

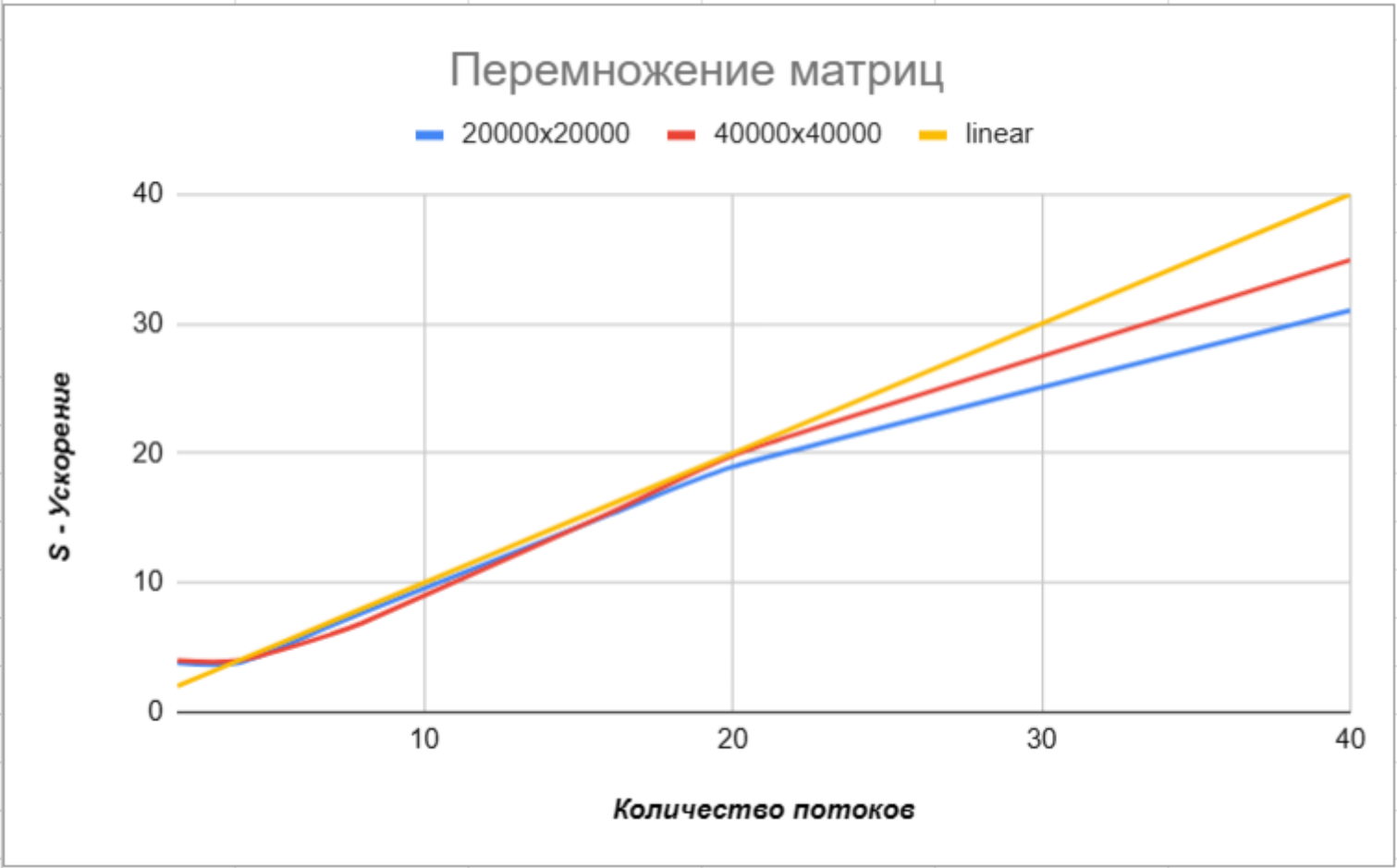
Задание 1

- **Server name:** ProLiant XL270d Gen10
- **CPU Model name:** Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz
- **available:** 2 nodes (0-1)
- **node 0 size:** 385636 MB
- **node 1 size:** 387008 MB
- **operating system:** Ubuntu 22.04.3 LTS

Таблица:

M+N	Количество потоков															
	2			4		7		8		16		20		40		
	T1	T2	S2	T4	S4	T7	S7	T8	S8	T16	S16	T20	S20	T40	S40	
20000	1,916485	1,028366	3,816445	0,502165	3,816445	0,286254	6,695051	0,250508	7,650394	0,12566	15,25135	0,101136	18,94958	0,061774	31,02414	
40000	8,394908	4,604547	3,998154	2,099696	3,998154	1,393185	6,025695	1,217807	6,893463	0,546283	15,36732	0,403214	19,81998	0,240454	34,91274	

График:



Вывод: при увеличении количества потоков ускорение падает, наиболее близкое к линейному при количестве потоков <=20.

