

Этап 1:

На нашем этапе 1, мы сгенерировали “топологию” связей по элементам.
Я передал свой 1й этап Ли Ваньшуню и получил этот этап от его.

Это сравнение выходных данных для маленькой сетки($N_x = 3, N_y = 3, N_z = 3, N = 27$).
У нас одинаковые выходные данные IA и JA.

```
miaohuihui@robot:~/pragram/c/courses/exercise1/main1$ ./pr data.bin
*****
IA
0 4 9 13 18 24 29 33 38 42 47 53 58 64 71 77 82 88 93 97 102 106 111 117 122 126 131 135
*****
JA
0 1 3 9 0 1 2 4 10 1 2 5 11 0 3 4 6 12 1 3 4 5 7 13 2 4 5 8 14 3 6 7 15 4 6 7 8 16 5 7 8 17 0 9 10 12 18 1 9 10 11 13 19 2 10 11 14 20 3 9 12 13 15 21 4 10 12 13 14 16 22 5 11 13 14 17 23 6 12 15 16 24 7
13 15 16 17 25 8 14 16 17 26 9 18 19 21 10 18 19 20 22 11 19 20 23 12 18 21 22 24 13 19 21 22 23 25 14 20 22 23 26 15 21 24 25 16 22 24 25 26 17 23 25 26
*****

miaohuihui@robot:~/pragram/c/courses/exercise1/main1$ ./pr data.bin
*****
IA
0 4 9 13 18 24 29 33 38 42 47 53 58 64 71 77 82 88 93 97 102 106 111 117 122 126 131 135
*****
JA
0 1 3 9 0 1 2 4 10 1 2 5 11 0 3 4 6 12 1 3 4 5 7 13 2 4 5 8 14 3 6 7 15 4 6 7 8 16 5 7 8 17 0 9 10 12 18 1 9 10 11 13 19 2 10 11 14 20 3 9 12 13 15 21 4 10 12 13 14 16 22 5 11 13 14 17 23 6 12 15 16 24 7
13 15 16 17 25 8 14 16 17 26 9 18 19 21 10 18 19 20 22 11 19 20 23 12 18 21 22 24 13 19 21 22 23 25 14 20 22 23 26 15 21 24 25 16 22 24 25 26 17 23 25 26
*****
```

Для сравнения времени работы: на моём этапе 1, время этапа генерации = 0.000050s, на этапе 1 Ли, время этапа генерации = 0.000098s.
Если $N_x = 20, N_y = 20, N_z = 20, N = 8000$, на моём этапе 1, время этапа генерации = 0.000979s; на этапе 1 Ли, время этапа генерации = 0.005426s. Мой код работает быстрее.

```
miaohuihui@robot:~/pragram/c/courses/exercise1/main1$ ./ma 3 3 3
N = 27

M = 135

Время этапа генерации = 0.000050
```

```
miaohuihui@robot:~/pragram/c/courses/exercise1/main1$ ./ma 3 3 3
N=27
Время этапа генерации = 0.000098
```

```
miaohuihui@robot:~/pragram/c/courses/exercise1/main1$ ./ma 20 20 20
N = 8000

M = 53600

Время этапа генерации = 0.000979
```

```
miaohuihui@robot:~/pragram/c/courses/exercise1/main1$ ./ma 20 20 20
N=8000
Время этапа генерации = 0.005426
```

Для сравнения расхода памяти. На моём ноутбуке, при использовании моего кода, максимум величины N может достигать 25000; при использовании кода Ли, максимум величины N может достигать 9000. Мой код лучше и меньше использует память.

```
miaohuihui@robot:~/pragram/c/courses/exercise1/main1$ ./ma 50 50 10
N = 25000

M = 168000

Время этапа генерации = 0.003084
```

```
miaohuihui@robot:~/pragram/c/courses/exercise1/main1$ ./ma 5 360 5
N=9000
Время этапа генерации = 0.004934
```

Этап 2 и этап 3:

Я соединил этап 1, этап 2 и этап 3 в одном файле main.c и печатал все выходные данные в двоичный файл data.bin. Для проверки данных я написал printdata.c.

В файле main.c я использовал код этапа 1 Ли.

