

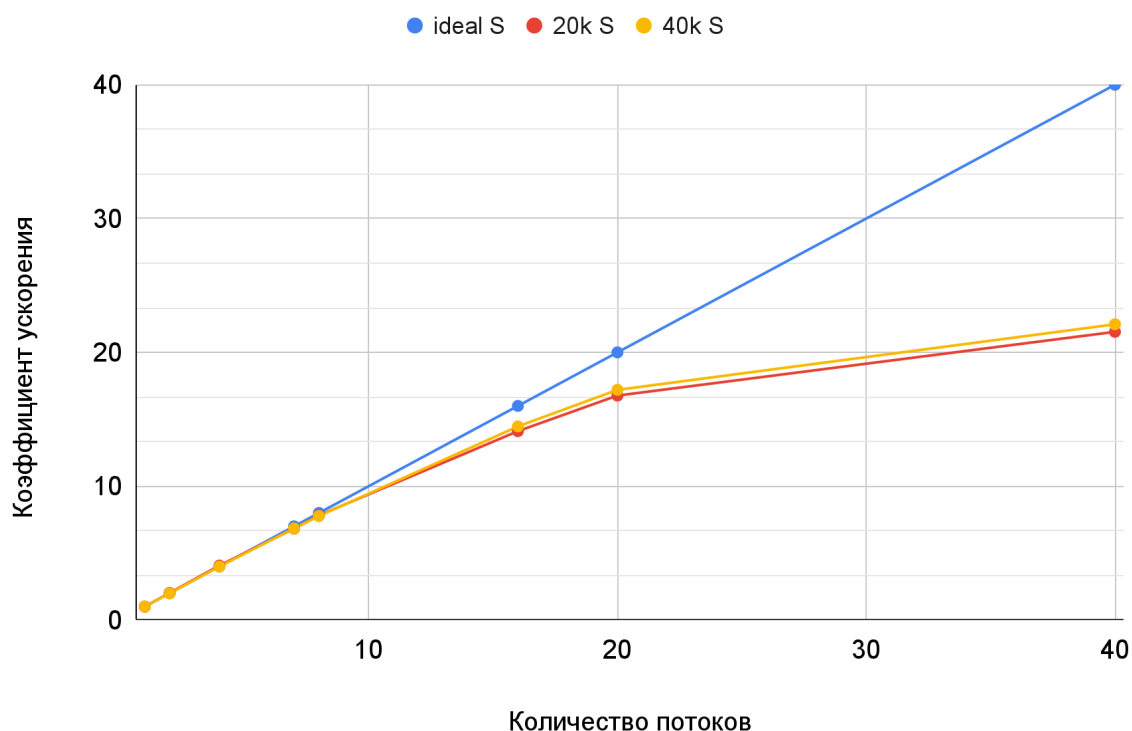
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Address sizes: 46 bits physical, 48 bits virtual
Byte Order: Little Endian
CPU(s): 80
On-line CPU(s) list: 0-79
Vendor ID: GenuineIntel
Model name: Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz
CPU family: 6
Model: 85
Thread(s) per core: 2
Core(s) per socket: 20
Socket(s): 2
Stepping: 7
CPU max MHz: 3900.0000
CPU min MHz: 1000.0000
Caches (sum of all):
L1d: 1.3 MiB (40 instances)
L1i: 1.3 MiB (40 instances)
L2: 40 MiB (40 instances)
L3: 55 MiB (2 instances)
NUMA:
NUMA node(s): 2
NUMA node0 CPU(s): 0-19,40-59
NUMA node1 CPU(s): 20-39,60-79
ProLiant XL270d Gen10
available: 2 nodes (0-1)
node 0 size: 385636 MB
node 0 free: 2878 MB
node 1 size: 387008 MB
node 1 free: 921 MB
PRETTY_NAME="Ubuntu 22.04.3 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.3 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy

Задание 1

Программа запускалась 5 раз подряд. В таблице средние данные по результатам этих запусков.

