриц. В рамках функции осуществляется перемножение матриц в соответствие с выбранным порядком индексов суммирования.

Результаты выполнения.

Результаты:

Проводилось перемножение двух матриц размерами 500 x500 и 500 x500. Зависимость времени выполнения от порядка индексов суммирования представлена на графике (время в миллисекундах).



Основные выводы.

Исследования показывают, что изменения порядка индексов суммирования оказывает влияние на время выполнения программы. Наименьшее время выполнения при следующем порядке индексов - ikj . При таком порядке доступ к элементам обеих входных матриц осуществляется последовательно. Наихудшее время при порядке kji . При таком подходе доступ к памяти осуществляется максимально непоследовательно.