List Minimum	Threads	Min	Time	Speedup	Efficiency
1	1	24	0.0980	1.00	100.00%
2	2	24	0.0377	2.60	129.97%
3	4	24	0.0298	3.29	82.21%
4	8	24	0.0240	4.08	51.04%
5	16	24	0.0243	4.03	25.21%
6	32	24	0.0247	3.97	12.40%
7	64	24	0.0242	4.05	6.33%
8	128	24	0.0247	3.97	3.10%
9	256	24	0.0257	3.81	1.49%
10	512	24	0.0268	3.66	0.71%
11	1024	24	0.0340	2.88	0.28%
12	2048	24	0.0613	1.60	0.08%
13	4096	24	0.1230	0.80	0.02%
14	8192	24	0.2442	0.40	0.00%

Barrier		
2	2	0.0002
3	4	0.0003
4	8	0.0004
5	16	0.0006
6	32	0.0009
7	64	0.0018
8	128	0.0036
9	256	0.0070
10	512	0.0144
11	1024	0.0295
12	2048	0.0603
13	4096	0.1238
14	8192	0.2523
15	16384	0.5022

List Statistics	Threads	Mean	Standard Deviaton	Time	Speedup	Efficiency
	1	24.00	0.00	0.0981	1.00	100.00%
	2	31.50	7.50	0.051	1.92	96.18%
	4	46.25	25.25	0.0301	3.26	81.48%
	8	72.13	36.20	0.0245	4.00	50.05%
	16	133.38	94.56	0.0248	3.96	24.72%
	32	250.56	233.01	0.0246	3.99	12.46%
	64	492.69	508.78	0.0251	3.91	6.11%
	128	1283.78	1487.63	0.026	3.77	2.95%
	256	2634.32	2765.05	0.0279	3.52	1.37%
	512	5432.88	5835.93	0.0322	3.05	0.60%
	1024	10951.99	10937.18	0.0428	2.29	0.22%
	2048	21348.19	21569.54	0.0724	1.35	0.07%
	4096	43838.39	44523.82	0.1514	0.65	0.02%
	8192	87238.53	87845.55	0.3053	0.32	0.00%

The reason why execution time increases with the number of threads for the find_minimum function is because the stored minimums for each threaded list must be updated before exiting the find_minimum function. As a result of updating the global minimum, the process becomes serial as it parses through each local minimum.

For the growth of execution time of Barrier with relation to the number of threads is the result of the sleeptime for the work function and the pthread wait command. As the number of threads increase, the numbe of threads in sleep waiting to be woke increases. Only until the last thread has initiated can the function broadcast and resume.