Thread Number	Speed (tok/s)	User Time	System Time	Use Time/System Time
0	47.21	5.41	0.05	108.2
1	41.31	5.33	0.57	9.350877193
2	54.88	5.58	1.24	4.5
4	63.76	5.91	2.2	2.686363636
6	58.65	6.34	3.66	1.732240437
8	51.58	6.79	4.64	1.463362069
10	45.08	6.81	6.3	1.080952381
12	39.74	6.36	8.06	0.7890818859
16	16.61	8.26	15.95	0.5178683386

Analysis:
The relation between the performance and the number of threads seems to be positively correlated until a certain amount. For my machine, it seems like the number would be 6.

The cause of this would possibly be the result of the overhead of context switching. Since the number of threads increases, the OS has to cater to more threads and switch between them more often which results in the CPU spending more time switching than doing actual work.

The initial increase in performance is because there are more idle cores within my machine then there are threads which allows for more efficient allocation between threads and CPU.

Why 4 and 6 seems to be the critical point could be because my machine has 6 cores. Then, why is 6 not the most optimum? It's because this program is not the only thing running on my computer so it's not like all cores are just waiting for my threads.