Для maxiter, я определил так: #define maxiter 50.

Невязка, это $r = Ax - b$, поэтому это вектор.

При условии: $N_x = 3$, $N_y = 3$, $N_z = 3$, $N = 27$, $tol = 0.000001$, у нас результат такой:

```
k = 0
1.000000 0.540302 -0.653644 -0.911130 -0.957659 0.991203 -0.127964 0.300593 0.391857 0
.776686 0.862319 -0.048664 0.871147 0.798496 0.342466 0.367319 -0.039791 0.999648 -0.9
14730 -0.960179 -0.525296 0.383671 0.981100 0.350741 -0.463775 -0.984387 -0.848538
k = 1
-0.328397 -0.000000 0.000000 -0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0
.000000 0.147146 0.000000 -0.123316 0.000000 0.000000 0.000000 0.000000 0.000000 -0.067
808 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
k = 2
0.000000 0.103241 -0.000000 0.068511 0.000000 0.000000 0.000000 0.000000 0.000000 0.10
7911 -0.000000 0.036033 0.000000 0.029707 0.000000 -0.020885 0.000000 0.000000 0.00000
0 -0.027303 0.000000 0.041923 0.000000 0.000000 0.000000 0.000000 0.000000
k = 3
-0.076451 0.000000 0.015862 -0.000000 0.001680 -0.000000 -0.015225 0.000000 -0.000000
0.000000 0.044083 0.000000 -0.027269 0.000000 0.000864 0.000000 -0.001432 0.000000 -0
.018820 0.000000 -0.012106 0.000000 0.010036 0.000000 -0.004865 0.000000 0.000000
k = 4
0.000000 0.028249 -0.000000 0.020133 0.000000 0.003675 0.000000 -0.002151 -0.000000 0
.026302 -0.000000 0.019263 0.000000 0.009174 -0.000000 -0.008112 0.000000 -0.000226 0.0
00000 -0.009879 0.000000 0.011915 -0.000000 0.002203 0.000000 -0.001037 0.000000
k = 5
-0.020211 0.000000 0.006717 0.000000 -0.000716 0.000000 -0.005045 0.000000 0.001056 0
.000000 0.013260 0.000000 -0.007775 0.000000 0.001052 -0.000000 -0.000102 0.000000 -0.0
05487 0.000000 -0.005625 0.000000 0.003165 0.000000 -0.001831 -0.000000 -0.000627
k = 6
0.000000 0.007942 -0.000000 0.005815 0.000000 0.001818 0.000000 -0.001312 -0.000000 0
.007233 -0.000000 0.007118 0.000000 0.002450 0.000000 -0.002536 0.000000 -0.000077 0.00
0000 -0.003477 0.000000 0.003508 -0.000000 0.000919 0.000000 -0.000607 0.000000
k = 7
-0.005668 0.000000 0.002345 -0.000000 -0.000492 -0.000000 -0.001561 0.000000 0.000582
0.000000 0.003990 0.000000 -0.002245 0.000000 0.000473 0.000000 0.000070 -0.000000 -0
.001697 0.000000 -0.002050 0.000000 0.000809 0.000000 -0.000614 0.000000 -0.000291
k = 8
0.000000 0.002298 -0.000000 0.001725 0.000000 0.000687 0.000000 -0.000544 -0.000000 0
.002078 -0.000000 0.002375 0.000000 0.000642 -0.000000 -0.000773 -0.000000 -0.000008 0
.000000 -0.001188 0.000000 0.001032 -0.000000 0.000299 0.000000 -0.000247 0.000000
k = 9
-0.001646 0.000000 0.000767 0.000000 -0.000210 -0.000000 -0.000482 0.000000 0.000234 0
.000000 0.001203 0.000000 -0.000654 0.000000 0.000171 -0.000000 0.000047 -0.000000 -0
.000530 0.000000 -0.000689 0.000000 0.000185 0.000000 -0.000197 0.000000 -0.000101
k = 10
0.000000 0.000677 -0.000000 0.000522 0.000000 0.000236 0.000000 -0.000198 -0.000000 0
.000612 0.000000 0.000761 -0.000000 0.000169 -0.000000 -0.000235 -0.000000 0.000004 0.0
00000 -0.000394 0.000000 0.000303 0.000000 0.000089 0.000000 -0.000088 0.000000
k = 11
-0.000488 0.000000 0.000244 0.000000 -0.000078 -0.000000 -0.000150 0.000000 0.000084 0
.000000 0.000364 -0.000000 -0.000192 0.000000 0.000057 0.000000 0.000021 -0.000000 -0
.000165 0.000000 -0.000223 0.000000 0.000039 -0.000000 -0.000062 0.000000 -0.000031
```

