

# Lab 2

## Характеристики

### ### CPU parameters

```
Architecture:          x86_64

CPU(s):                80

Model name:            Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz

Thread(s) per core:    2

Core(s) per socket:    20

Socket(s):             2

CPU max MHz:           3900.0000
```

### ### Server name

```
name:                  ProLiant XL270d Gen10
```

### ### NUMA parameters

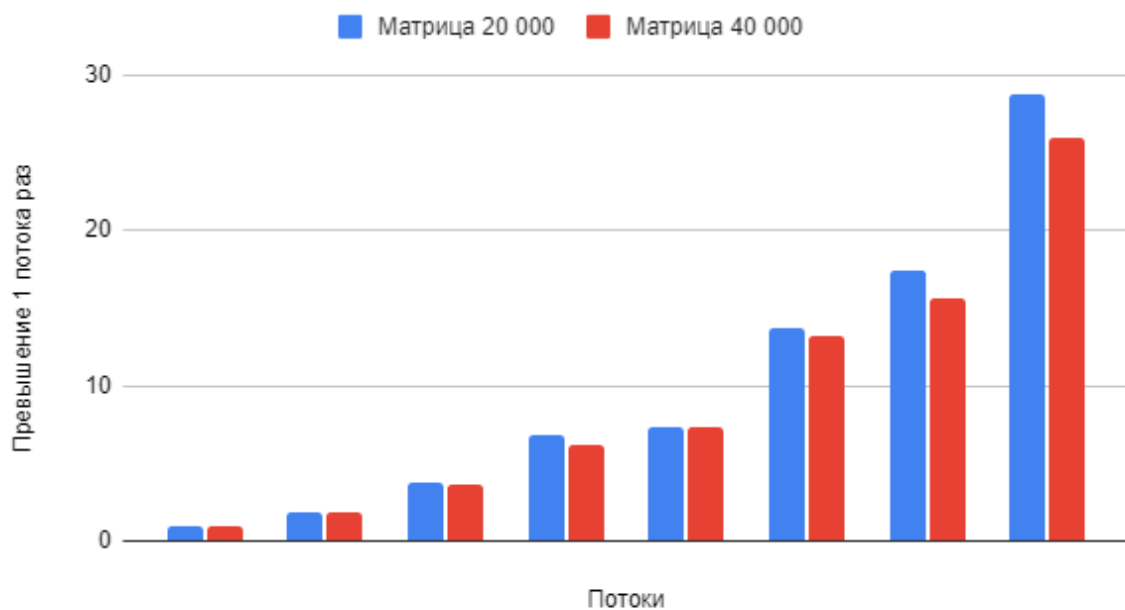
```
available:             2 nodes (0-1)

node size:              385636 MB
```

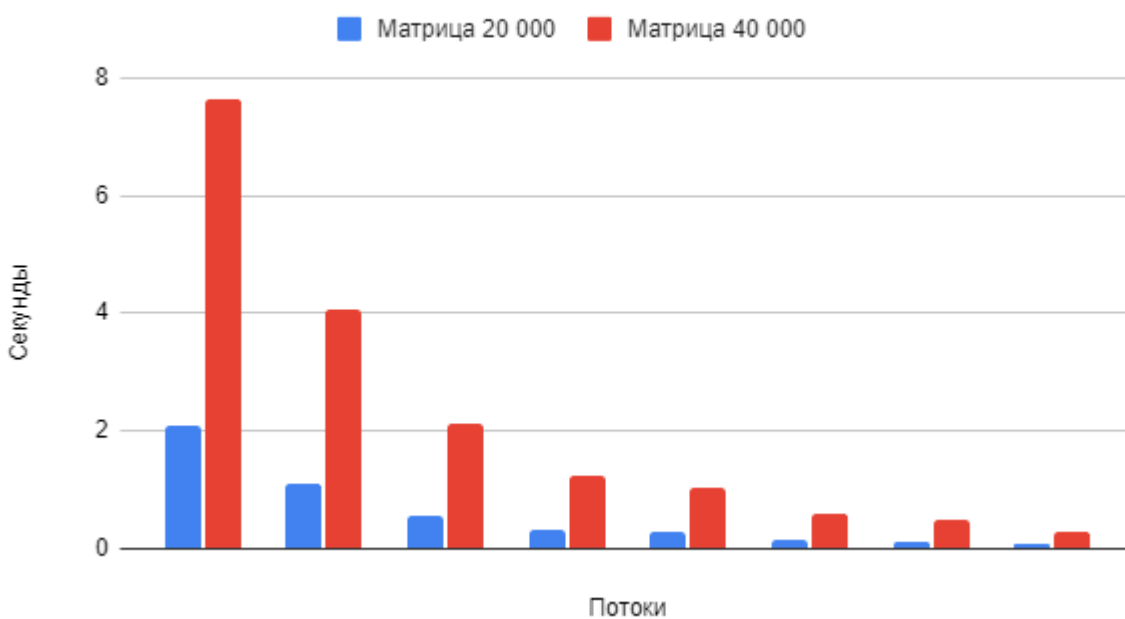
### ### OS

```
os:                    Ubuntu 22.04.3 LTS
```