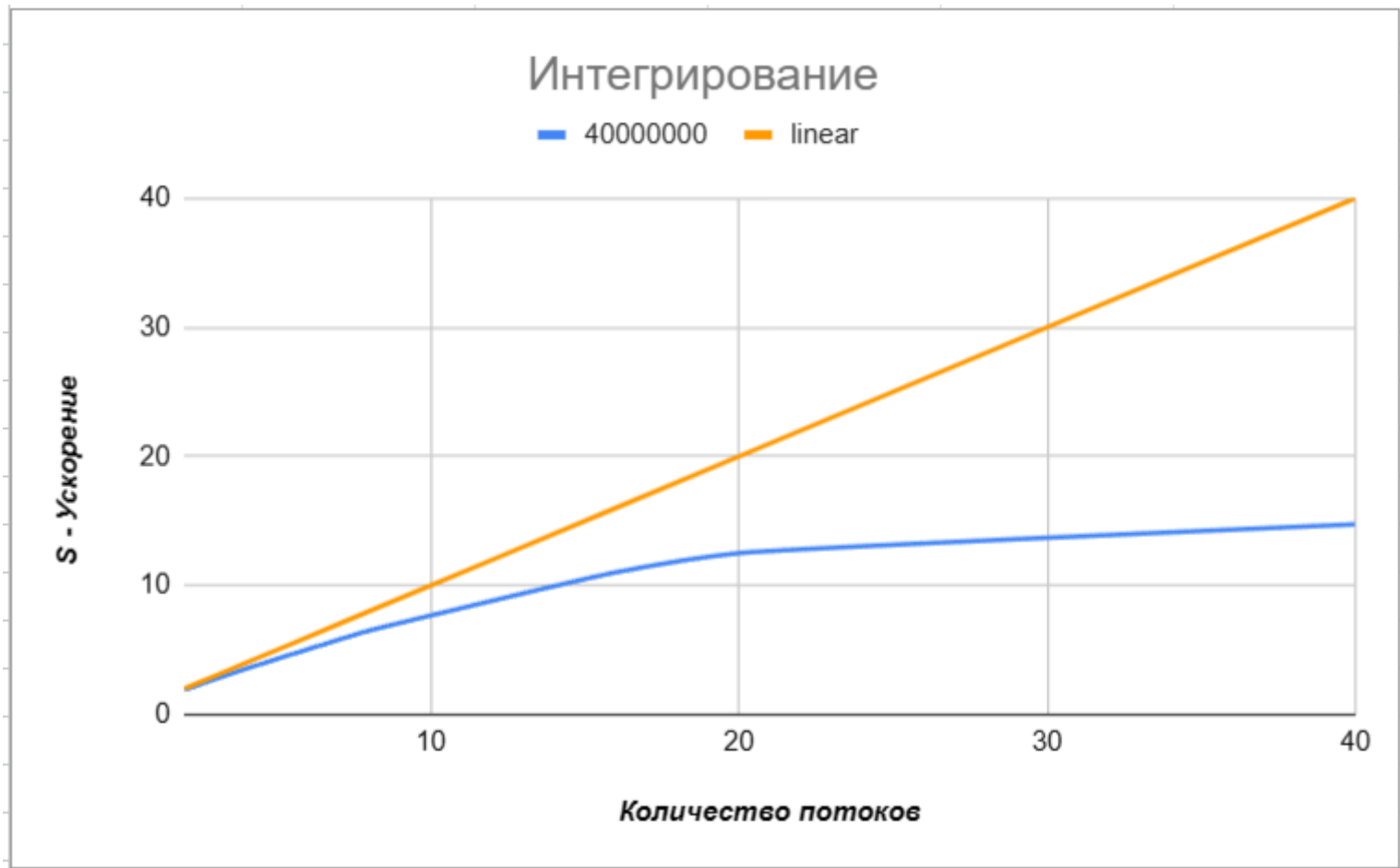
Задание 2

- **Server name:** ProLiant XL270d Gen10
- **CPU Model name:** Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz
- **available:** 2 nodes (0-1)
- **node 0 size:** 385636 MB
- **node 1 size:** 387008 MB
- **operating system:** Ubuntu 22.04.3 LTS

Таблица:

nsteps	Количество потоков															
	2			4		7		8		16		20		40		
	T1	T2	S2	T4	S4	T7	S7	T8	S8	T16	S16	T20	S20	T40	S40	
40000000	0,465904	0,243566	1,912849	0,131504	3,542884	0,080555	5,783701	0,071676	6,500101	0,042308	11,01218	0,037266	12,50213	0,031626	14,73192	

График:



