Luke Grammer

CSCE-313

4/18/19

Programming Assignment 5 Analysis

Analysis

Maximum number of workers

FIFO:
The maximum number of workers possible with my FIFO implementation is 500 without raising the ulimit for open files in the operating system. Therefore the maximum number of FIFO queues open simultaneously is 1000 without modifying this limit. This is the baseline for PA4, although for the previous project, temporary modifications to the ulimit were made in order to go significantly beyond 500 simultaneous workers.

SHM:
The maximum number of workers possible with my shared memory object implementation is 460. This means that the maximum number of shared memory objects that can be open simultaneously is 920. This is slightly lower than the FIFO limit, although it opens significantly more file handles for kernel semaphores as well as shared memory segments.

MQ:
The maximum number of workers that can be active simultaneously with my message queue implementation is 110. This is significantly lower than the maximum number of FIFO channels or SHM channels, and this is the lowest number of workers out of the three implementations. This means that there can be approximately 220 message queues open simultaneously.

Limits

FIFO:
The limit for the FIFO channel is due to the open file handle user limit imposed by the operating system. Attempting to go beyond this limit results in a ‘too many open files’ error and terminates the program.

SHM:
The limit for the SHM channel is likely due to the available space for shared memory allocation. Attempting to go beyond the worker thread limit results in a segmentation fault and terminates the program.

MQ:
Similarly to the FIFO channel, the limit for the Message Queue channel may be due to an open file handle limit imposed by the operating system for IPC objects. Attempting to go beyond this limit results in a ‘bad file descriptor’ error and terminates the program.

Cleanup Activities:
FIFO:

The FIFO channel object is destroyed by first closing the reading pipe and the writing pipe using close(), and then removing the pipes using remove().

SHM:

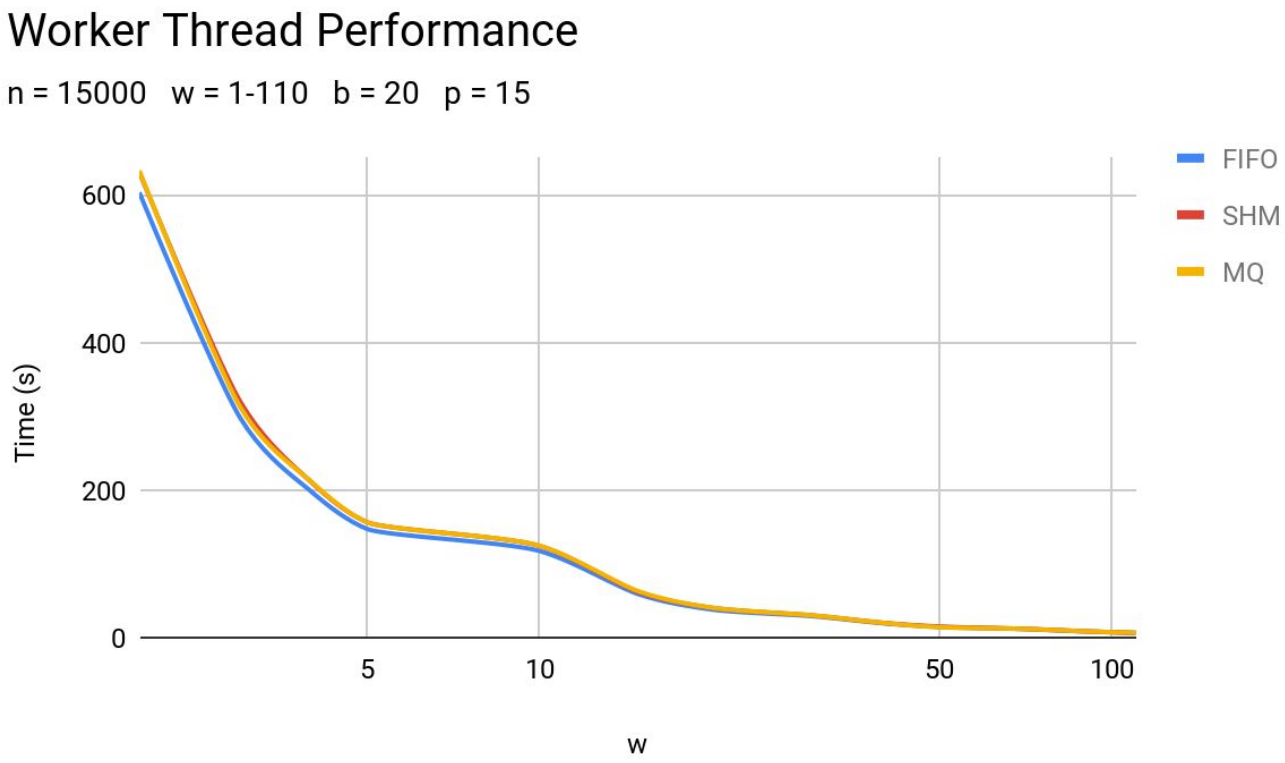
The shared memory channels have two shared memory buffer objects. Each bounded buffer object also uses two kernel semaphore objects. When a kernel semaphore is destroyed, the named semaphore is closed and then unlinked using sem_unlink(). When the buffer is destroyed, it deletes it’s semaphores, closes the file descriptor associated with it’s shared memory and then unmaps it’s shared memory using munmap() and unlinks it using shm_unlink(). When the request channel is deleted it simply deletes each of its buffers.