Для большой величины N , я печатал среднюю величину всех величин вектора g .

При условии: $N_x = 20$, $N_y = 20$, $N_z = 20$, $N = 8000$, $tol = 0.01$, у нас результат такой:

```
mlaohuihui@robot:~/pragram/c/courses/exercise1/main1$ ./ma 20 20 20 0.01
N=8000
Время этапа генерации = 0.005182

Время этапа заполнения = 0.016660

k = 0
0.000964
k = 1
0.001422
k = 2
0.000821
k = 3
0.000224
k = 4
0.000217
k = 5
0.000016
k = 6
0.000053
k = 7
-0.000003
k = 8
0.000013
```

При условии: $N_x = 20$, $N_y = 20$, $N_z = 20$, $N = 8000$, $\text{tol} = 0.0001$, у нас результат такой:

```
miaohuihui@robot:~/program/c/courses/exercise1/main1$ ./ma 20 20 20 0.0001
N=8000
Время этапа генерации = 0.004902
Время этапа заполнения = 0.017221

k = 0
0.000964
k = 1
0.001422
k = 2
0.000821
k = 3
0.000224
k = 4
0.000217
k = 5
0.000016
k = 6
0.000053
k = 7
-0.000003
k = 8
0.000013
k = 9
-0.000002
k = 10
0.000003
k = 11
-0.000001
```

При условии: $N_x = 20$, $N_y = 20$, $N_z = 20$, $N = 8000$, $\text{tol} = 0.000001$, у нас результат такой:

```
miaohuihui@robot:~/program/c/courses/exercise1/main1$ ./ma 20 20 20 0.000001
N=8000
Время этапа генерации = 0.005115

Время этапа заполнения = 0.016836

k = 0
0.000964
k = 1
0.001422
k = 2
0.000821
k = 3
0.000224
k = 4
0.000217
k = 5
0.000016
k = 6
0.000053
k = 7
-0.000003
k = 8
0.000013
k = 9
-0.000002
k = 10
0.000003
k = 11
-0.000001
k = 12
0.000001
k = 13
-0.000000
k = 14
0.000000
```

Результат покажет, что величина k возрастает, величина невязки уменьшается.

Когда я определил $N_x = 20$, $N_y = 20$, $N_z = 20$, $N = 8000$ и только изменил величину tol . Результат покажет, что если величина tol уменьшается, то величина k возрастает.

При условии: $N_x = 10$, $N_y = 10$, $N_z = 10$, $N = 1000$, $\text{tol} = 0.000001$, у нас результат такой:

```
miaohuihui@robot:~/program/c/courses/exercise1/main1$ ./ma 10 10 10 0.000001
N=1000
Время этапа генерации = 0.000990

Время этапа заполнения = 0.002327

k = 0
0.012990
k = 1
-0.002717
k = 2
0.000262
k = 3
-0.000548
k = 4
0.000455
k = 5
-0.000028
k = 6
0.000178
k = 7
0.000017
k = 8
0.000059
k = 9
0.000010
k = 10
0.000019
k = 11
0.000004
k = 12
0.000006
k = 13
0.000001
k = 14
0.000002
```

При условии: $N_x = 15$, $N_y = 10$, $N_z = 20$, $N = 3000$, $\text{tol} = 0.000001$, у нас результат такой:

```
miaohuihui@robot:~/pragram/c/courses/exercise1/main1$ ./ma 15 10 20 0.000001
N=3000
Время этапа генерации = 0.000921

Время этапа заполнения = 0.001337

k = 0
0.002401
k = 1
0.005541
k = 2
0.011106
k = 3
0.000730
k = 4
0.002807
k = 5
0.000162
k = 6
0.000721
k = 7
0.000043
k = 8
0.000195
k = 9
0.000012
k = 10
0.000055
k = 11
0.000004
k = 12
0.000016
k = 13
0.000001
k = 14
0.000005
k = 15
0.000000
k = 16
0.000001
```

Когда я определил tol и только изменил величину N . Результат покажет, что у нас нет ясного отношения между величиной N и величиной k .