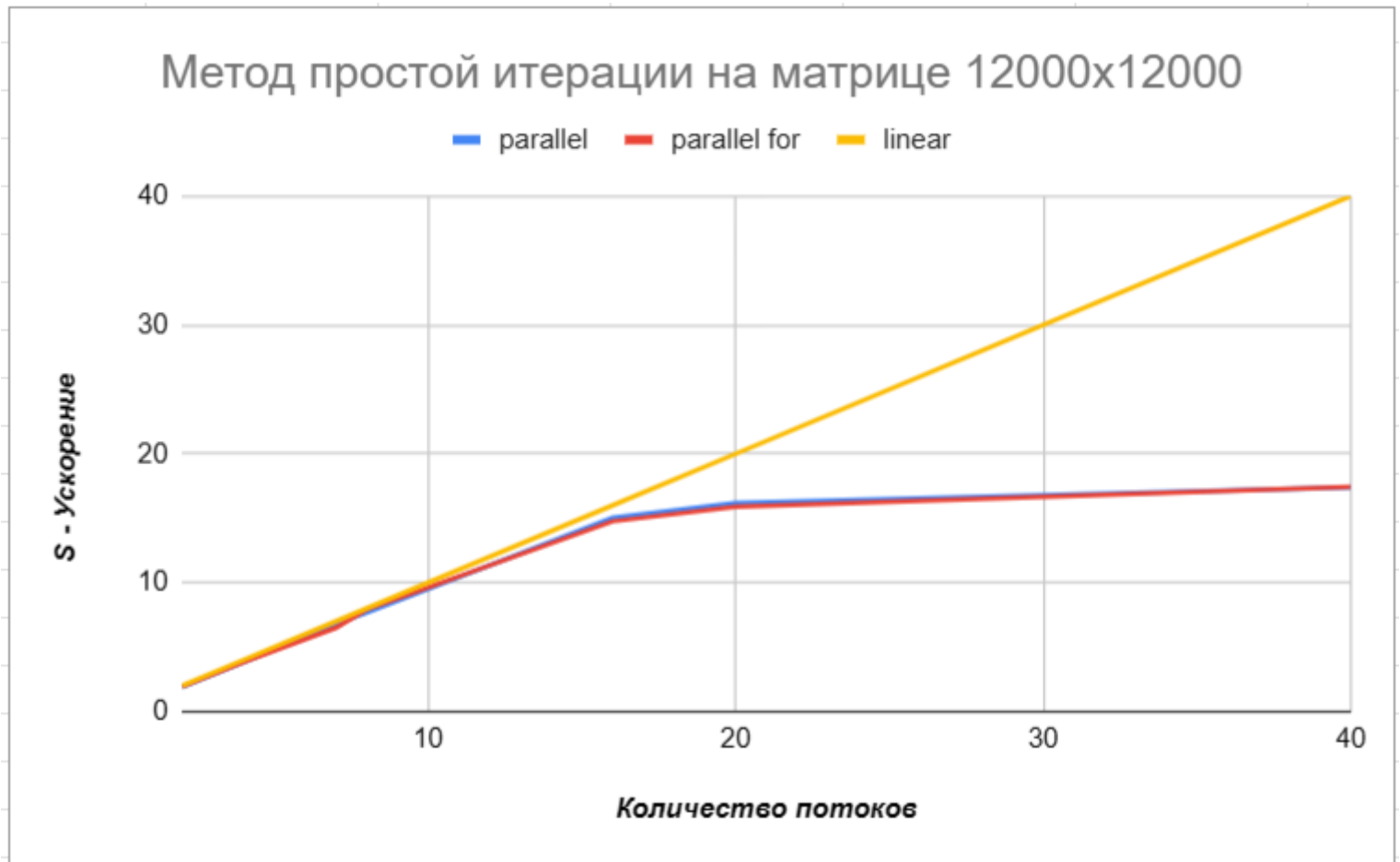
Вывод: при увеличении количества потоков ускорение падает, наиболее близкое к линейному при количестве потоков <=20.

Задание 3

Таблица:

	Количество потоков														
	2			4		7		8		16		20		40	
12000	T1	T2	S2	T4	S4	T7	S7	T8	S8	T16	S16	T20	S20	T40	S40
parallel	43,4963	23,1067	1,882411	11,3008	3,848958	6,40407	6,791978	5,68032	7,657368	2,89389	15,03	2,68616	16,193	2,50285	17,379
parallel for	43,2381	22,4675	1,924473	11,3277	3,817024	6,65948	6,492714	5,46784	7,907711	2,92943	14,76	2,72348	15,876	2,47847	17,445

График:



Вывод: Оптимизируя код, я пришёл к оптимальному подсчёту количества итераций на поток

$N = m / (\text{numThread} * 3)$, где m -количество строк матрицы, а numThread -количество потоков. Из графика видно, что разница между методами не такая значительная, минимальный прирост заметен в методе охватывающим весь итерационный алгоритм между 10-30 потоками.