Assignment 4

Implement solutions to Readers-Writers and Fair Readers-Writers problems using Semaphores

Name : Shivashish Suman
Roll No. : CS17BTECH11037

Design of Program:

In this assignment, we have implemented the reader-writer solution starving the writers and fair reader-writer solution.

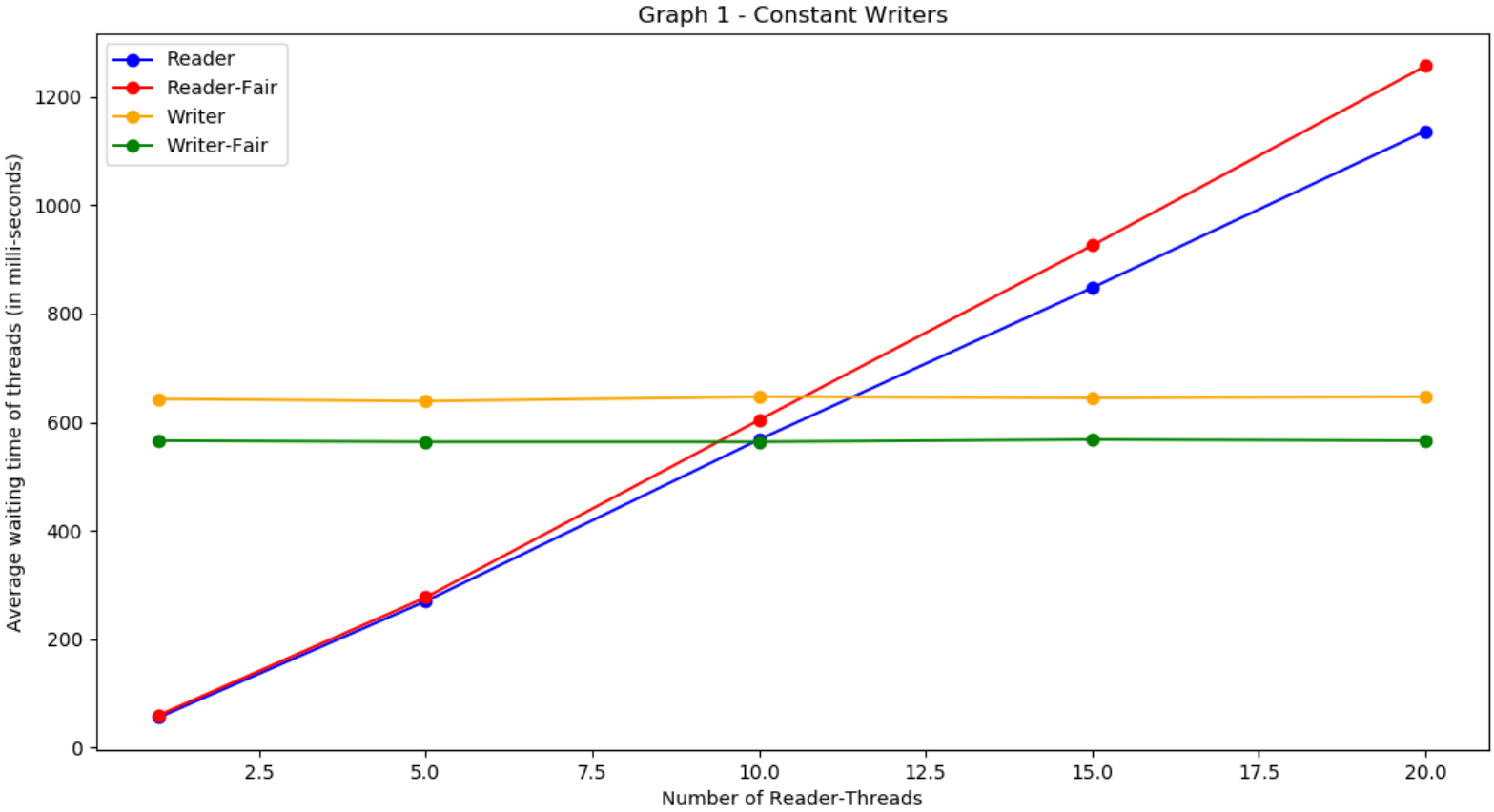
In the reader-writer solution starving the writers, we have used two semaphores ‘rw_mutex’ and ‘mutex’. Semaphore ‘rw_mutex’ functions as a mutual exclusion semaphore for the writers. Semaphore ‘mutex’ is used to ensure mutual exclusion when the variable read_count is updated.

