



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

Отчёт № 2.

**Анализ зависимости производительности алгоритма
блочного матричного перемножения от выбора порядка
индексов и размера блока**

Работу выполнил
Васильев С.М. гр. 323

Постановка задачи и формат данных.

Задача: Реализовать последовательный алгоритм блочного матричного умножения и измерить следующие величины: время выполнения, количество тактов, промахи кэша L1, промахи кэша L2, TLB, FLOPs

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

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	NxM элементов	Массив элементов матрицы

Описание алгоритма.

Математическая постановка: Алгоритм матричного умножения ($C = A \times B$) можно представить в следующем виде: $c_{ij} = \sum_k (a_{ik} \cdot b_{kj})$ для каждого элемента матрицы C .

Матрицы A , B , C можно разбить на блоки и выполнять умножение последовательно по блокам.

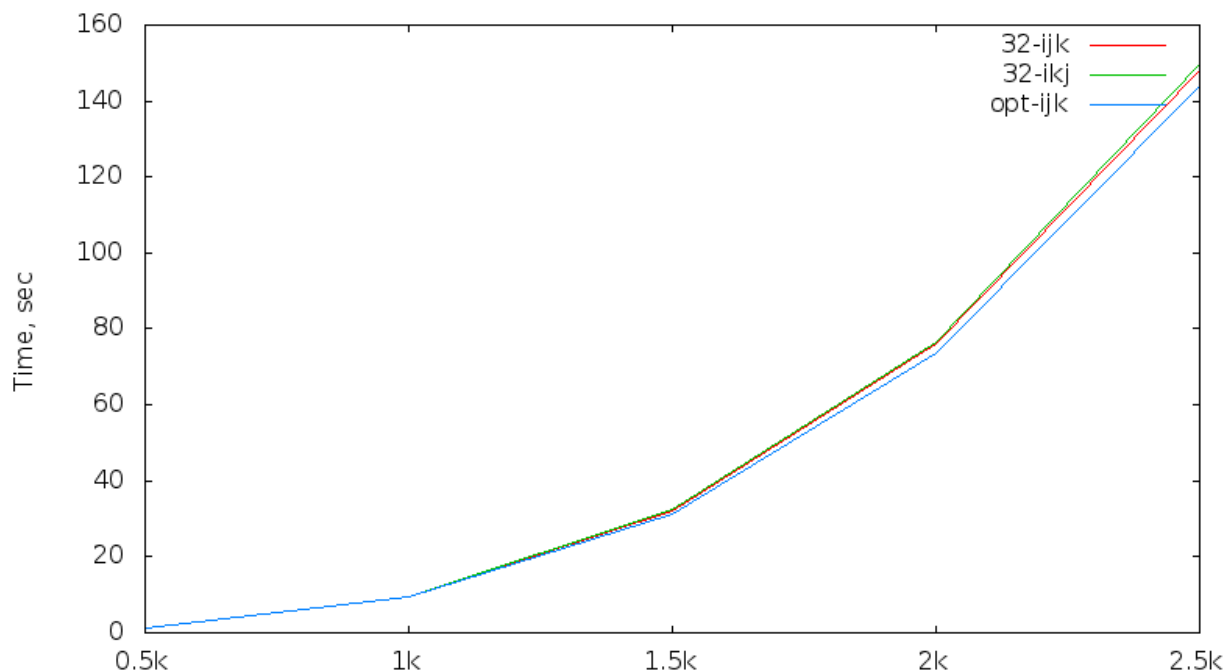
Анализ искомых величин: Для оценки времени выполнения программы использовалась функция: `clock()`, остальные величины вычислялись средствами PAPI.

