

## Подсчет количества cache miss для операции матричного умножения в зависимости от порядка итерирования

**Постановка задачи:** При помощи RAPI снять значения аппаратных счетчиков промахов L1/L2 кэшей при выполнении операции умножения квадратных матриц. Сравнить полученные значения с теоретическими для каждого порядка итерирования.

### Результат работы программы:

Запуск производился на системе Polus, поэтому из-за недоступности некоторых счетчиков не удалось посчитать промахи L2 кэша

Размеры матрицы: 1000x1000

Тип элементов: long long

	L1 load	L1 store	L1 cache	L2 load	L2 store	L2 cache
ijk	1866906648	24908	1867011711	invalid	invalid	invalid
ikj	55298881	7528	55356919	invalid	invalid	invalid
kij	54999211	12044	55069227	invalid	invalid	invalid
jik	1867661088	375853	1868096835	invalid	invalid	invalid
jki	3651192009	18327	3651480784	invalid	invalid	invalid
kji	3623288397	20884	3623487697	invalid	invalid	invalid

Размеры матрицы: 1000x1000

Тип элементов: int

	L1 load	L1 store	L1 cache	L2 load	L2 store	L2 cache
ijk	1848298296	2348	1848434078	invalid	invalid	invalid
ikj	27302563	7528	27360268	invalid	invalid	invalid
kij	29063826	5090	29098831	invalid	invalid	invalid

<b>jik</b>	1840623435	31288	1840686921	invalid	invalid	invalid
<b>jki</b>	3572028539	2454	3572114139	invalid	invalid	invalid
<b>kji</b>	3585975429	4271	3586068855	invalid	invalid	invalid

### Сравнение с теоретическими значениями (из лекций)

- ijk (и jik)
  - 2 loads, 0 stores
  - 1,25 cache misses/iteration
- ikj (и kij)
  - 2 loads, 1 store
  - 0,5 cache misses/iteration
- jki (и kji)
  - 2 loads, 1 store
  - 2,0 cache misses/iteration

	<b>L1 cache (int64_t)</b>	<b>Theor</b>	$\frac{Theor}{L1\ cache}$	<b>L1 cache (int32_t)</b>	<b>Theor</b>	$\frac{Theor}{L1\ cache}$
<b>ijk</b>	1867011711	1250000000	0,66951909	1848434078	1250000000	0,67624808
<b>ikj</b>	55356919	500000000	9,03229458	27360268	500000000	18,2746747
<b>kij</b>	55069227	500000000	9,07948099	29098831	500000000	17,1828208
<b>jik</b>	1868096835	1250000000	0,66913019	1840686921	1250000000	0,67909430
<b>jki</b>	3651480784	2000000000	0,54772299	3572114139	2000000000	0,55989252
<b>kji</b>	3623487697	2000000000	0,55195440	3586068855	2000000000	0,55771377

**Вывод:** Были сняты значения счётчиков RAPI, с помощью которых можно было определить значения промахов кэша при перемножении больших матриц. Время выполнения программы зависит от попадания в кэш, поэтому из полученных результатов можно сделать вывод о том, что перемножение матриц с порядком итераций ikj и kij наиболее эффективно.