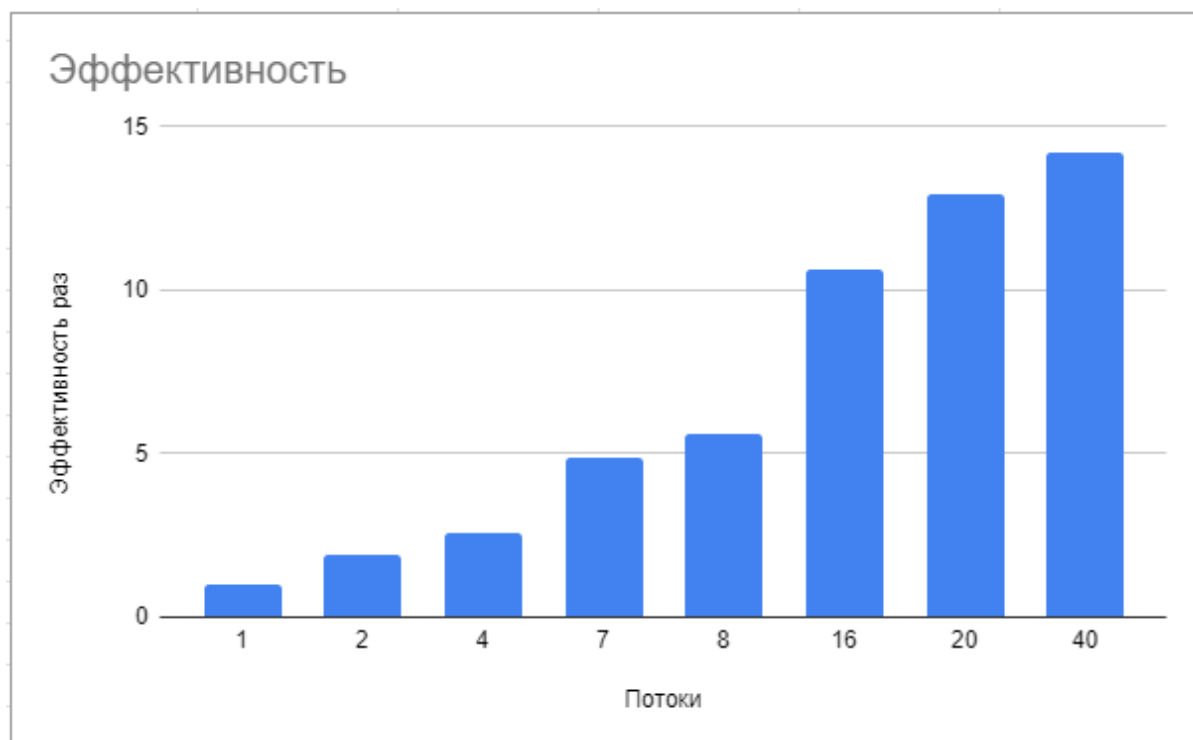
## Эффективность



## Время расчета



Потоки	Секунд на 20к	Секунд на 40к	На поток 20к	На поток 40к	1 Поток 40к	1 Поток 20к
1	2,088276	7,61973	1	1	7,61973	2,088276
2	1,1053	4,077479	1,889329594	1,868735559	7,61973	2,088276
4	0,548931	2,107584	3,804259552	3,615386148	7,61973	2,088276
7	0,306101	1,224775	6,822179607	6,221330448	7,61973	2,088276
8	0,283513	1,031749	7,365715152	7,385255522	7,61973	2,088276
16	0,152893	0,578685	13,65841471	13,16731901	7,61973	2,088276
20	0,120085	0,488302	17,3899821	15,60454391	7,61973	2,088276
40	0,07248	0,293886	28,81175497	25,9275025	7,61973	2,088276



Потоки	Время	Улучшение раз	
1	0,484725	1	0,484725
2	0,252489	1,919786605	0,484725
4	0,189821	2,553589961	0,484725
7	0,100001	4,847201528	0,484725
8	0,086962	5,573986339	0,484725
16	0,045644	10,61968714	0,484725
20	0,037442	12,94602318	0,484725
40	0,034119	14,20689352	0,484725

Вывод: Производительность растет нелинейно, несмотря на все оптимизации

## Задача 3

Сравнение производительности по потокам, относительно размера матрицы

	10000	20000	30000	40000	50000	60000	70000
1	6	13	25	35	41	64	89
8 dynamic	0,9	1,8	3,2	4,6	5,6	8	11
20 dynamic	0,4	0,8	1,5	2,1	2,5	3,6	5
40 dynamic	0,2	0,5	0,9	1,2	1,4	2	2,6
40 static	0,2	0,4	0,7	1	1,3	1,8	2,8
40 guided	0,2	0,4	0,7	1	1,2	1,8	2,4
40 for	0,2	0,4	0,7	1	1,2	1,8	2,7
40 global paralle	0,3	0,4	1,2	1,8	1,9	2	2,9 unstable?

В моих тестах влияние метода распараллеливания цикла оказалось незначительным, единственное, что важно, использовать в каждой функции отдельное

распараллеливание, таким образом время исполнения программы было более стабильным.