



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

**Отчёт № 1.**

**Анализ влияния кэша на операцию матричного  
умножения.**

Работу выполнил  
**Ухин С.А.**

## Постановка задачи.

**Задача:** Реализовать последовательный алгоритм блочного матричного умножения и оценить влияние кэша на время выполнения программы. Дополнить отчёт результатами сбора информации с аппаратных счётчиков, используя систему RAPI

## Описание алгоритма.

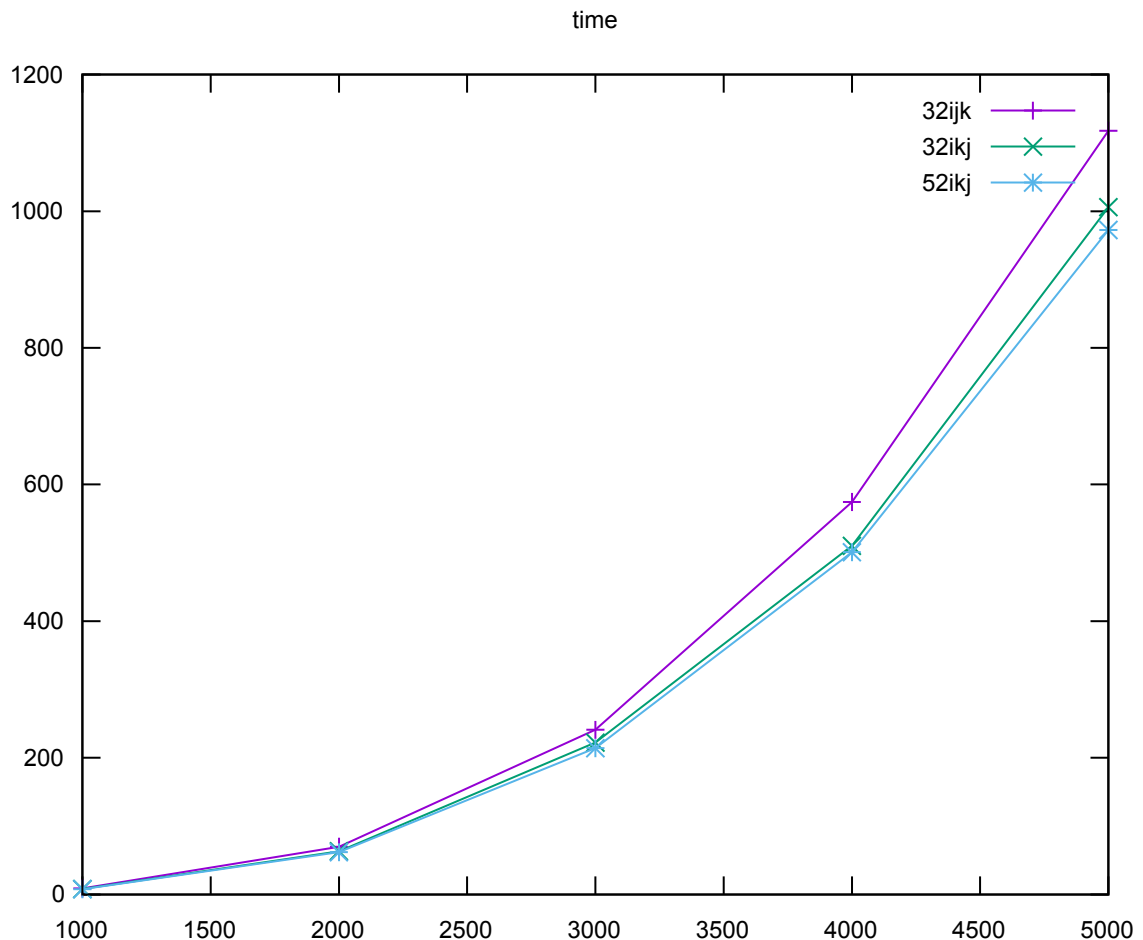
Матрицы делятся на маленькие блоки и происходит блочное перемножение матриц. При этом размер блока подбирается так, чтобы все данные, нужные для вычисления блока матрицы C поместились в кэш.