MQ:

The message queue channel is closed in almost exactly the same way as the FIFO channel. First the reading and writing file descriptors are closed, and then the queue names are unlinked using mq_unlink().

Results

Figure 1:



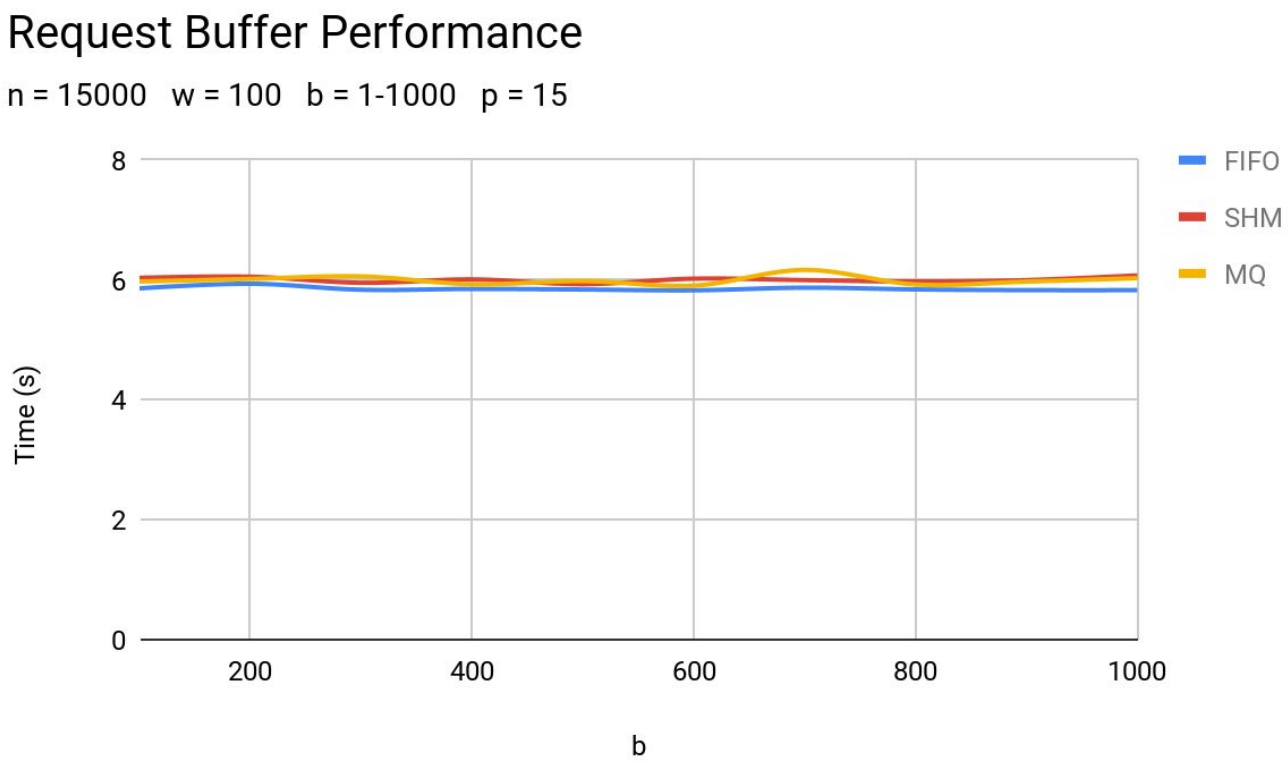
Explanation:

There is very little difference in the running times of the different implementations for varying numbers of worker threads. This indicates that each method of IPC has a very similar read/write speed. The FIFO queue is slightly faster than the other implementations, but only by a small constant factor. This difference is only appreciable at very small numbers of worker threads. The graph is monotonically decreasing, which as expected indicates that a larger number of worker threads will make the program run faster. There is a difference in performance, however, at approximately 10 workers. This may be due to the operating system's scheduler reducing the priority of the threads after they have been running for a certain amount of time. This makes the program run even faster with a larger number of worker threads because they all complete before the scheduler reduces their priority.