Этап 4:

На этапе 4 я сделаю сравнение времени работы между исходной программой и параллельной программой. Я только написал код с помощью OpenMP на этапе 2 и этапе 3.

```
miaohuihui@robot:~/pragram/c/courses/exercise1/main1$ ./ma 10 10 10 0.000001
N=1000
Время этапа генерации = 0.000685

Время этапа заполнения = 0.001962

k = 0
Время операции SpMV = 0.000135
k = 1
Время операции SpMV = 0.000047 Время операции dot = 0.000012 Время операции SpMV = 0.000132 Время операции dot = 0.000012 Время операции ахрбу = 0.000015
k = 2
Время операции SpMV = 0.000048 Время операции dot = 0.000012 Время операции ахрбу = 0.000015 Время операции SpMV = 0.000134 Время операции dot = 0.000012
Время операции ахрбу = 0.000016
k = 3
Время операции SpMV = 0.000044 Время операции dot = 0.000012 Время операции ахрбу = 0.000016 Время операции SpMV = 0.000132
Время операции dot = 0.000012
Время операции ахрбу = 0.000016
Время операции ахрбу = 0.000015
k = 4
Время операции SpMV = 0.000045 Время операции dot = 0.000012 Время операции ахрбу = 0.000015 Время операции SpMV = 0.000131
Время операции dot = 0.000012
Время операции ахрбу = 0.000015
Время операции ахрбу = 0.000015
k = 5
Время операции SpMV = 0.000045 Время операции dot = 0.000012 Время операции ахрбу = 0.000016 Время операции SpMV = 0.000135
Время операции dot = 0.000012
Время операции ахрбу = 0.000015
Время операции ахрбу = 0.000016
k = 6
Время операции SpMV = 0.000047 Время операции dot = 0.000012 Время операции ахрбу = 0.000015 Время операции SpMV = 0.000148
Время операции dot = 0.000012
Время операции ахрбу = 0.000016
Время операции ахрбу = 0.000016
k = 7
Время операции SpMV = 0.000035 Время операции dot = 0.000011 Время операции ахрбу = 0.000014 Время операции SpMV = 0.000118
Время операции dot = 0.000011
Время операции ахрбу = 0.000016
Время операции ахрбу = 0.000014
k = 8
Время операции SpMV = 0.000033 Время операции dot = 0.000011 Время операции ахрбу = 0.000015 Время операции SpMV = 0.000094
Время операции dot = 0.000012
Время операции ахрбу = 0.000016
Время операции ахрбу = 0.000016
k = 9
Время операции SpMV = 0.000034 Время операции dot = 0.000012 Время операции ахрбу = 0.000014 Время операции SpMV = 0.000100
Время операции dot = 0.000012
Время операции ахрбу = 0.000015
Время операции ахрбу = 0.000014
