CS 342- Operating Systems

Project 1 Report

21.03.2023

Instructor: İbrahim Körpeoğlu

Section: 03

Group Members:

Gülin Çetinus - 21902115

Zeynep Selcen Öztunç -21902941

1. Introduction

In this project, we were asked to develop two applications that will find the K most frequently occurring words in a given input data set. We write those words to an output file in the descending order with respect to their occurrences. In part A, we did this via child processes and in part B, we did it by using threads. After implementing both parts, we measured the time it takes to run the programs A and B for various values of N while having a fixed value for K. Then we also measured the time it takes to run the programs A and B for various K values while having a fixed N value. The results can be seen in the Graphs subsection.

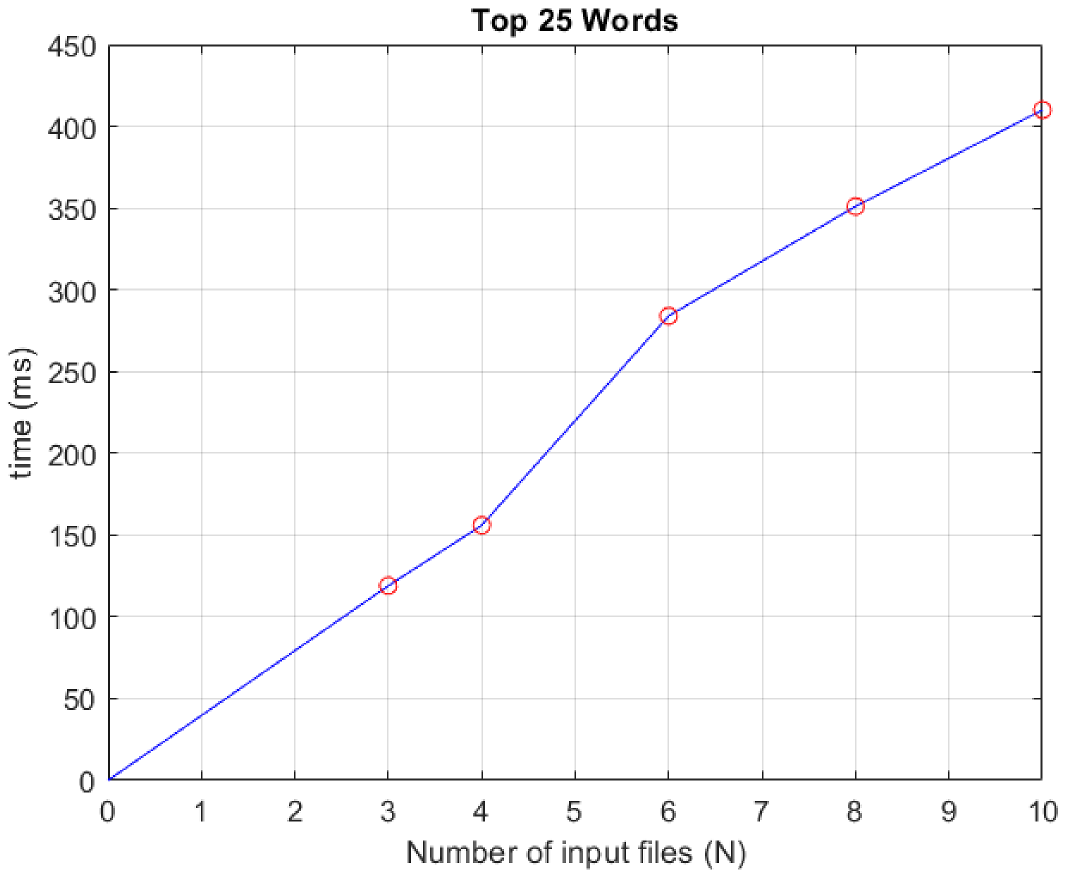
2. Graphs

2.1. Processes' Graphs

Table 1-K value is fixed to 25

N	3	4	6	8	10
Run-time (ms)	119	156	284	351	410

Graph 1-K value is fixed to 25

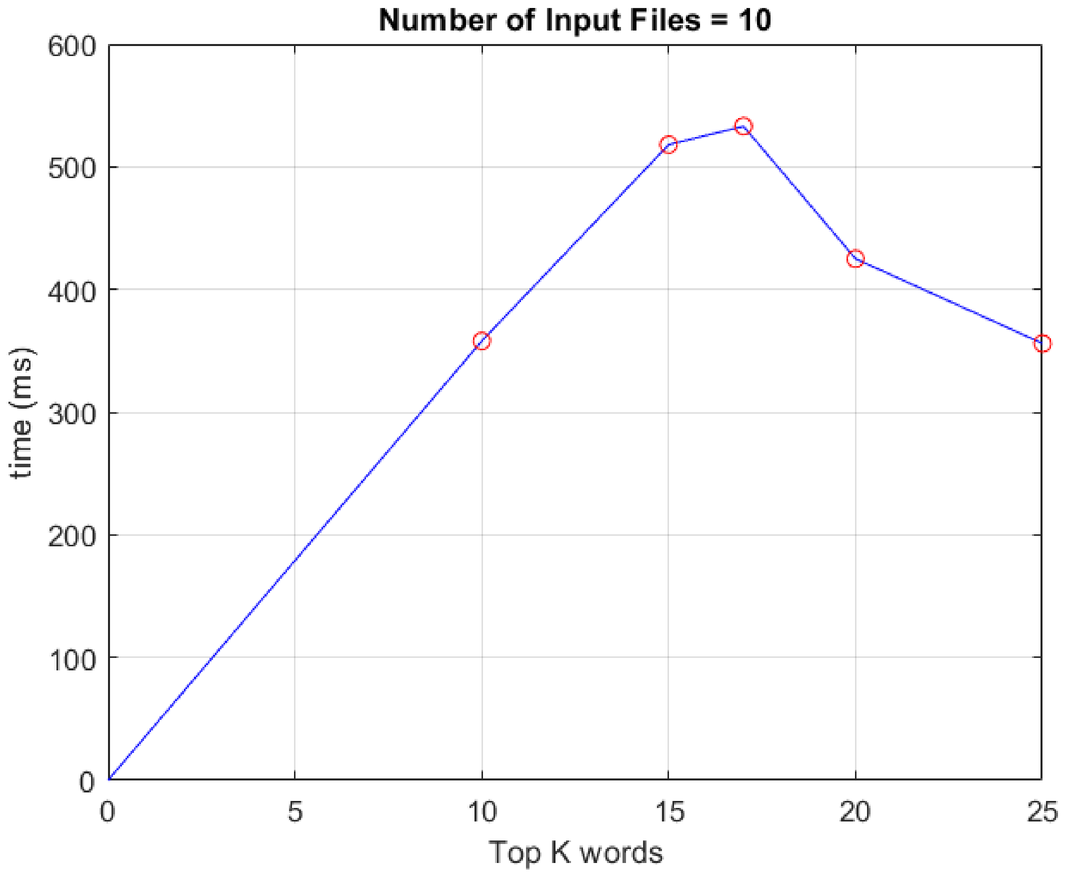


