

Московский Государственный Университет им. М.В. Ломоносова
Факультет Вычислительной Математики и Кибернетики
_____Кафедра Суперкомпьютеров и Квантовой Информатики_____



**Спецкурс: системы и средства параллельного
программирования**

Отчёт № 1

**Анализ влияния кэша на операцию матричного
умножения**

Работу выполнила
Домрачева Д. А.

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 матрицы C> <режим, порядок индексов>.

Режимы:

0 – ijk , 1 – ikj , 2 – kij , 3 – jik , 4 – jki , 5 – kji .

Формат файла с матрицей:

Матрица представляются в виде бинарного файла следующего формата:

Тип	Значение	Описание
Число типа <i>char</i>	$T = f(float)$ или $d(double)$	Тип элементов
Число типа <i>size_t</i>	N – натуральное число	Число строк матрицы
Число типа <i>size_t</i>	M – натуральное число	Число столбцов матрицы
Массив чисел типа T	$N \times M$ элементов	Массив элементов матрицы

Элементы матрицы хранятся построчно.

Описание алгоритма

Математическая постановка:

Алгоритм матричного умножения ($A \times B = C$) можно представить в следующем виде:

$$c_{ij} = \sum_k (a_{ik} \cdot b_{kj}) \quad \text{для каждого элемента матрицы } C.$$

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

Анализ времени выполнения:

Для оценки времени выполнения программы использовалась функция *clock()*.

Верификация:

Для проверки корректности работы программы использовались тестовые данные.

Основные функции:

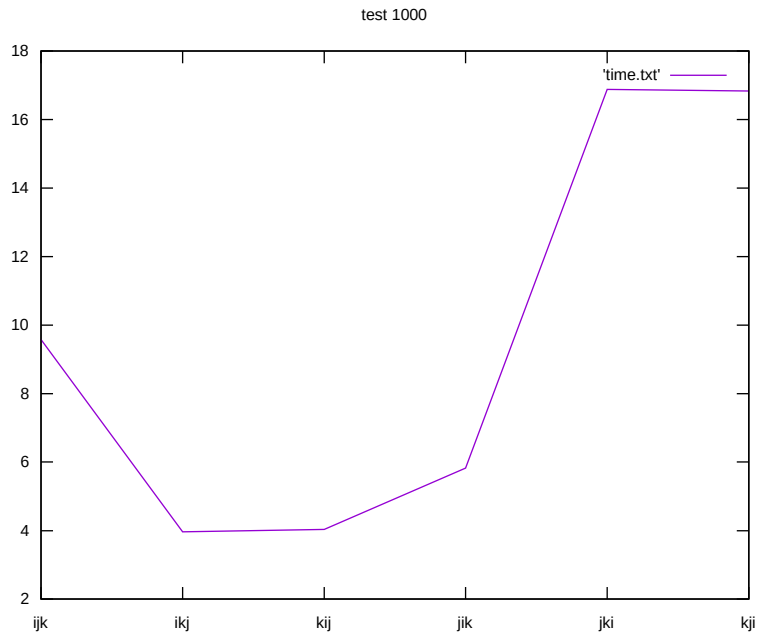
- Чтение матриц из бинарных файлов *read_matrix*. Шаблонная функция, считывающая из файла матрицы в формате float или double. Аналогичная функция для вывода матриц в файл – *write_matrix*.
- Основная функция, осуществляющая определение типов перемножаемых матриц и результата умножения – *get_time*. Вызывает дополнительную функцию, реализующую умножение. Возвращает время выполнения умножения функцией *multiply* в секундах.
- Шаблонная функция, реализовывающая умножение матриц в зависимости от типа данных и порядка индексов – *multiply*. Возвращает количество тактов, за время которых выполнялись операции.

