

I. Rajasekhar Reddy - CS20BTECH11020

Programming Assignment-5

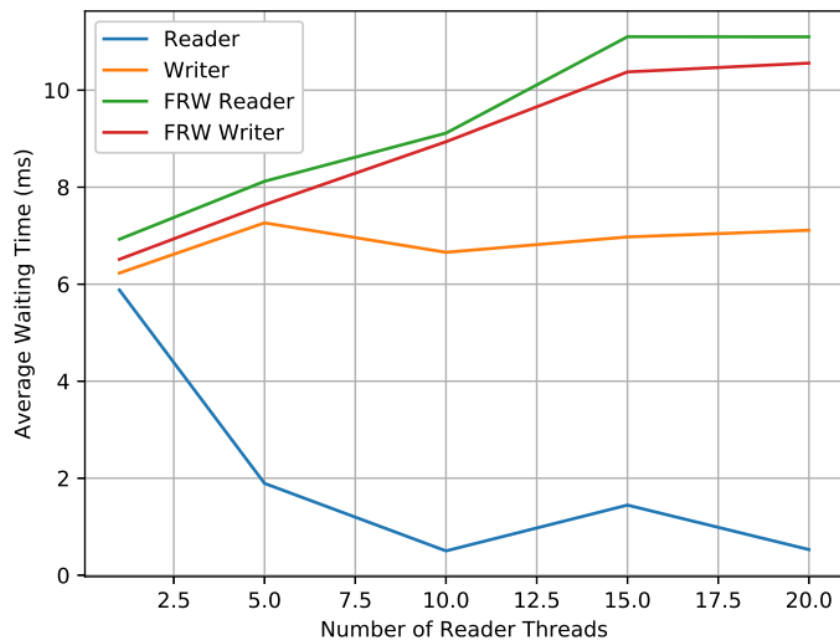
Both the algorithms i.e., Readers-Writers and Fair Readers-Writers have been implemented using Semaphores from the header semaphore.h.

Semaphores have been implemented using the header semaphore.h

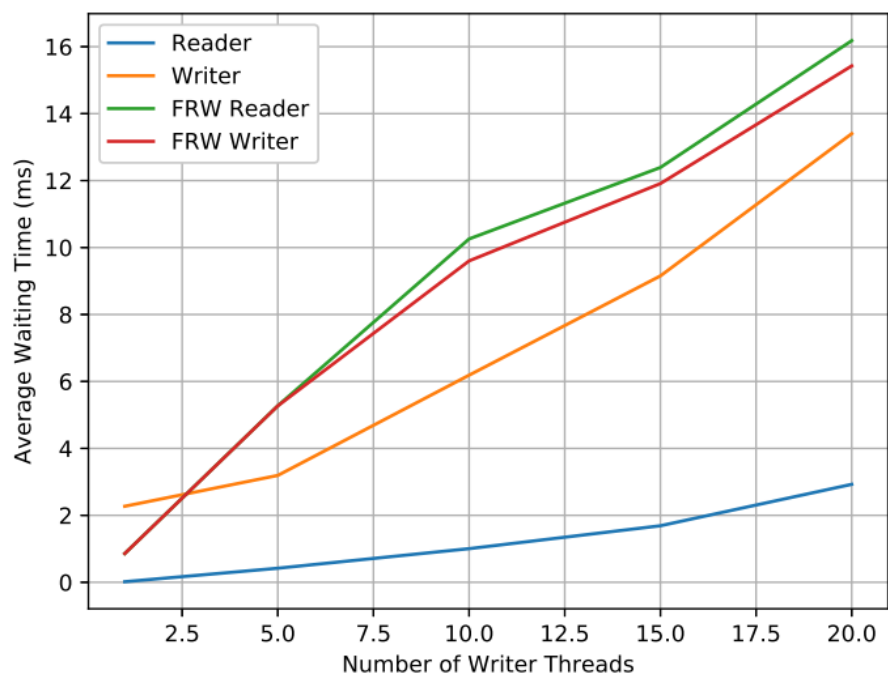
1. Readers-Writers is solved using the two semaphores rwm, mut and an integer count_r.
2. The semaphore mut is used to avoid race condition when accessing the variable count_r.
3. The variable count_r is used to keep a count on the number of readers, when the number of readers is ≥ 1 , then the semaphore rwm makes sure that no writer can access its critical section.
4. Once the number of readers is 0, the semaphore rwm is signalled which gives a chance to the writers to enter their critical section, all this is accomplished by putting conditions on the variable read count every time a new reader attempts to enter its critical section.

