Patel Harshank
Dr. Ahmed
UIN: 527009145

Assignment Report

CSCE-313-512

PA4 Threading and Synchronization

Video Demo – Done in Lab to TA Wilson (*pinlyu@tamu.edu* or Pin Lyu)
Video Demo – Done in Lab to TA Wu, Wenxuan as well (just for fun)
Bonus Feature Implemented and Demoed to TA Wilson (*pinlyu@tamu.edu* or Pin Lyu)
(10 EXTRA POINTS MUST BE AWARDED IN THIS ASSIGNMENT)

Data Transfer

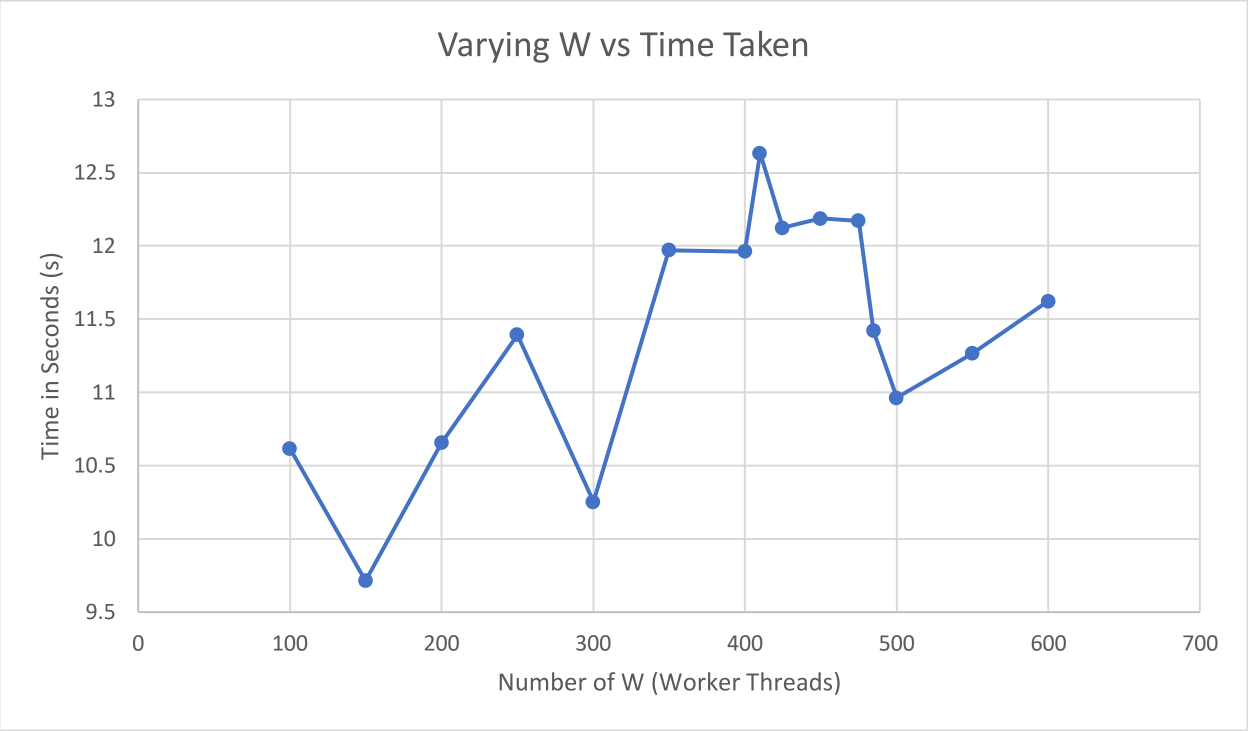
Different W (Worker Threads)
./client -n 15000 -p 15 -h 10 -b 500 -w ___ as given below

W Size	Time Taken (sec)
100	10.616324
150	9.713676
200	10.655598
250	11.392291
300	10.252771
350	11.971111
400	11.962043
410	12.63452
425	12.122761
450	12.18632
475	12.171793
485	11.4214
500	10.963461
550	11.264686
600	11.623812

General Observation for Different W (worker thread):

The worker threads give steads results but for worker threads oof around 410 to 475, it seems to be increasing in time performance and then decreasing in time performance. So, the DIMINISHING point is around 410 worker threads. Having too many worker threads is one of the likely reasons why the worker threads tend to slow down. Having “too many cooks spoil the broth” and having too many worker threads it gives slow runtime.

