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
**Федоров В. В.**

Москва 2020

# 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-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 size_t | N – натуральное число | Число строк матрицы |
| Число типа size_t | 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 (500).
**Верификация:** Для проверки корректности работы программы использовались тестовые данные.
**Основные функции:**

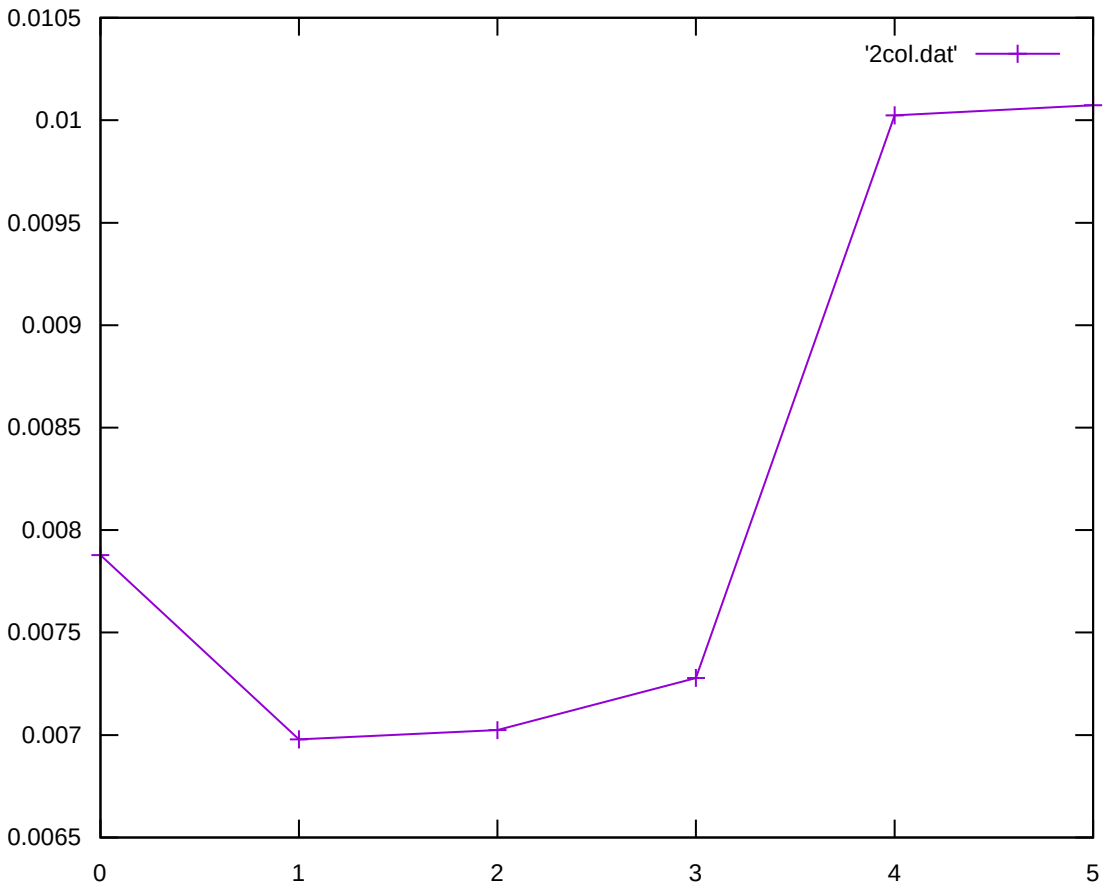
- `int main(...);`
  - В рамках функции осуществляется анализ и разбор командной строки, а также чтение из входных файлов информации о типе данных и размерах матриц, на основе которых анализируется их совместимость, после чего запускается функция `handlr`.
- `int handlr(...);`
  - В рамках функции осуществляется выделение памяти для матриц, вызов функций `getm`, вызов необходимой функции умножения матриц и запись результата в файл.
- `int getm(...);`
  - В рамках функции осуществляется считывание матрицы из файла.
- `int mult_***(...);` где *** - перестановка из i, j, k
  - В рамках функций осуществляется перемножение матриц с соответствующим порядком индексов.

# Результаты выполнения.

**Результаты:**

Проводилось перемножение двух квадратных матриц размерами $127 \times 127$ . Зависимость времени выполнения от порядка индексов суммирования представлена на графике (время в секундах).

# Основные выводы.




Исследования показывают, что изменения порядка индексов суммирование оказывает влияние на время выполнения программы. Наименьшее время выполнения при следующих порядках индексов — ikj (1) и kij (2). При таких порядках доступ к элементам обеих входных матриц осуществляется последовательно, а значит, кэш используется максимально эффективно. Наихудшее время - при порядках jki (4) и kji (5). При таких подходах доступ к памяти осуществляется максимально непоследовательно.