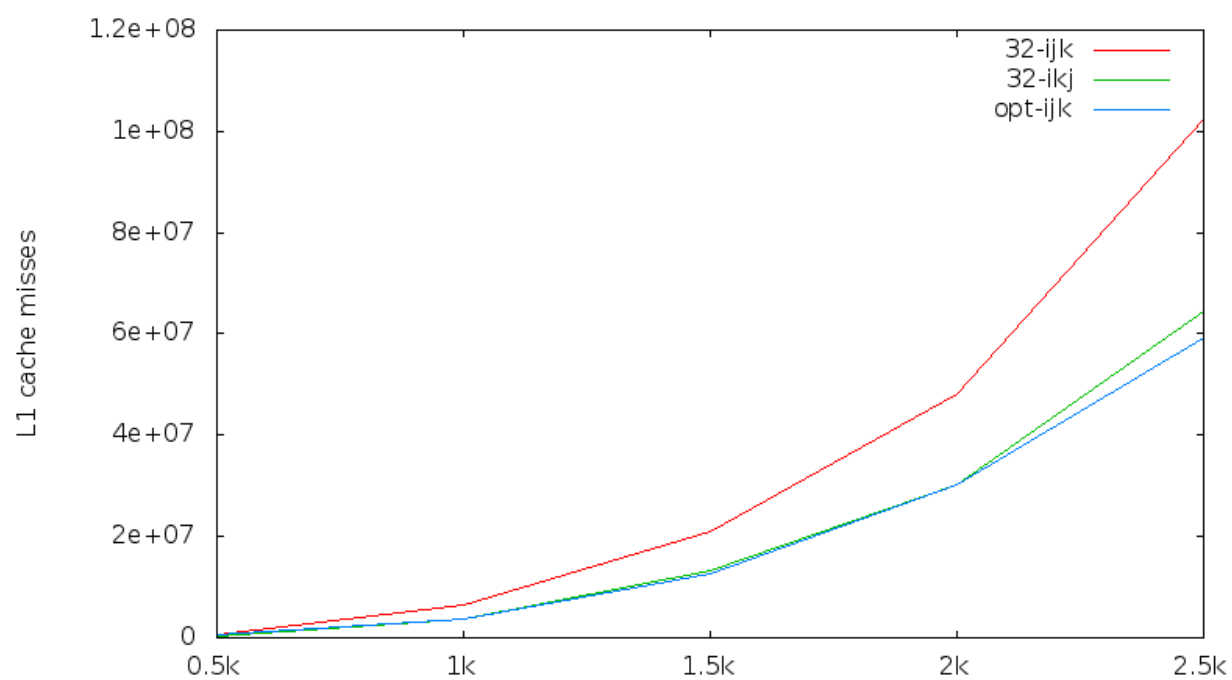
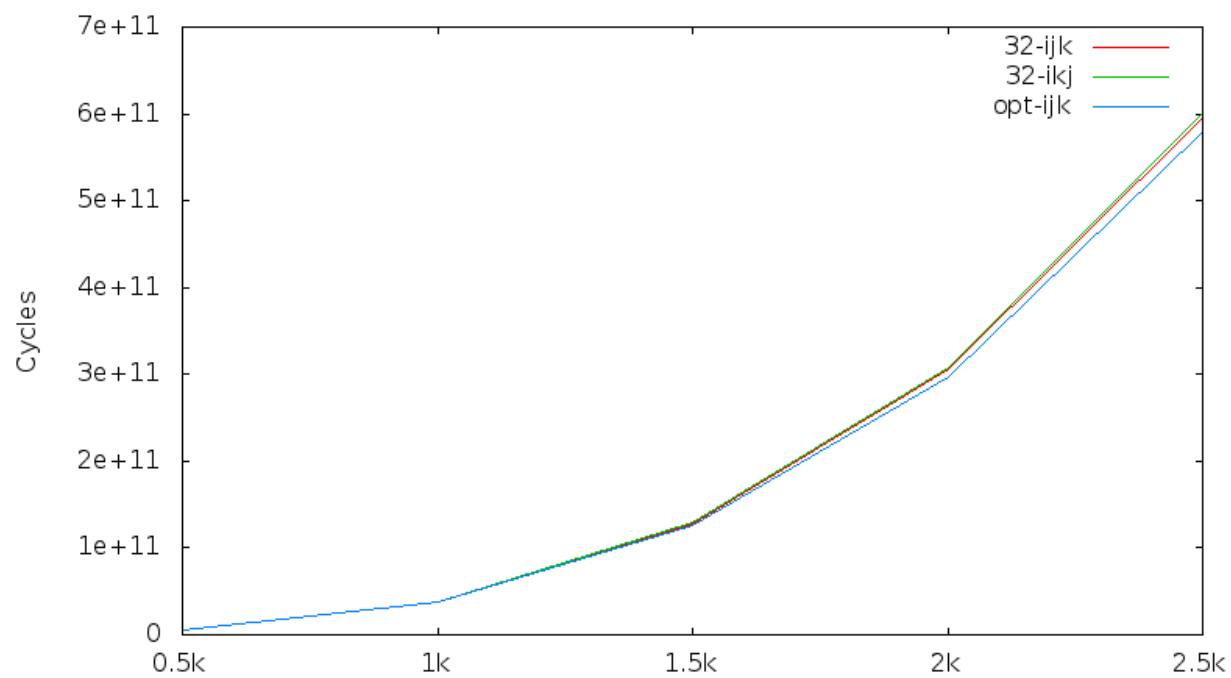
Верификация: Для проверки корректности работы программы использовалась функция `test()`, сравнивающая результат алгоритма блочного перемножения со стандартным алгоритмом перемножения матриц, реализованным и проверенным ранее.

