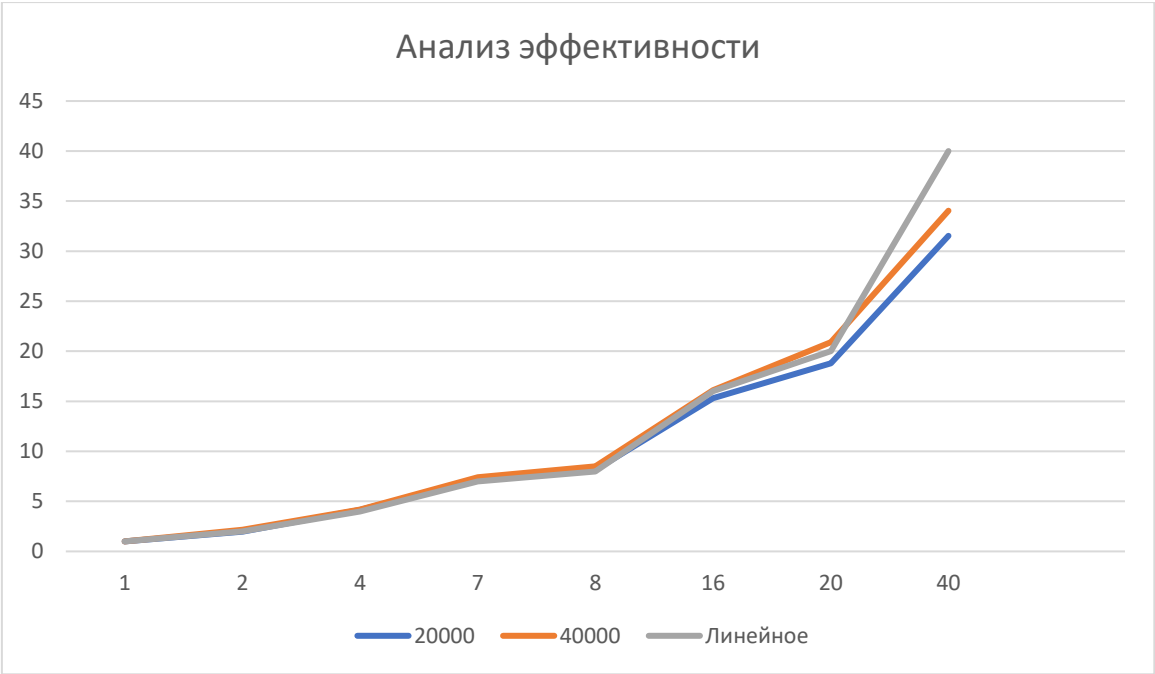
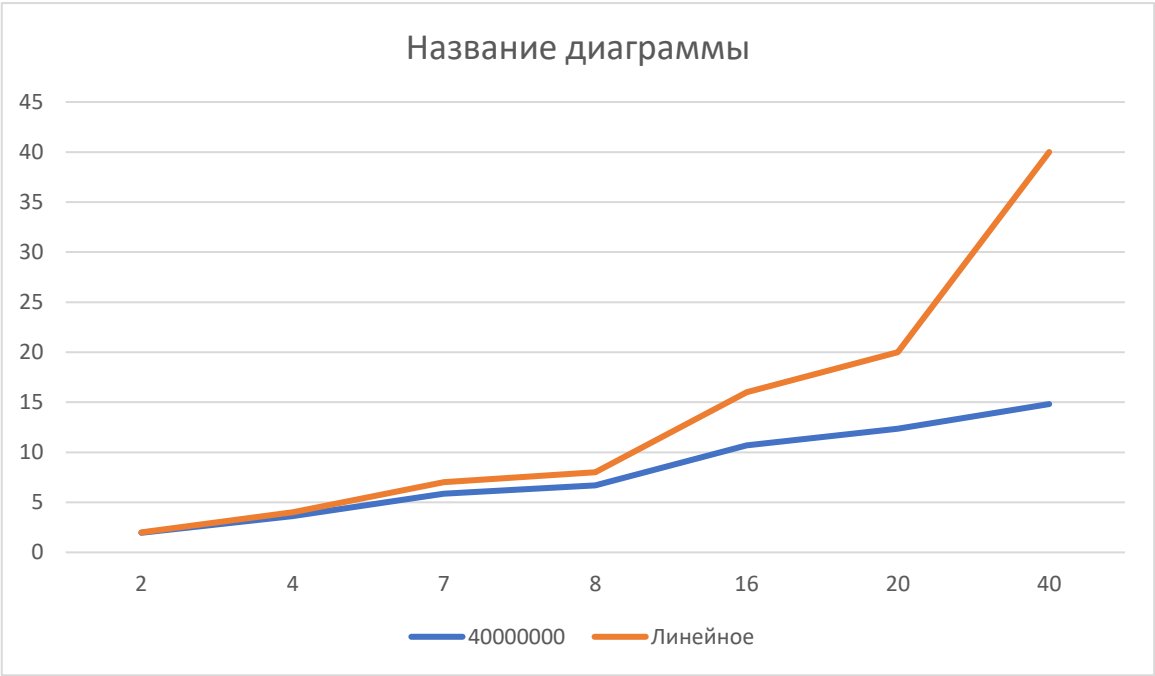