Table 1:

Number of worker threads	Time (FIFO)	Time (SHM)	Time (MQ)
1	603.365192	631.125362	632.931769
2	297.981055	320.097225	312.935438
3	197.495576	210.309041	210.834525
4	147.281532	156.330125	156.247306
5	117.465736	124.406594	124.647586
10	58.62617	60.853194	61.960914
15	38.66184	40.438052	40.520939
20	29.43142	30.079594	30.061244
30	19.318621	19.943638	19.834114
40	14.500476	14.885089	14.084077
50	11.635117	11.941675	11.894674
70	8.372896	8.562779	8.498681
90	6.582618	6.18252	6.622202
110	5.494446	5.451639	5.352593

Figure 2:



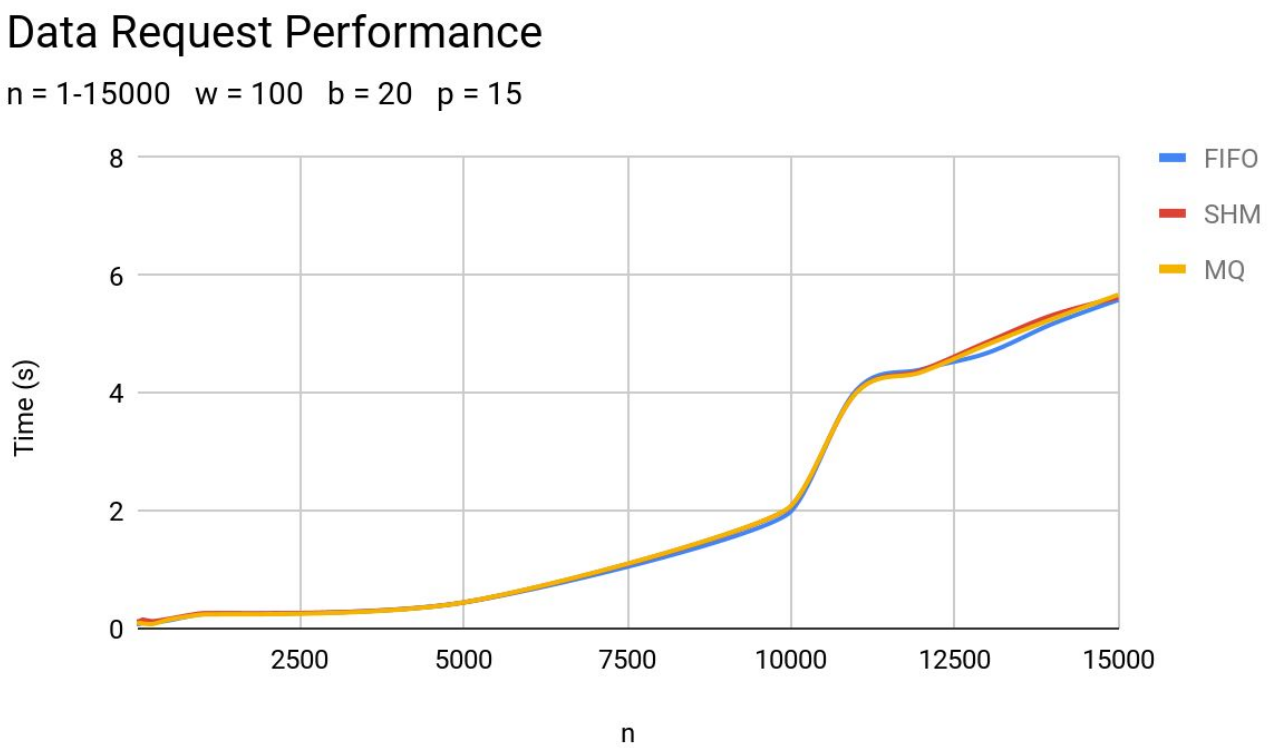
Explanation:

Much like in programming assignment 4, request buffer size makes very little impact on implementation performance since there are very few production channels in relation to the number of consumption channels. Therefore, the request buffer will likely be empty most of the time.

Table 2:

Bounded Buffer Size	Time (FIFO)	Time (SHM)	Time (MQ)
1	5.859943	6.035796	5.975077
100	5.93779	6.053942	6.017143
200	5.837124	5.953175	6.059706
300	5.850591	6.009314	5.928827
400	5.840653	5.93033	5.98208
500	5.827669	6.021906	5.90612
600	5.871481	5.99773	6.168209
700	5.842259	5.974043	5.925511
800	5.830272	5.996941	5.972736
900	5.829795	6.076051	6.030108
1000	5.946276	6.047814	5.948733

Figure 3:



Explanation:

Again, there is very little difference between the time of the different implementations. While the FIFO queue implementation may perform slightly faster by a small constant factor, the difference is not appreciable. Though we do see approximately linear scaling in a large portion of the graph, there is a large overhead between 10000 - 12500 data point requests. Much like the first graph, this may be due to the operating systems scheduling system reducing the priority of the workers after they have been running for several seconds.

Table 3:

Number of datapoint requests	Time (FIFO)	Time (SHM)	Time (MQ)
1	0.076084	0.118855	0.084289
10	0.07815	0.125996	0.093786
50	0.09418	0.150379	0.085704
100	0.097729	0.124665	0.076566
250	0.139017	0.16484	0.150569
500	0.243161	0.25877	0.237316
1000	0.442402	0.443285	0.443356
5000	1.989241	2.076477	2.081806
10000	4.051209	4.013169	4.009666
11000	4.404023	4.388112	4.358536
12000	4.682301	4.867925	4.823785
13000	5.177915	5.32641	5.261901
14000	5.582268	5.623195	5.670857
15000	5.990436	6.138342	6.035371