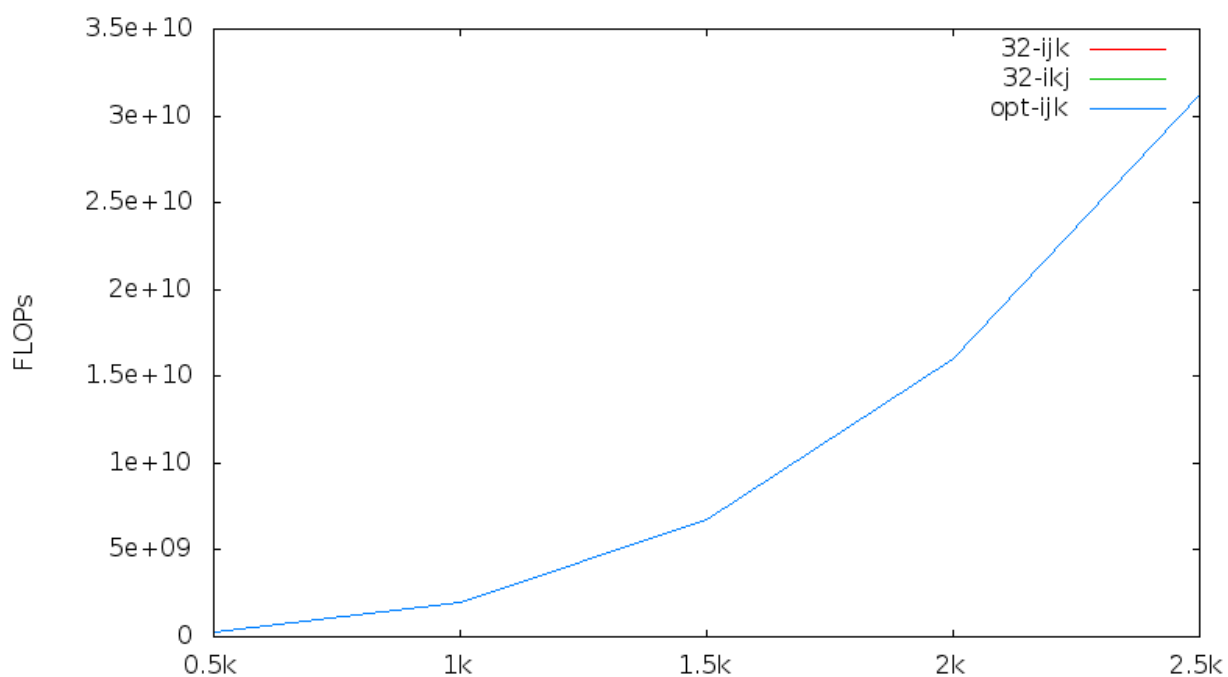
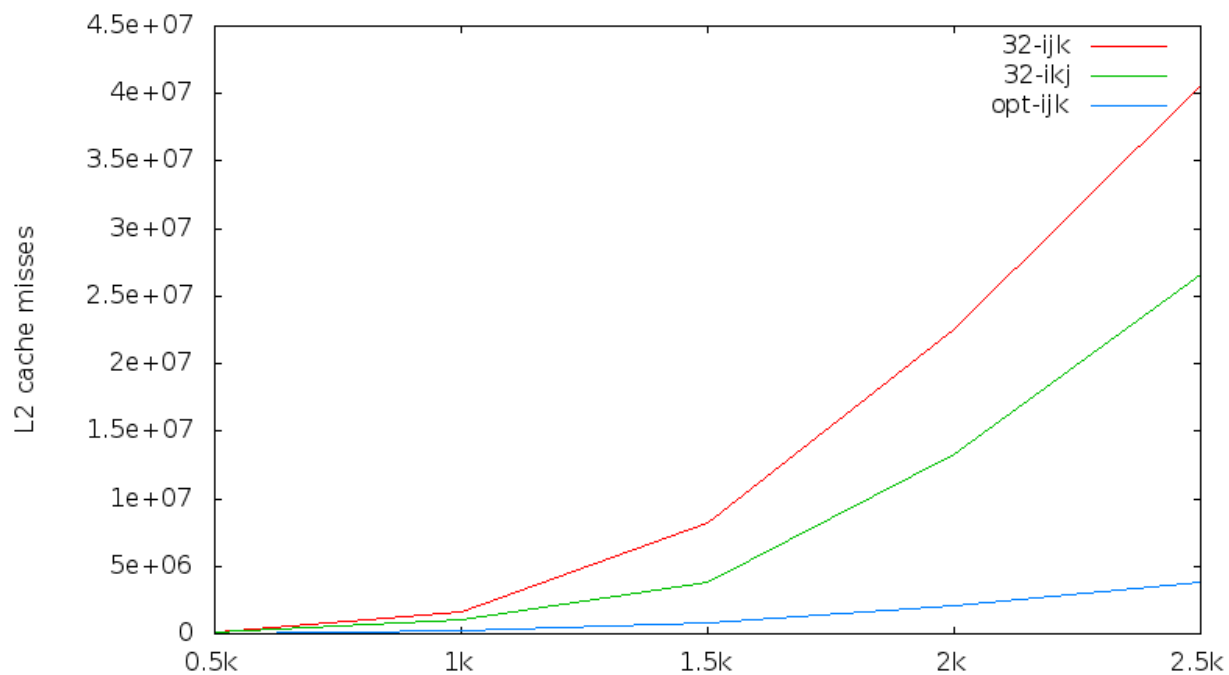
Примечание: Система не поддерживает сбор данных о величинах: TLB.

Результаты выполнения.

Проводилось перемножение двух матриц размерами 500 x 500, 1000 x 1000, 1500 x 1500, 2000 x 2000, 2500 x 2500. Зависимость искомых величин от порядка индексов и размера блоков представлена на графиках.







Основные выводы.

Исследования показывают, что изменения порядка индексов суммирования и размера блоков оказывает влияние исследуемые величины. Алгоритм показывает наилучшую производительность при порядке индексов ikj и оптимальном размере блоков (70).