Graph:



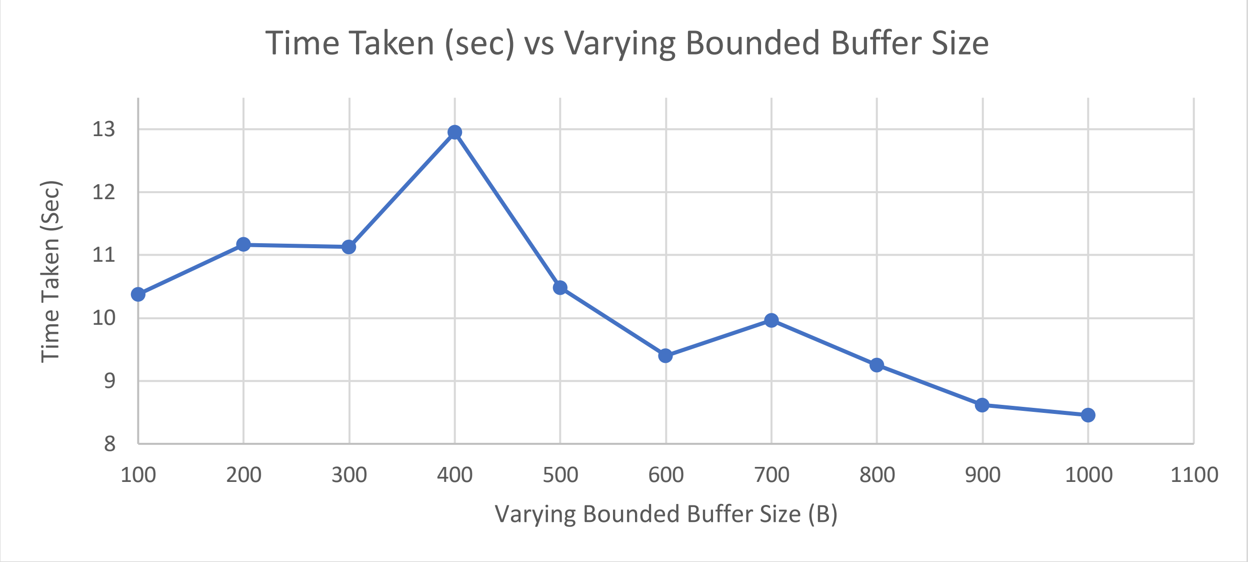
Different B (Bounded Buffer Size)

./client -n 15000 -p 15 -h 10 -w 150 -b ______ where B is:

B Size	Time Taken (sec)
100	10.376423
200	11.163223
300	11.126293
400	12.947962
500	10.478131
600	9.399443
700	9.96236
800	9.249679
900	8.614524

1000	8.456241

General Observation: The changing the Bounded Buffer size doesn't really affect the runtime as per the professor's lecture on April 7, 2021.
Graph:

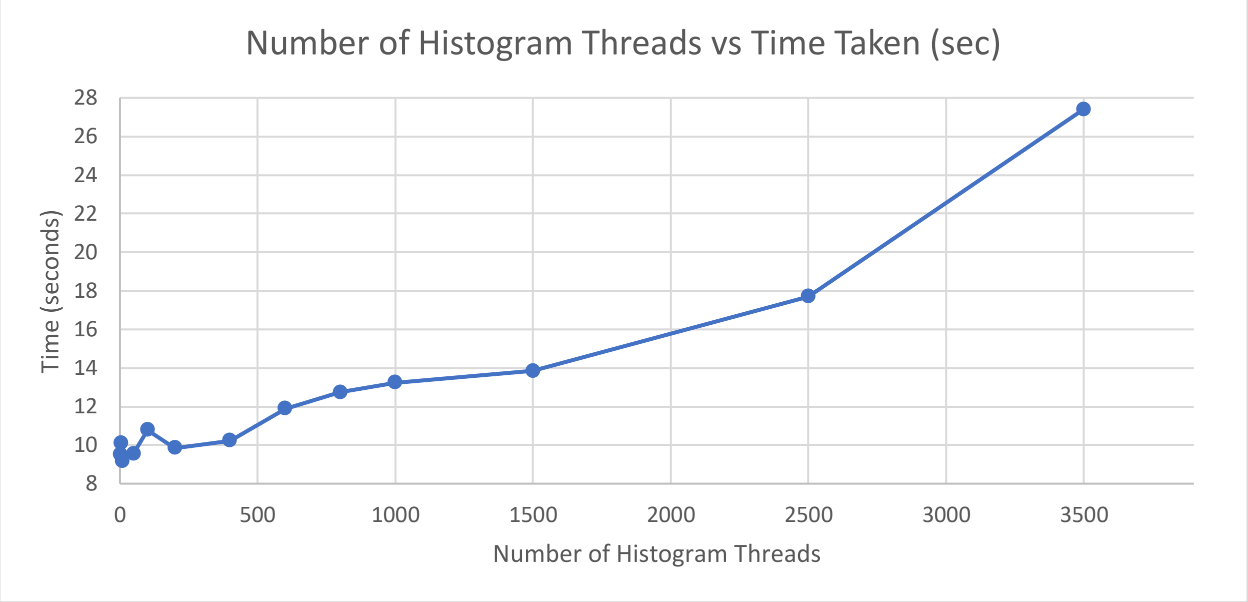


Different H (Histogram Threads)
./client -n 15000 -p 15 -w 100 -b 1024 -h _______ as shown below

Number of Histogram Thread	Time Taken (sec)
1	9.525252
5	10.111841
10	9.172354
50	9.550171
100	10.778639
200	9.846454
400	10.232342
600	11.88454
800	12.73234
1000	13.234
1500	13.834439
2500	17.6939
3500	27.408468

General Observation: In this case, having more Histogram Threads give minimum performance because there is more starvation and therefore it is increasing the time of the program.

Graph:

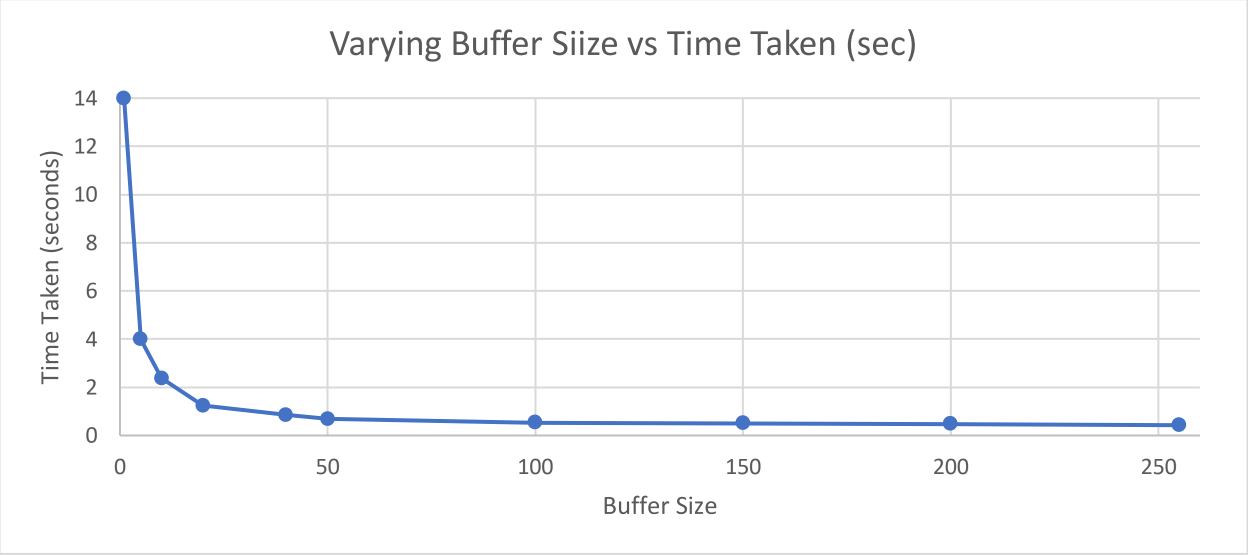


File Transfer

Varying M (Buffer Size)
./client -f 15.csv -w 100 -m _____ as shown

Varying Message Buffer	Time Taken (sec)
1	13.99322
5	3.992423
10	2.378583
20	1.235234
40	0.852232
50	0.691212
100	0.526242
150	0.507809
200	0.483185
255	0.427337

General Observation: The size of buffer cannot be larger than 256 and as show above and from the graph below, the smaller the buffer size, the larger amount of time it takes for you to download a file and having more context switches and more overhead.
Graph:



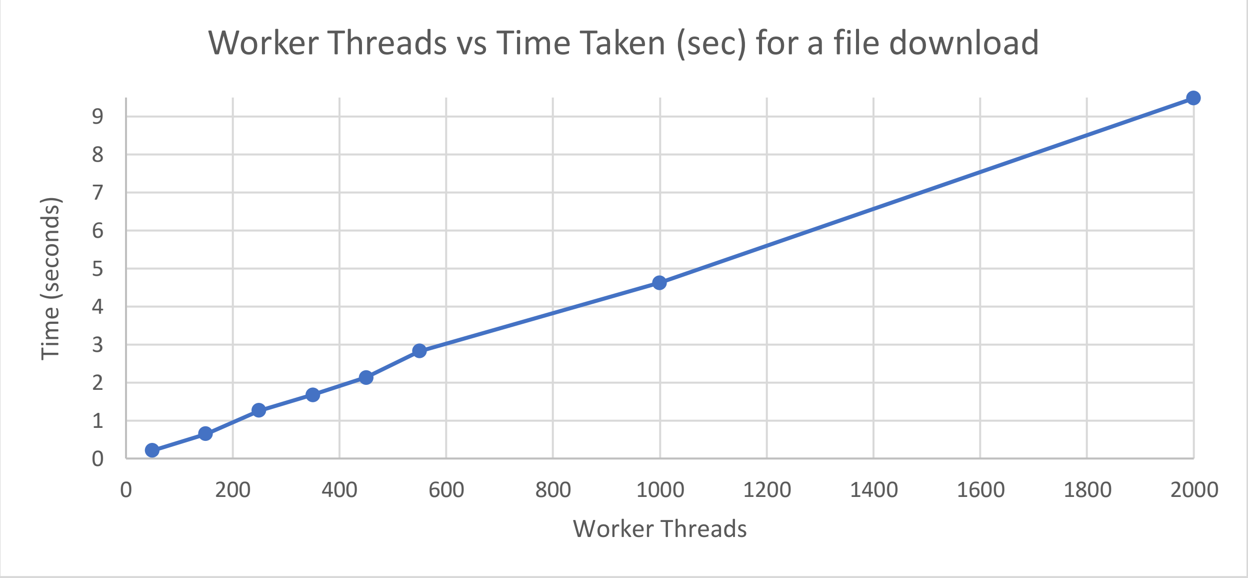
Varying W (Worker Threads)

./client -f 15.csv -m 256 -w ______as shown below

Varying Worker Threads	Time Taken (sec)
50	0.21003
150	0.649575
250	1.265308
350	1.680214
450	2.130762
550	2.830009
1000	4.622764
2000	9.479541

General Observation: As shown from the table above and graph below, the more the number of worker threads, it seems that the runtime is not varying too much with increasing in the worker threads as we are working on the same disk.

Graph:


Worker Threads vs Time Taken (sec) for a file download
Time (seconds)
Worker Threads