k = 10
Время операции SpMV = 0.000034 Время операции dot = 0.000012 Время операции ахрбу = 0.000016 Время операции SpMV = 0.000098
Время операции dot = 0.000011
Время операции ахрбу = 0.000016
Время операции ахрбу = 0.000015
k = 11
Время операции SpMV = 0.000033 Время операции dot = 0.000012 Время операции ахрбу = 0.000015 Время операции SpMV = 0.000123
Время операции dot = 0.000012
Время операции ахрбу = 0.000015
Время операции ахрбу = 0.000015
k = 12
Время операции SpMV = 0.000038 Время операции dot = 0.000012 Время операции ахрбу = 0.000015 Время операции SpMV = 0.000099
Время операции dot = 0.000012
Время операции ахрбу = 0.000015
Время операции ахрбу = 0.000033
k = 13
Время операции SpMV = 0.000026 Время операции dot = 0.000008 Время операции ахрбу = 0.000011 Время операции SpMV = 0.000070
Время операции dot = 0.000008
Время операции ахрбу = 0.000011
Время операции ахрбу = 0.000011
k = 14
Время операции SpMV = 0.000025 Время операции dot = 0.000008 Время операции ахрбу = 0.000011 Время операции SpMV = 0.000071
Время операции dot = 0.000008
Время операции ахрбу = 0.000011
Время операции ахрбу = 0.000012

Время этапа решения СЛАУ = 0.028780
```



```

miaohuihui@robot:~/program/c/courses/exercise1/main1$ ./ma 20 20 20 0.000001
N=8000
Время этапа генерации = 0.015361

Время этапа заполнения = 0.028968

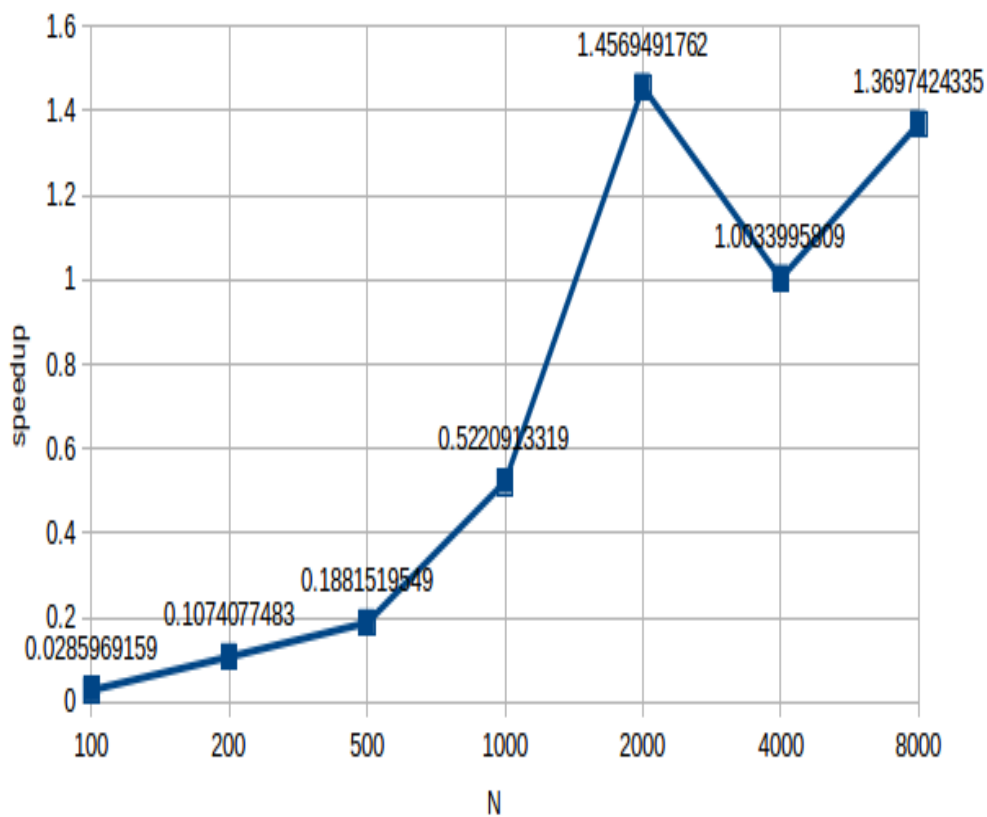
k = 0
Время операции SpMV = 0.000473
k = 1
Время операции SpMV = 0.000197 Время операции dot = 0.000043 Время операции SpMV = 0.000584 Время операции dot = 0.000037
Время операции ахрбу = 0.000048 Время операции ахрбу = 0.000054
k = 2
Время операции SpMV = 0.000144 Время операции dot = 0.000039 Время операции ахрбу = 0.000051 Время операции SpMV = 0.000427
Время операции dot = 0.000048 Время операции ахрбу = 0.000053 Время операции ахрбу = 0.000064
k = 3
Время операции SpMV = 0.000145 Время операции dot = 0.000043 Время операции ахрбу = 0.000051 Время операции SpMV = 0.000366
Время операции dot = 0.000035 Время операции ахрбу = 0.000043 Время операции ахрбу = 0.000043
k = 4
Время операции SpMV = 0.000151 Время операции dot = 0.000061 Время операции ахрбу = 0.000054 Время операции SpMV = 0.000318
Время операции dot = 0.000069 Время операции ахрбу = 0.000045 Время операции ахрбу = 0.000056
k = 5