M = N	Количество потоков							
	1	2	4	7	8	16	20	40
20k T	9,252318	4,562834	2,273896	1,35425	1,183314	0,655497	0,5516348	0,4296272
40k T	37,20206	18,63902	9,313534	5,43243	4,78683	2,571206	2,162848	1,683396
ideal S	1	2	4	7	8	16	20	40
20k S	1	2,027756872	4,068927515	6,83206055	7,818988029	14,1149662	16,77254227	21,53568955
40k S	1	1,995923605	3,994408567	6,848143464	7,771752914	14,46872013	17,20049675	22,09941095

Ускорение программы относительно идеального графика ускорения



Вывод:

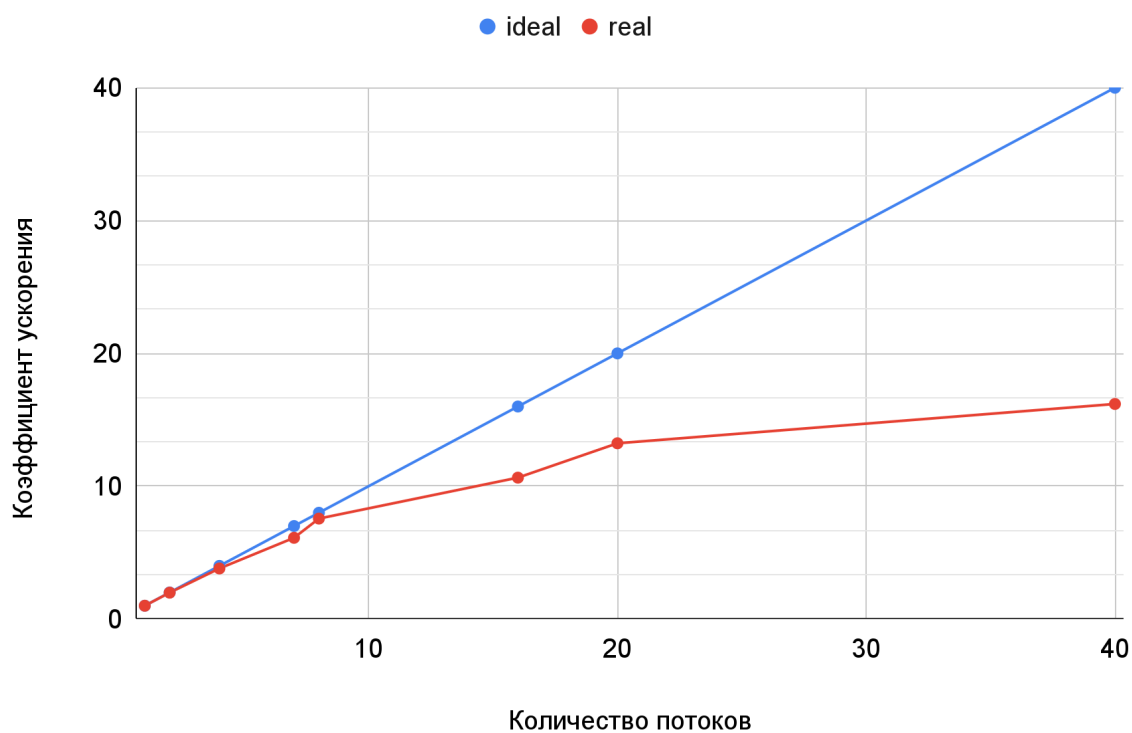
Коэффициент ускорения снижается с увеличением числа потоков, но не зависит от размера матрицы.

Задание 2

Программа запускалась 5 раз подряд. В таблице средние данные по результатам этих запусков.

