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 выполнил
Чепурнов А. В.

Москва 2018

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

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

Тип	Значение	Описание
Число типа char	T – f (float) или d (double)	Тип элементов
Число типа int	N – натуральное число	Число строк матрицы
Число типа int	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: $c_{ij} = \sum_k (a_{ik} \cdot b_{kj})$ 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 (10).

Верификация: для проверки корректности работы программы использовались тестовые данные.

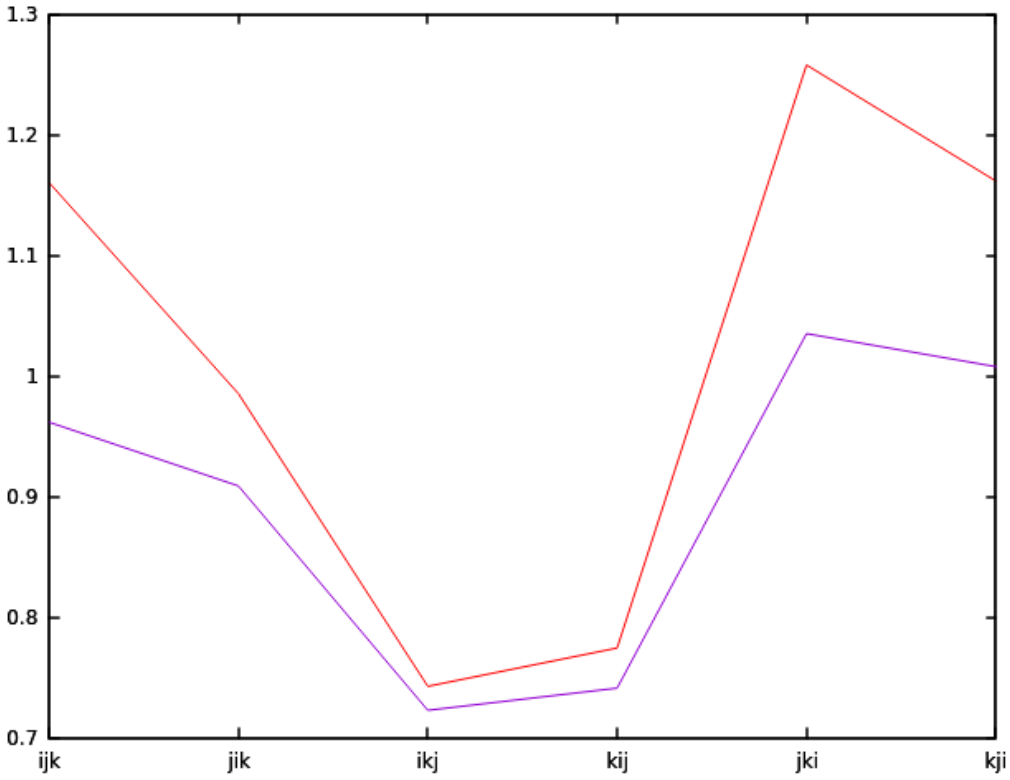
Основные функции:

- **Разбор командной строки.** В рамках функции осуществляется анализ и разбор командной строки.
- **Чтение файлов матриц.** В рамках функции осуществляется анализ совместимости входных матриц и их чтение.
- **Перемножение матриц.** В рамках функции осуществляется перемножение матриц в соответствие с выбранным порядком индексов суммирования.

Результаты выполнения

Результаты:

Проводилось перемножение двух матриц размерами $500x500$ с данными типа float и с данными типа double. Зависимость времени выполнения от порядка индексов суммирования представлена на графике (время в секундах).



Синий – float, красный – double

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 как для данных типа float, так и для данных типа double. Наихудшее время при порядке jki. При таком подходе доступ к памяти осуществляется максимально непоследовательно.