Время операции SpMV = 0.000178 Время операции dot = 0.000051 Время операции ахрбу = 0.000056 Время операции SpMV = 0.000646
Время операции dot = 0.000059 Время операции ахрбу = 0.000079 Время операции ахрбу = 0.000077
k = 6
Время операции SpMV = 0.000231 Время операции dot = 0.000093 Время операции ахрбу = 0.000080 Время операции SpMV = 0.000445
Время операции dot = 0.000043 Время операции ахрбу = 0.000046 Время операции ахрбу = 0.000046
k = 7
Время операции SpMV = 0.000095 Время операции dot = 0.000036 Время операции ахрбу = 0.000040 Время операции SpMV = 0.000270
Время операции dot = 0.000030 Время операции ахрбу = 0.000041 Время операции ахрбу = 0.000040
k = 8
Время операции SpMV = 0.000092 Время операции dot = 0.000031 Время операции ахрбу = 0.000039 Время операции SpMV = 0.000277
Время операции dot = 0.000030 Время операции ахрбу = 0.000040 Время операции ахрбу = 0.000039
k = 9
Время операции SpMV = 0.000084 Время операции dot = 0.000029 Время операции ахрбу = 0.000042 Время операции SpMV = 0.000258
Время операции dot = 0.000030 Время операции ахрбу = 0.000039 Время операции ахрбу = 0.000039
k = 10
Время операции SpMV = 0.000084 Время операции dot = 0.000029 Время операции ахрбу = 0.000043 Время операции SpMV = 0.000265
Время операции dot = 0.000030 Время операции ахрбу = 0.000039 Время операции ахрбу = 0.000039
k = 11
Время операции SpMV = 0.000082 Время операции dot = 0.000030 Время операции ахрбу = 0.000039 Время операции SpMV = 0.000266
Время операции dot = 0.000030 Время операции ахрбу = 0.000039 Время операции ахрбу = 0.000040
k = 12
Время операции SpMV = 0.000083 Время операции dot = 0.000029 Время операции ахрбу = 0.000042 Время операции SpMV = 0.000256
Время операции dot = 0.000030 Время операции ахрбу = 0.000039 Время операции ахрбу = 0.000042
k = 13
Время операции SpMV = 0.000084 Время операции dot = 0.000030 Время операции ахрбу = 0.000039 Время операции SpMV = 0.000290
Время операции dot = 0.000032 Время операции ахрбу = 0.000039 Время операции ахрбу = 0.000042
k = 14
Время операции SpMV = 0.000083 Время операции dot = 0.000030 Время операции ахрбу = 0.000043 Время операции SpMV = 0.000264
Время операции dot = 0.000029 Время операции ахрбу = 0.000039 Время операции ахрбу = 0.000039

Время этапа решения СЛАУ = 0.485526