This graph is the expected graph for time-number of input files. In this graph, the time to execute the program increases with decreasing acceleration. Increase in the number of files that the program has to read directly translates into more time to run the code. Hence, the number of input files greatly increases the runtime of the program.

Table 2-N value is fixed to 10

K	10	15	17	20	25
Run-time (ms)	410	586	683	590	431

Graph 2-N value is fixed to 10



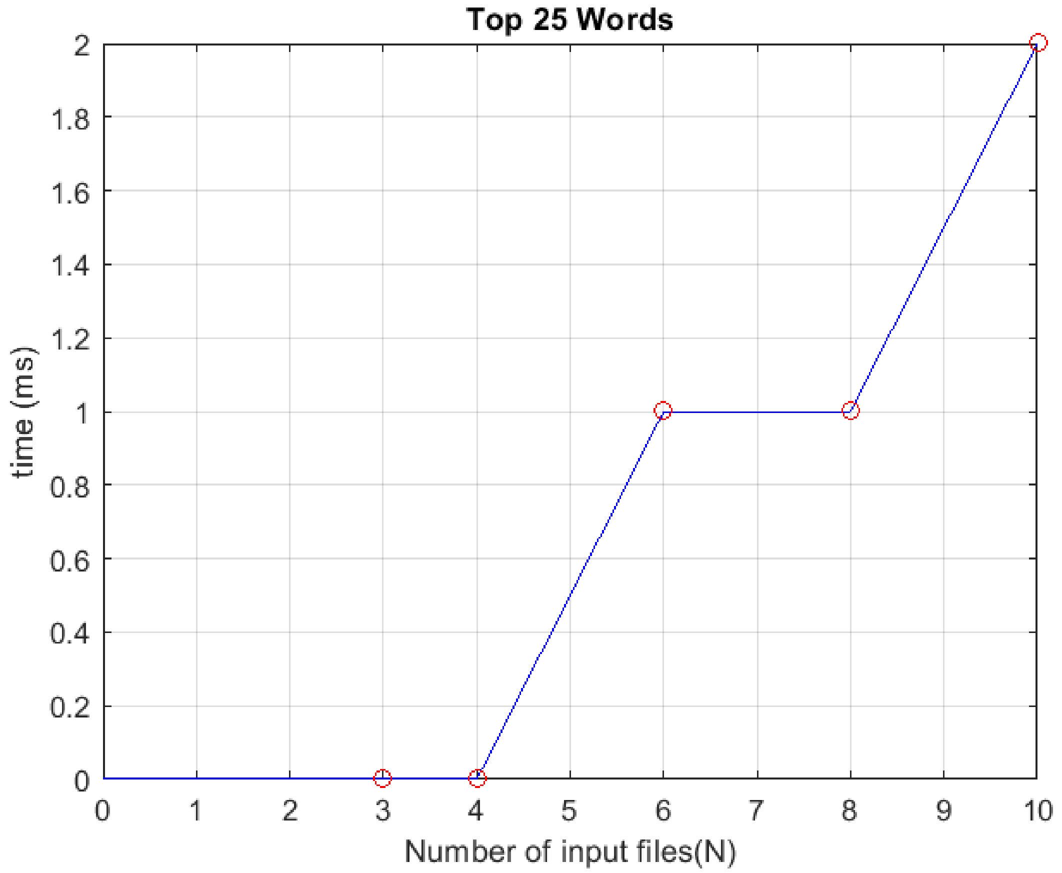
At the values from K=10 to 17, the time to execute the program increases as the words wanted increases. But from N=20 and onwards, the time to execute the program decreases. This decrease might be caused by the background applications' usage of the CPU in the previous values such as K=15 and 17 to cause a strike in the time to execute and this would cause us to incorrectly assume that the time to execute the program decreases, even though this might be a pseudo decrease. We have tried multiple times, but were unable to get a result that is quite similar to the expected result of exponential increase.