RESULTS AND GRAPHS:

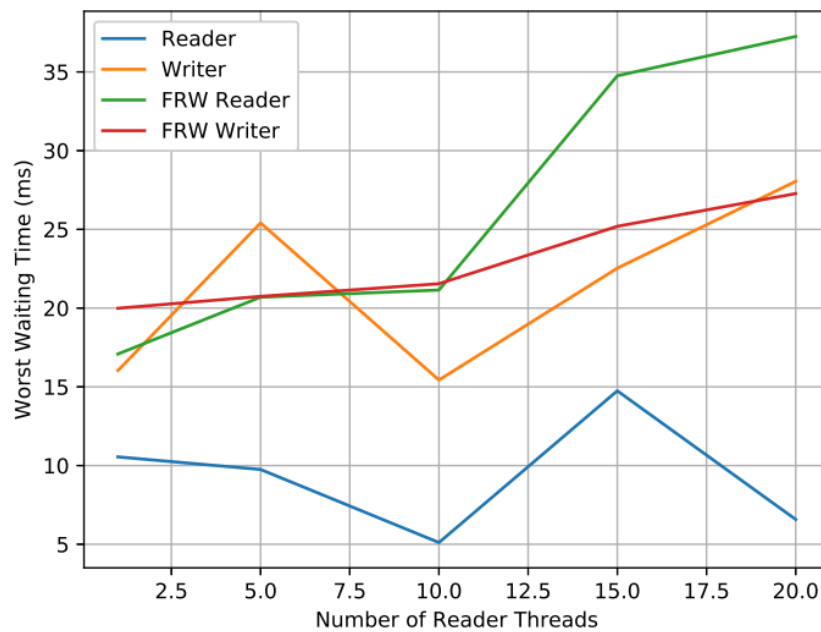
Average Waiting Times with Constant Writers



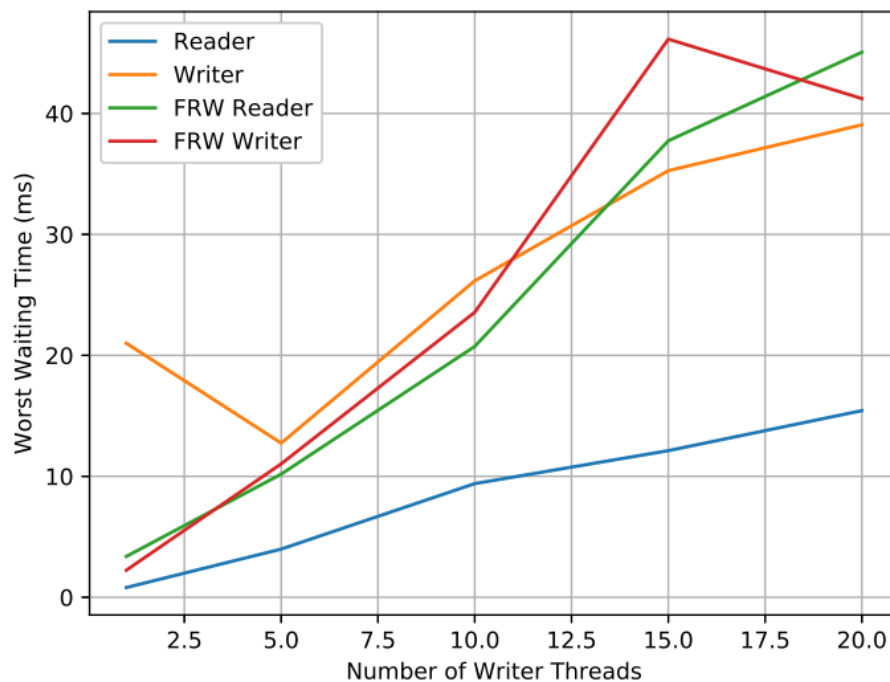
Average Waiting Times with Constant Readers



Worst-case Waiting Times with Constant Writers



Worst-case Waiting Times with Constant Readers



1. Average Waiting Times with Constant Writers

(a) As the number of reader threads increases, all curves show an increase in average wait time, but not reader threads for common readers. In general, the latency of the Fair Readers-Writers algorithm is longer than that of the Readers-Writers algorithm because it contains more instructions.

(b) The difference in waiting time of reader and writer threads for Fair Readers writer is almost negligible compared to Normal Readers writer, which illustrates the effectiveness of FRW algorithm and writer starvation in RW algorithm.

2. Average Waiting Times with Constant Readers

(a) This graph also tells pretty much the same story, except the fact that all the curves start off from a common low starting point, this is not unusual since we started off with a small number of writer threads and these are the threads that contribute most to the waiting times through a convoy effect.

(b) Once again, the divergence in the waiting times of reader and writer threads of the normal RW algorithm is clearly visible. The writers starve often. The fair readers writers show a smooth increase in the waiting times but at the same time have very low difference in the waiting times of the reader and writer threads.