## Результаты выполнения.

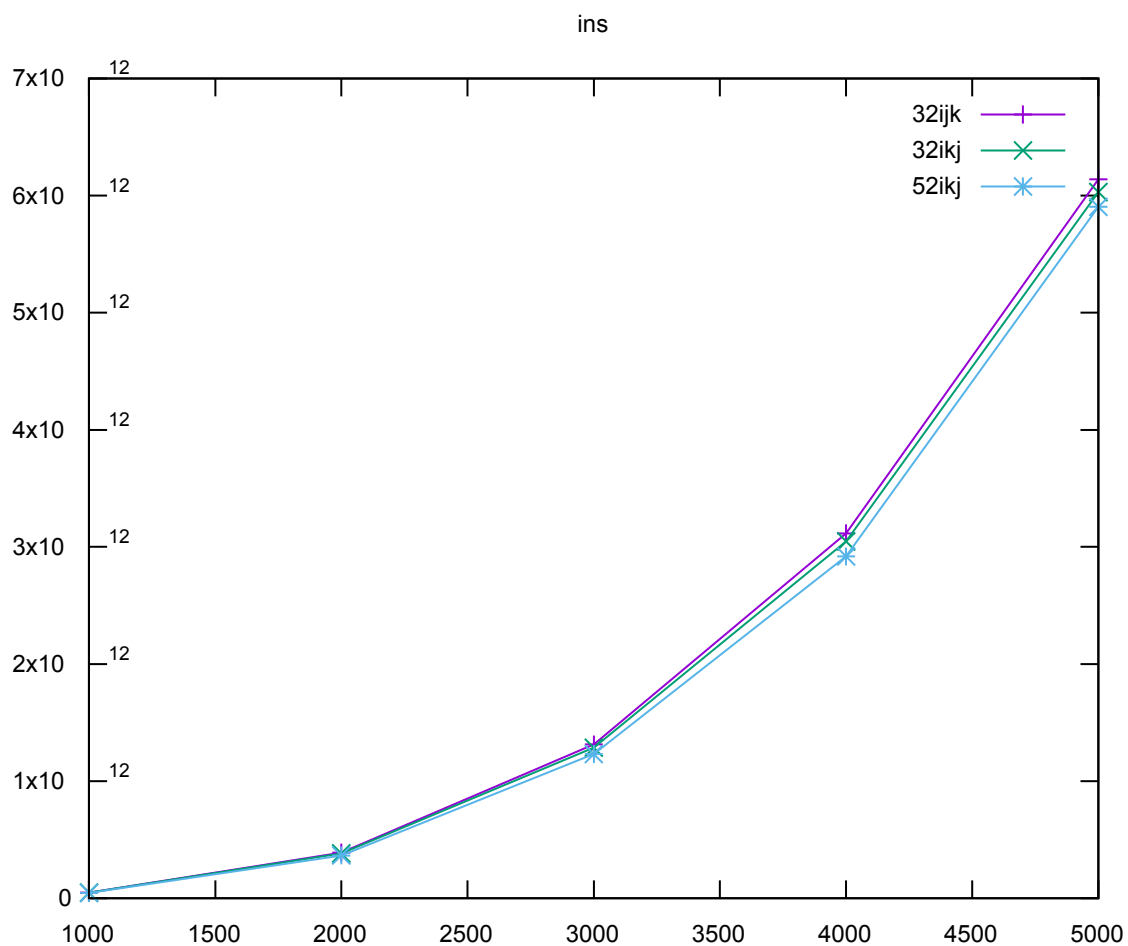
Перемножение выполнялось для квадратных матриц размером  $n = 1000, 2000, 3000, 4000, 5000$ . Тип данных float. Тремя разными способами : 0) blocksize = 32 индексы ijk;  
1) blocksize = 32 индексы ikj;  
2) blocksize =  $\sqrt{65536 / 12} = 72$  индексы ikj;  
Tlb misses не удалось посчитать на Polus. Подсчет выполнялся с помощью Rapi.