M = N	Количество потоков							
	1	2	4	7	8	16	20	40
	T	T	T	T	T	T	T	T
real	0,514239 6	0,259181	0,135138 6	0,084093 44	0,068004 08	0,048311 9	0,038867	0,031760 86
ideal	1	2	4	7	8	16	20	40
real	1	1,984094 513	3,805275 473	6,115097 682	7,561893 345	10,64416 013	13,23075 102	16,19098 475

Ускорение программы относительно идеального графика ускорения



Вывод:

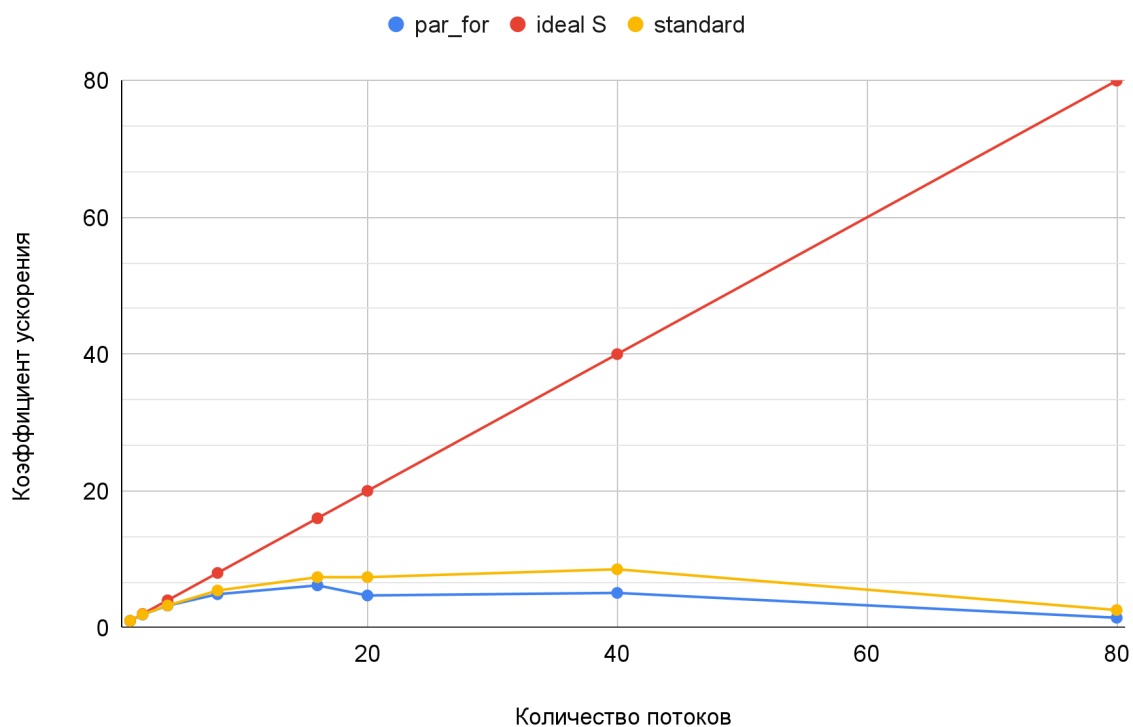
Кoeffициент ускорения снижается с увеличением числа потоков.

Задание 3

Программа запускалась 5 раз подряд. В таблице средние данные по результатам этих запусков.

M = N	Количество потоков							
	1	2	4	8	16	20	40	80
parallel	1,860386	0,960520 6	0,573679 6	0,3423848	0,251973 4	0,251869 8	0,217907 8	0,721989 6
par_for	1,648664	0,864247 2	0,512107 6	0,3370946	0,266686 4	0,350116 4	0,324377 4	1,136721 4
ideal S	1	2	4	8	16	20	40	80
parallel	1	1,936851 745	3,242900 741	5,4336115 39	7,383263 471	7,386300 382	8,537491 545	2,576749 028
par_for	1	1,907630 132	3,219370 304	4,8908051 33	6,182032 53	4,708902 525	5,082548 908	1,450367 698

Ускорение программы относительно идеального графика ускорения



Вывод:

Коэффициент ускорения снижается с увеличением числа потоков.

Стандартный способ на малом количестве потоков работает медленнее, но с увеличением количества потоков он становится

быстрее, чем способ с `#pragma omp parallel for`. Ещё стандартный способ имеет больший коэффициент ускорения.