```

Результат покажет, что если N возрастает, то времени работы(время этапа генерации, время этапа заполнения, время этапа решения СЛАУ, время операции dot, время операции ахрбу, время операции SpMV) тоже возрастает.

Сейчас я использую OpenMP при условии $N_x = 20$, $N_y = 20$, $N_z = 20$, $N = 8000$, $\text{tol} = 0.000001$.
Здесь я только сравнил время этапа решения СЛАУ.



Результат покажет, что если N возрастает, то ускорение возрастает.

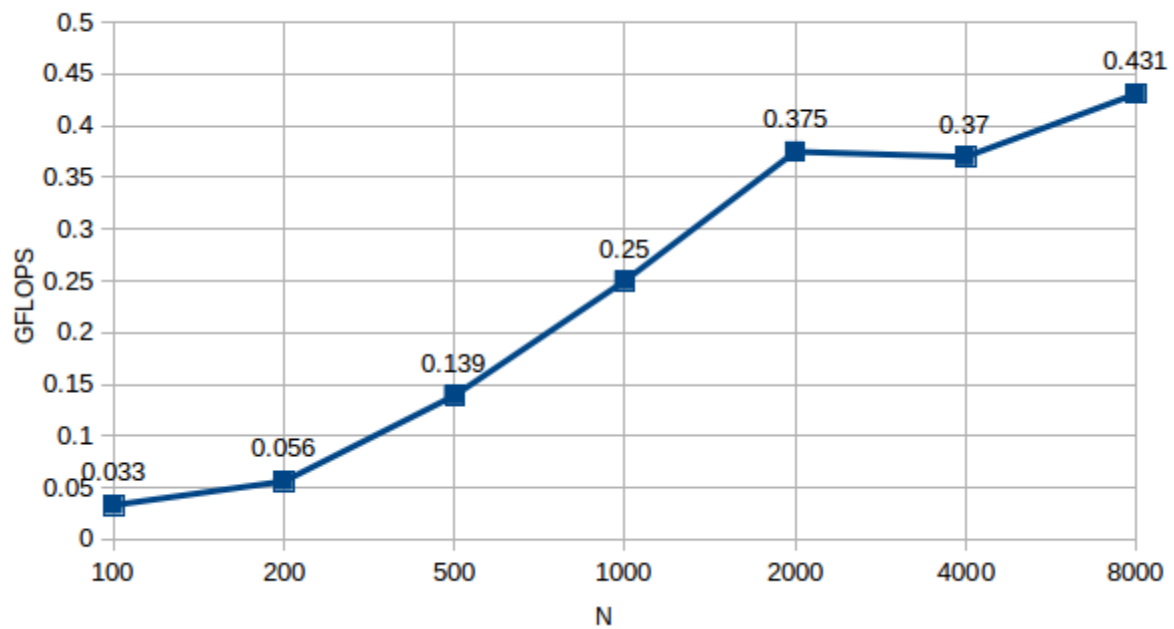
Здесь у меня интересное открытие для операции SpMV:

```
74
75 double *SpMV(int *IA, int *JA, double *A, double *Vector, int N){
76
77     double t1 = omp_get_wtime();
78     int i, j, k;
79     double *tmp = (double *)malloc(N * sizeof(double));
80     #pragma omp parallel for private(i, j, k) num_threads(16) schedule(dynamic, 100)
81     for(i = 0; i < N; i++){
82         double sum = 0.0;
83         const int j_begin = IA[i];
84         const int j_end = IA[i + 1];
85         for(j = j_begin; j < j_end; j++){
86             k = JA[j];
87             sum += A[j] * Vector[k];
88         }
89         tmp[i] = sum;
90     }
91     double t2 = omp_get_wtime();
92     printf("Время операции SpMV = %f ", t2 - t1);
93     return tmp;
94 }
```

```
69
70 double *SpMV(double **Matrix, double *Vector, int N){
71
72     double t1 = omp_get_wtime();
73     int i, j;
74     double *tmp = (double *)malloc(N * sizeof(double));
75     for(i = 0; i < N; i++){
76         for(j = 0; j < N; j++){
77             tmp[i] += Matrix[i][j] * Vector[j];
78         }
79     }
80     double t2 = omp_get_wtime();
81     printf("Время операции SpMV = %f ", t2 - t1);
82     return tmp;
83 }
84
```

У нас два формата операции SpMV. Если я определю операцию SpMV с входными данными формата CSR, время чрезвычайно меньше чем с входными данными формата матрицы. Поэтому в моей программе я определил операцию SpMV с входными данными формата CSR. Но у нас есть недостаток о чём, что ускорение не очевидно.

График GFLOPS от N



Я подсчитал GFLOPS с помощью Valgrind. ($\text{GFLOPS} = \text{refs} / \text{time} * e^{-9}$)

$\text{TBP} = \text{BW} * \text{AI} = 6\text{GiB/s} * 0.25\text{FLOP/byte} = 1.5\text{GFLOPS} \gg 0.431\text{GFLOPS}$