

## Отчет по задачам problem1 – problem2

Выполнение заданий выполнялось на сервере polus.cmc.msu.ru, так как операционная система на моём личном ноутбуке не поддерживает инструмент PAPI.

Размер L1-data кэша = 64K, L1-instruction кэш = 32K, L2 кэш = 512K.

Ширина кэша = 64B, соответственно в строке кэша может храниться 16 и 8 слов соответственно для типов элементов матриц int32\_t и int64\_t.

Рассмотрим 6 видов матричного умножения (ijk, jik, ikj, kij, jki, kji) и посчитаем промахи кэша:

Для ijk и jik:

////////	Матрица A	Матрица B	Матрица C
int32_t	0.0625	1	0
int64_t	0.125	1	0

Для ikj и kij:

////////	Матрица A	Матрица B	Матрица C
int32_t	0	0.0625	0.0625
int64_t	0	0.125	0.125

Для jki и kji:

////////	Матрица A	Матрица B	Матрица C
int32_t	1	0	1
int64_t	1	0	1

Остюда получаем теоретические значения счетчиков:

	int32_t	int64_t
ijk	1.0625	1.125
jik	1.0625	1.125
ikj	0.125	0.25
jki	2	2
kij	0.125	0.25
kji	2	2

На системе Полюс не доступны счетчики уровня L2, поэтому далее буду работать только с L1

Далее тестирование для всех видов матричного умножения;  
Размеры матриц 1000x1000

Для int32\_t:

	L1 Load	L1 Store	L1 Cache	Theor	Theor/L1
ijk	1592959853	1805956	1594765809	1062500000	0,666242023752843
jik	1611600428	629356	1612229784	1062500000	0,659025165360672
ikj	16891833	134	16891967	125000000	7,3999670967863
jki	3239331549	36303295	3275634844	2000000000	0,610568666914571
kij	19134605	15812	19150417	125000000	6,52727300925092
kji	3166972235	35941623	3202913858	2000000000	0,624431404861092

Для int64\_t:

	L1 Load	L1 Store	L1 Cache	Theor	Theor/L1
ijk	1771640086	1101129	1772741215	1125000000	0,634610393486
jik	1776134467	14736105	1790870572	1125000000	0,628186099871878
ikj	58544633	3547	58548180	250000000	4,26998755554827
jki	3394889769	4578752	3399468521	2000000000	0,588327259877574
kij	49426698	49426698	49753561	250000000	5,02476596599789
kji	3326828526	4634576	3331463102	2000000000	0,600336830625357