M=N	Количество потоков																
	1			2		4		7		8		6		20		40	
	T1	T2	S2	T3	S3	T4	S4	T5	S5	T6	S6	T7	S7	T8	S8	T9	S9
20000	4.08	4.08	1.00	2.06	1.97	0.98	4.15	0.57	7.17	0.50	8.14	0.27	15.32	0.22	18.78	0.13	31.53
40000	16.68	16.68	1.00	7.70	2.16	3.98	4.19	2.25	7.42	1.96	8.51	1.03	16.10	0.80	20.91	0.49	34.04



При увеличении потоков ускорение стремится к линейному значению

40000000	2		4		7		8		6		20		40	
	T	S	T	S	T	S	T	S	T	S	T	S	T	S
	0.24	1.96	0.13	3.61	0.08	5.86	0.07	6.71	0.044	10.68	0.038	12.36	0.0317	14.82



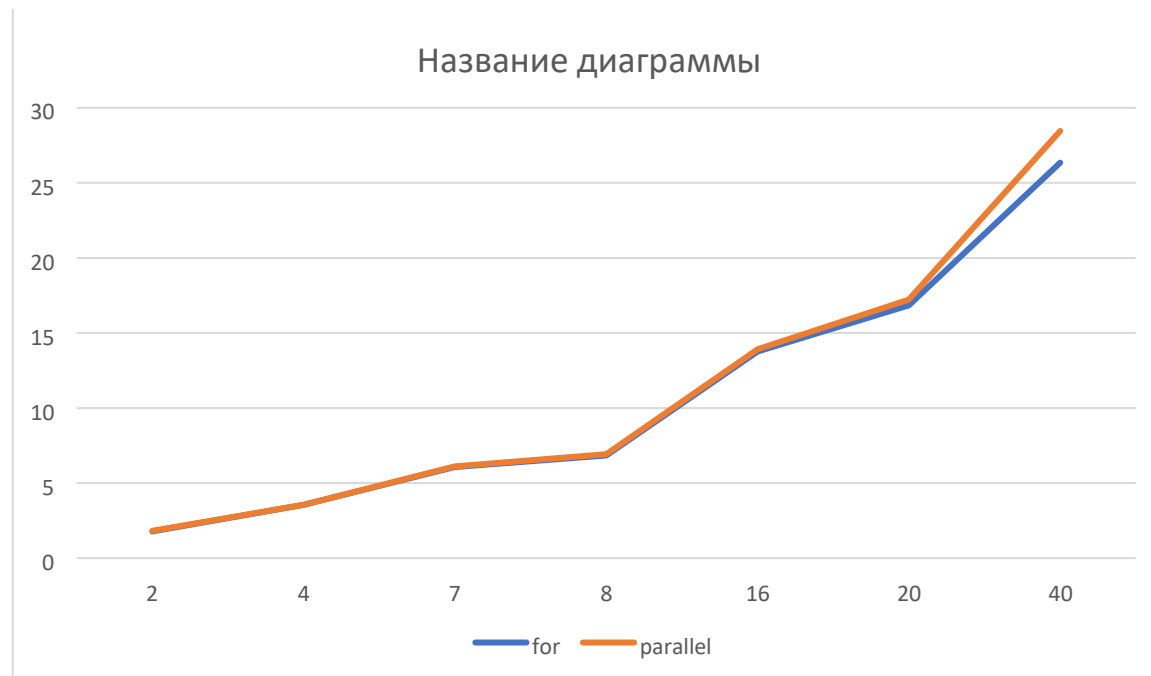
После 16-ти потоков распараллеливание становится незначительным

#pragma omp for

13700	2		4		7		3		16		20		0	
	T	S	T	S	T	S	T	S	T	S	T	S	T	S
	21.98	1.78	11.01	3.56	6.46	6.07	5.75	6.83	2.85	13.78	2.33	16.85	1.49	26.35

#pragma omp parallel

13700	2		4		7		3		16		20		0	
	T	S	T	S	T	S	T	S	T	S	T	S	T	S
	21.50	1.82	11.07	3.55	6.42	6.11	5.67	6.93	2.82	13.92	2.28	17.22	1.38	28.46



#pragma omp parallel работает лучше на больших количествах потоков, но разница незначительна на маленьких

#pragma omp parallel for schedule(dynamic, m/n) num\_threads(n) Дает самое большое ускорение, где  $m$  – размер матрицы,  $n$  – кол-во потоков