**Результаты:**

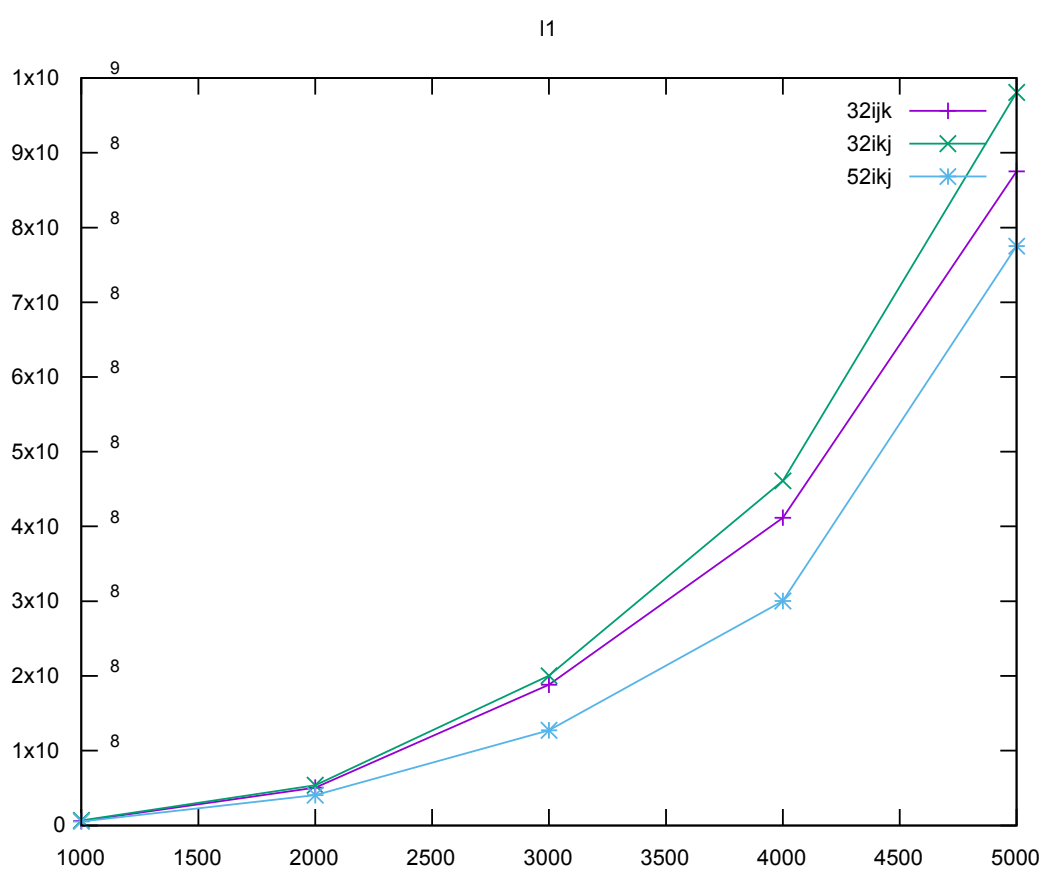
Время выполнения:



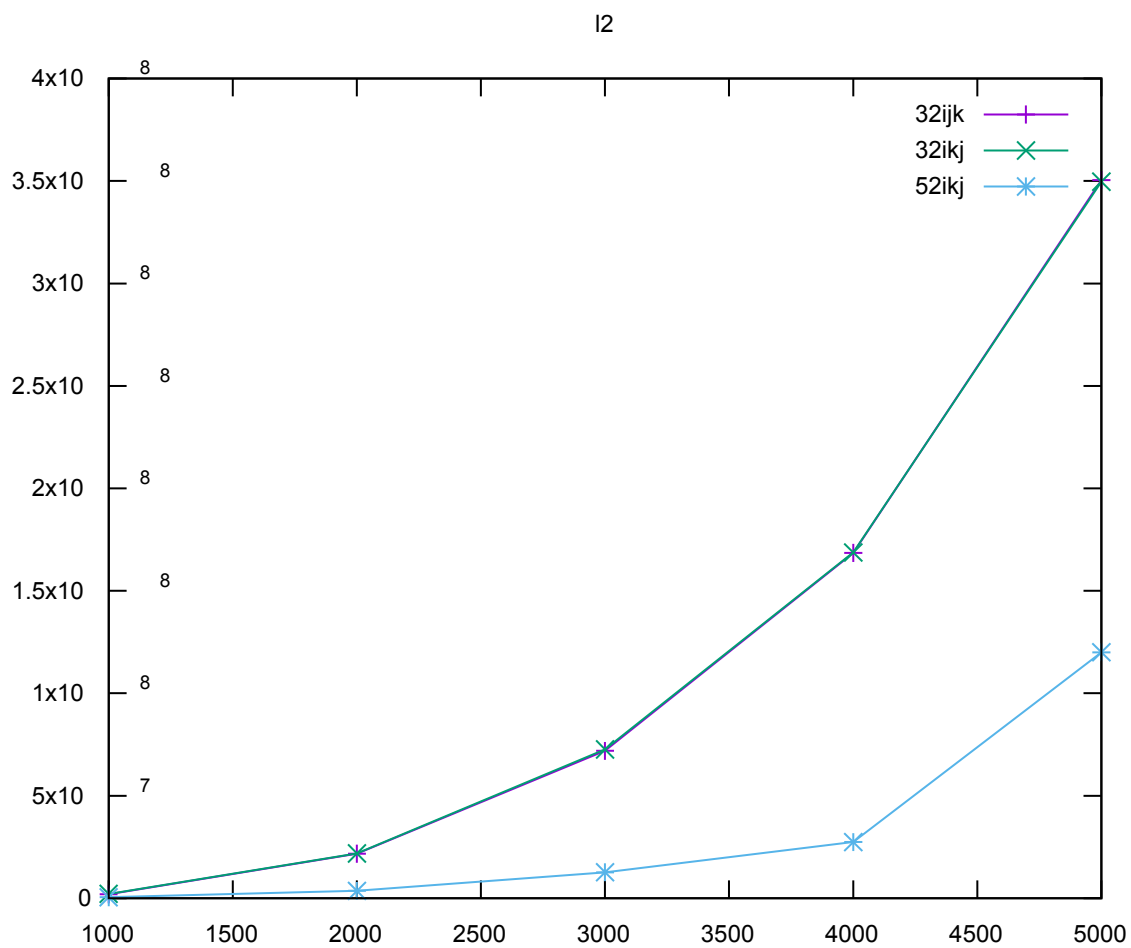
TOT\_INS:



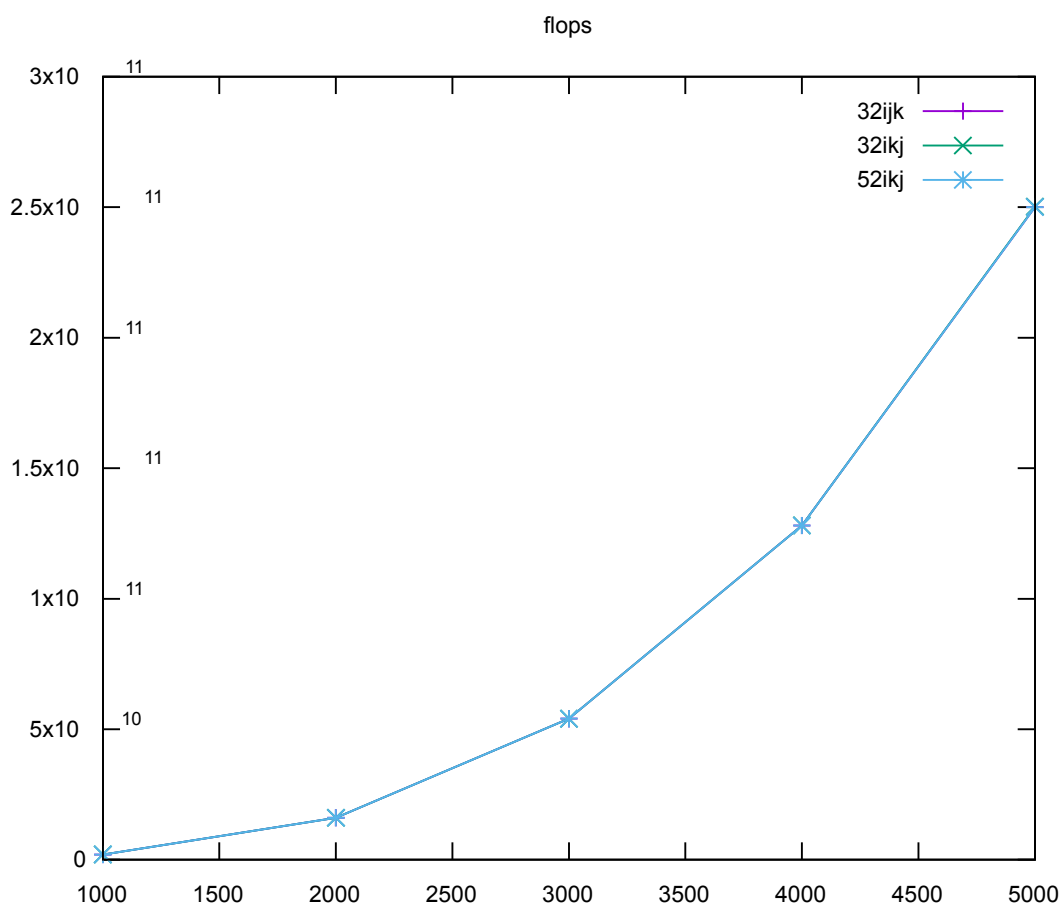
L1\_DCM:



L2\_DCM:



FLOPS:



### **Выводы:**

Время выполнения программы зависит от попаданий в кэш. Выбор оптимального размера блока приводит к уменьшению времени выполнения программы.