Результаты выполнения

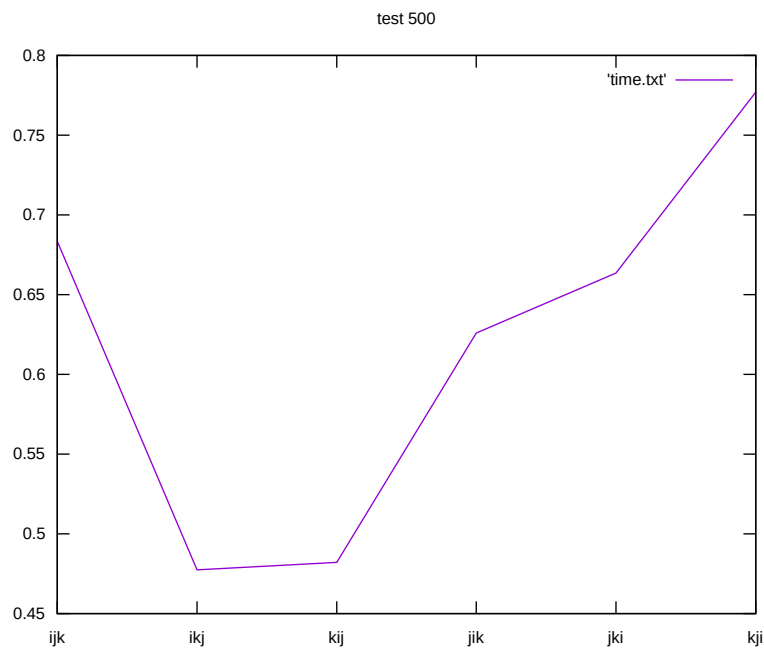
Результаты:

Проводилось перемножение матриц размерами 300×300 и 300×300 , 500×500 и 500×500 , 1000×1000 и 1000×1000 . Зависимость времени выполнения от порядка индексов суммирования представлена на графике (время в секундах).

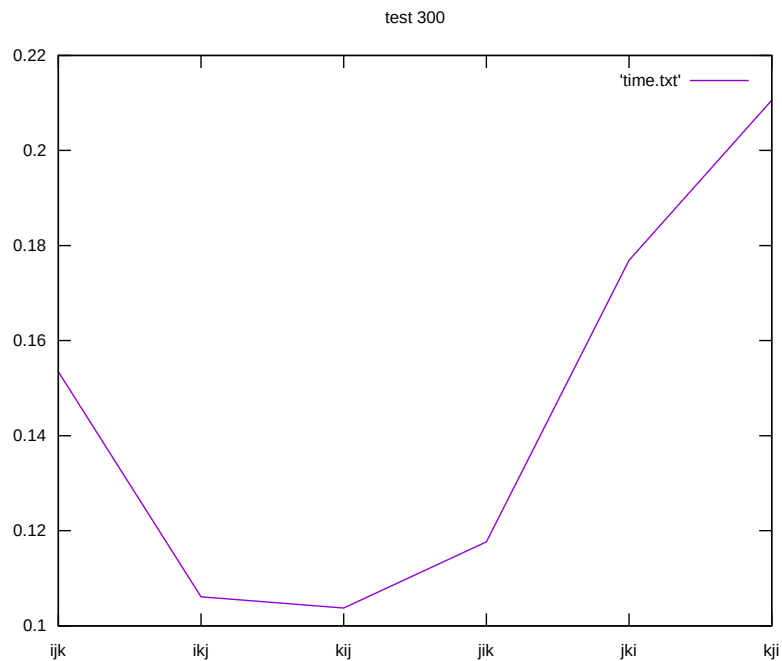
1000×1000 и 1000×1000 :



500×500 и 500×500 :



300×300 и 300×300:



Основные выводы

Исследования показывают, что изменения порядка индексов суммирования оказывает влияние на время выполнения программы. Наименьшее время выполнения при следующих порядках индексов – ikj и kij . При таких порядках доступ к элементам обеих входных матриц осуществляется последовательно. Наихудшее время при порядках jki и kji . При таком подходе доступ к памяти осуществляется максимально непоследовательно.