2.2. Threads' Graphs

Table 3-K value is fixed to 25

N	3	4	6	8	10
Run-time (ms)	0	0	1	1	2

Graph 3-K value is fixed to 25

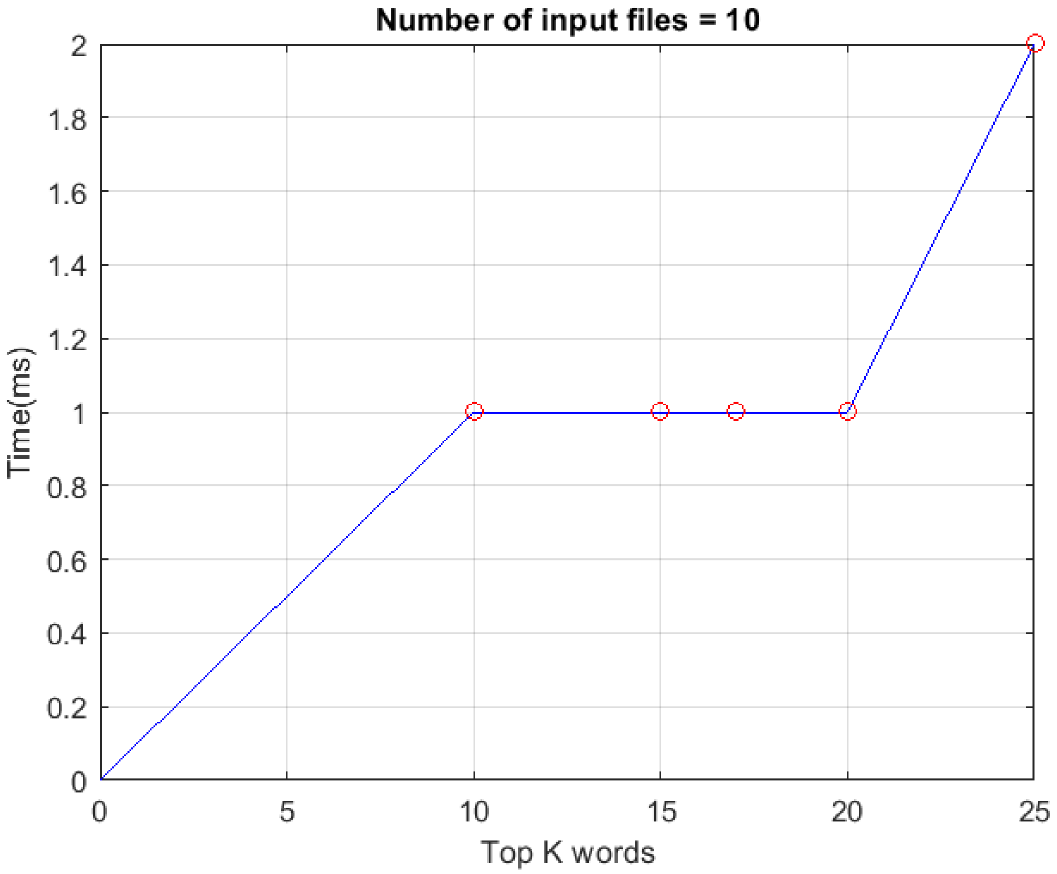


The graph's slope is increasing as the number of input files increases, which is the expected outcome. The time it takes to execute the program is really small.

Table 4-N value is fixed to 10

K	10	15	17	20	25
Run-time (ms)	1	1	1	1	2

Graph 4-N value is fixed to 10



As the number of top words increases the time to execute the program increases as expected. The time to execute is quite small. The time is the same for some intervals.

3. Conclusion

In conclusion, while using processes, the time to execute the program and the number of input files are directly proportional. But when we fix the number of files to a number, we cannot come to a definitive conclusion about the effect of the top K words on time to execute the program. All we can do is to guess what this difference is caused by, and in this case the guess is the background applications creating more work for the CPU to process. Compared to threads, processes take more time to execute. The time to execute a process is around 300 milliseconds, and the time for a thread is around 1 milliseconds. Since the gap is quite huge, this is caused by threads’ approach to handle CPU workload.