

# CSCE 313.506-F18: Programming Assignment 4

Tristan Seifert

Due: Thursday, Nov. 1, 2018

## 1 Performance Data

This data was gathered for varying values of  $w$  while  $n = 10000$ . The system used runs macOS 10.14.1 beta 2, with 64GB of RAM and dual Xeon E5-2670 v2 (10 physical cores, 20 virtual cores) processors at 2.6 GHz.

A maximum of approximately 2000 threads are possible until the file descriptor limit of 4096 was reached. If the program reaches this, calls to create a new channel return with **EMFILE**; the program prints the message **Too many open files** and attempts to continue with the threads that have already been spawned.

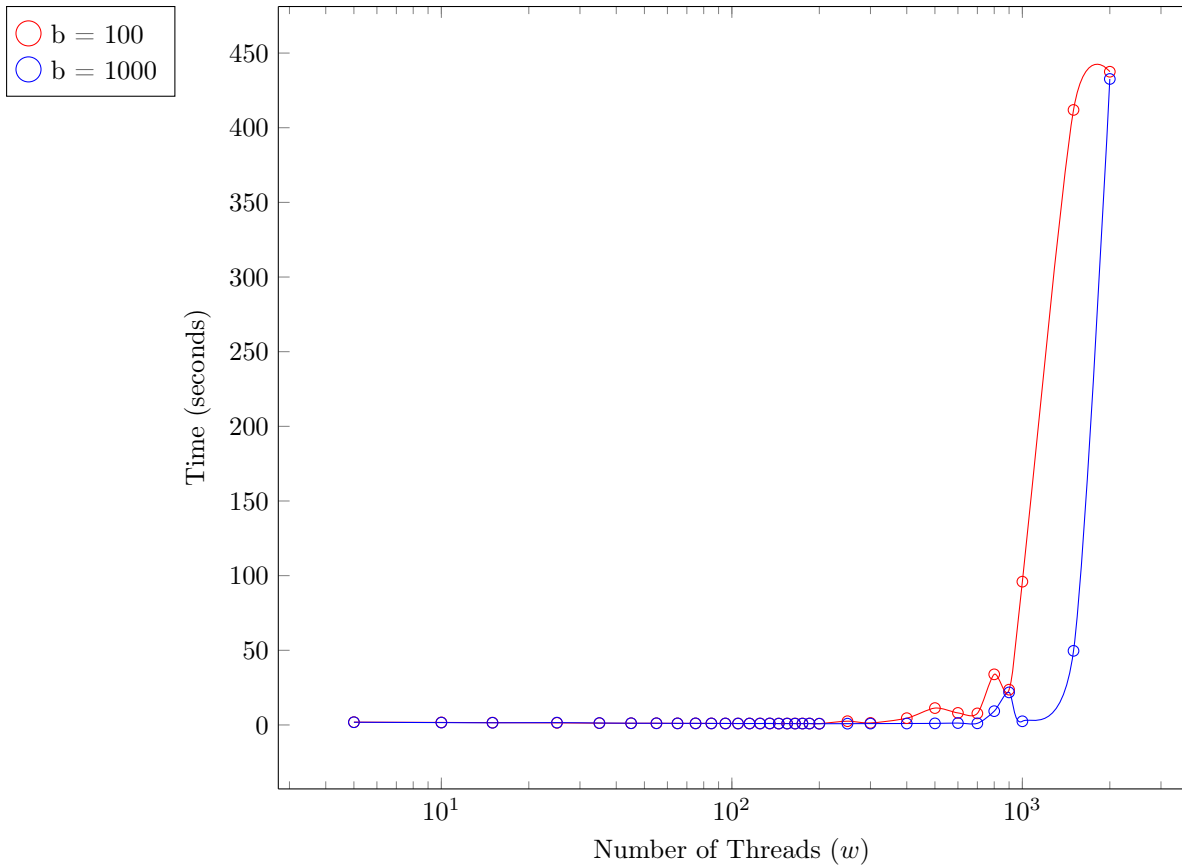


Figure 1: Runtimes for  $n = 10000$

Compared to PA3, this program does run a fair bit slower, which can be attributed to the additional synchronization and error checking in the **BoundedBuffer** class.

Unlike PA3, the optimal number of worker threads appears to be around 105-115, regardless of the buffer size. This is roughly 5x more than PA3, but can be explained because the threads will be waiting for data to arrive on the buffer instead of all running simultaneously. Better performance can possibly be attained by tweaking the size of the request buffer size in comparison to the total number of requests and worker threads.

Worker Threads	b = 100	b = 1000
5	1.87483	1.82058
10	1.69082	1.58102
15	1.55342	1.45176
25	1.40466	1.56471
35	1.20532	1.28997
45	1.13468	1.16334
55	1.13611	1.12449
65	1.06531	1.05702
75	1.03595	1.07653
85	1.01859	1.02945
95	0.99349	0.997434
105	0.96444	0.970505
115	0.98367	0.964283
125	0.97590	0.964424
135	0.95225	1.02618
145	0.93276	0.919301
155	0.96534	0.925374
165	0.97023	0.902376
175	0.96852	0.914494
185	0.96476	0.937188
200	0.87838	0.8306
250	2.50741	0.86805
300	1.35759	0.916199
400	4.53737	0.96884
500	11.30150	1.04914
600	8.06313	1.26772
700	7.74134	1.09808
800	33.90010	9.24121
900	23.63470	21.8044
1000	95.96800	2.46151
1500	411.91800	49.6417
2000	437.50200	432.654