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 – ikj, 2 – jik, 3 – jki, 4 – kij, 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: $c_{ij} = \sum_k \left(a_{ik} \cdot b_{kj}\right)$ 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:

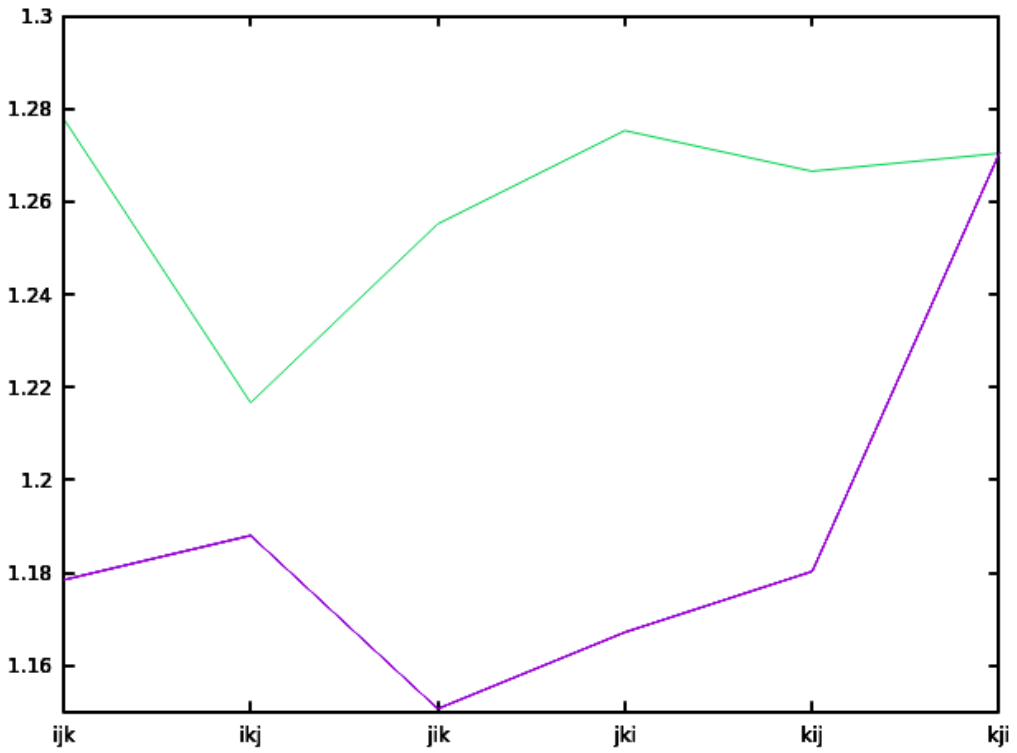
- **Разбор командной строки.** В рамках функции осуществляется анализ и разбор командной строки.
- **Чтение файлов матриц.** В рамках функции осуществляется анализ совместимости входных матриц и их чтение.
- **Перемножение матриц.** В рамках функции осуществляется перемножение матриц в соответствие с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

Результаты выполнения.

Результаты:

Проводилось перемножение двух матриц размерами $300x300$ с данными типа float и перемножение матриц размерами $500x100$ и $100x500$ с данными типа double.

Зависимость времени выполнения от порядка индексов суммирования представлена на графике (время в секундах).



Зелёный – тип float, синий – тип double

Основные выводы.

Исследования показывают, что изменения порядка индексов суммирование оказывает влияние на время выполнения программы. Для перемножения матриц с данными типа float наименьшее время выполнения при следующем порядке индексов – ikj, для матриц с данными типа double – jik. При таких порядках доступ к элементам обеих входных матриц осуществляется последовательно в каждом случае соответственно. Наихудшее время при порядке ijk для матриц float и kji для матриц double. При таком подходе доступ к памяти осуществляется максимально непоследовательно.