In the fair reader-writer solution, we have used three semaphores ‘in’, ‘queue’ and ‘mutex’ . Semaphore ‘in’ controls access (read/write) to the resource . Semaphore ‘queue’ preserves ordering of requests (signaling must be FIFO) which ensures fairness. Semaphore ‘mutex’ is used for syncing changes to shared variable ‘readCount’. Variable ‘readCount’ maintains the count of readers currently accessing the resource.

In both the programs, variables ‘wait_reader’ and ‘wait_writer’ stores the total wait time to enter the CS of reader and writer threads respectively. Variables ‘worst_reader’ and ‘worst_writer’

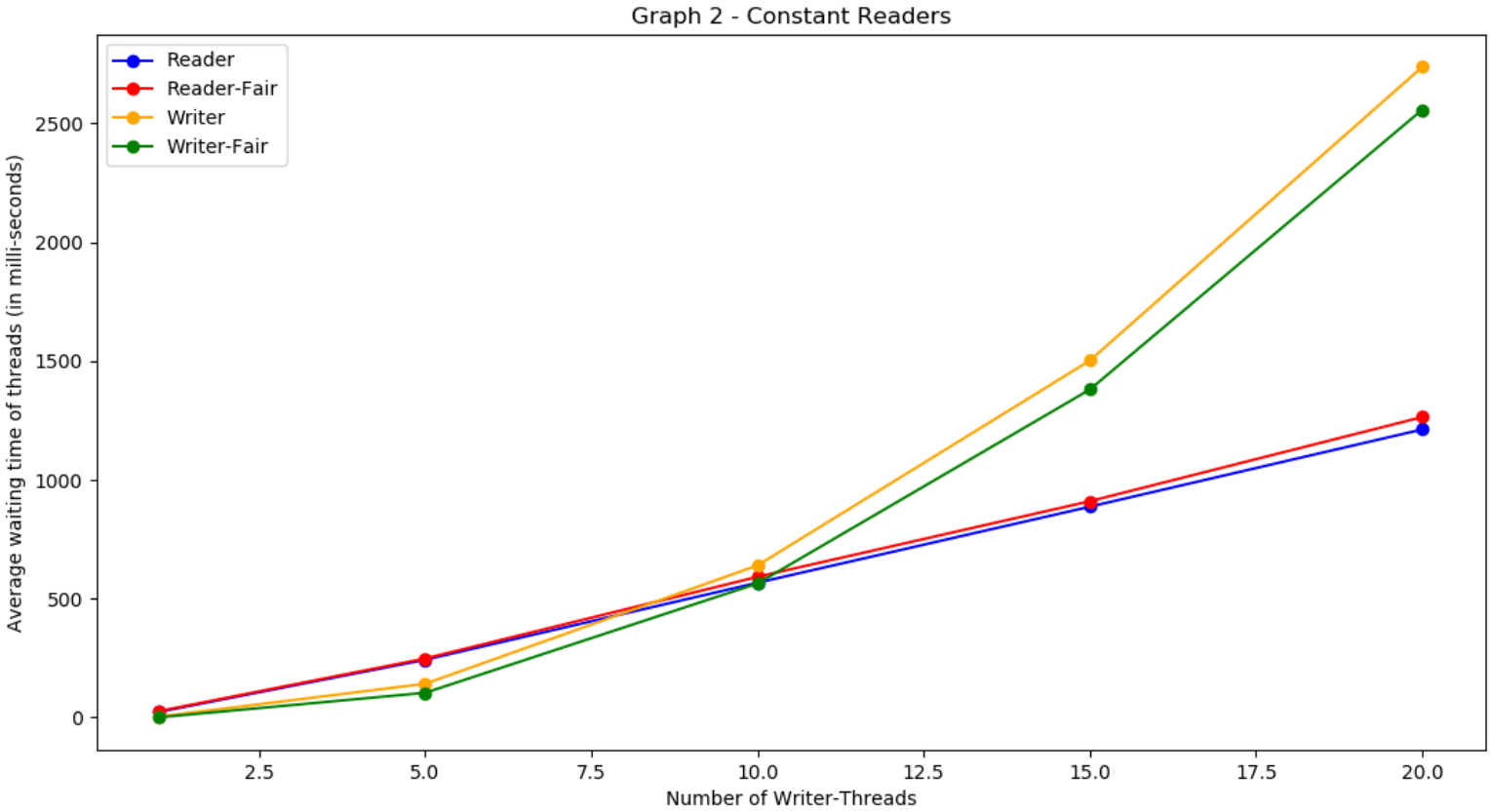
stores the worst case waiting time for reader and writer threads respectively.

Graph 1:
Average waiting time of threads with constant writer threads



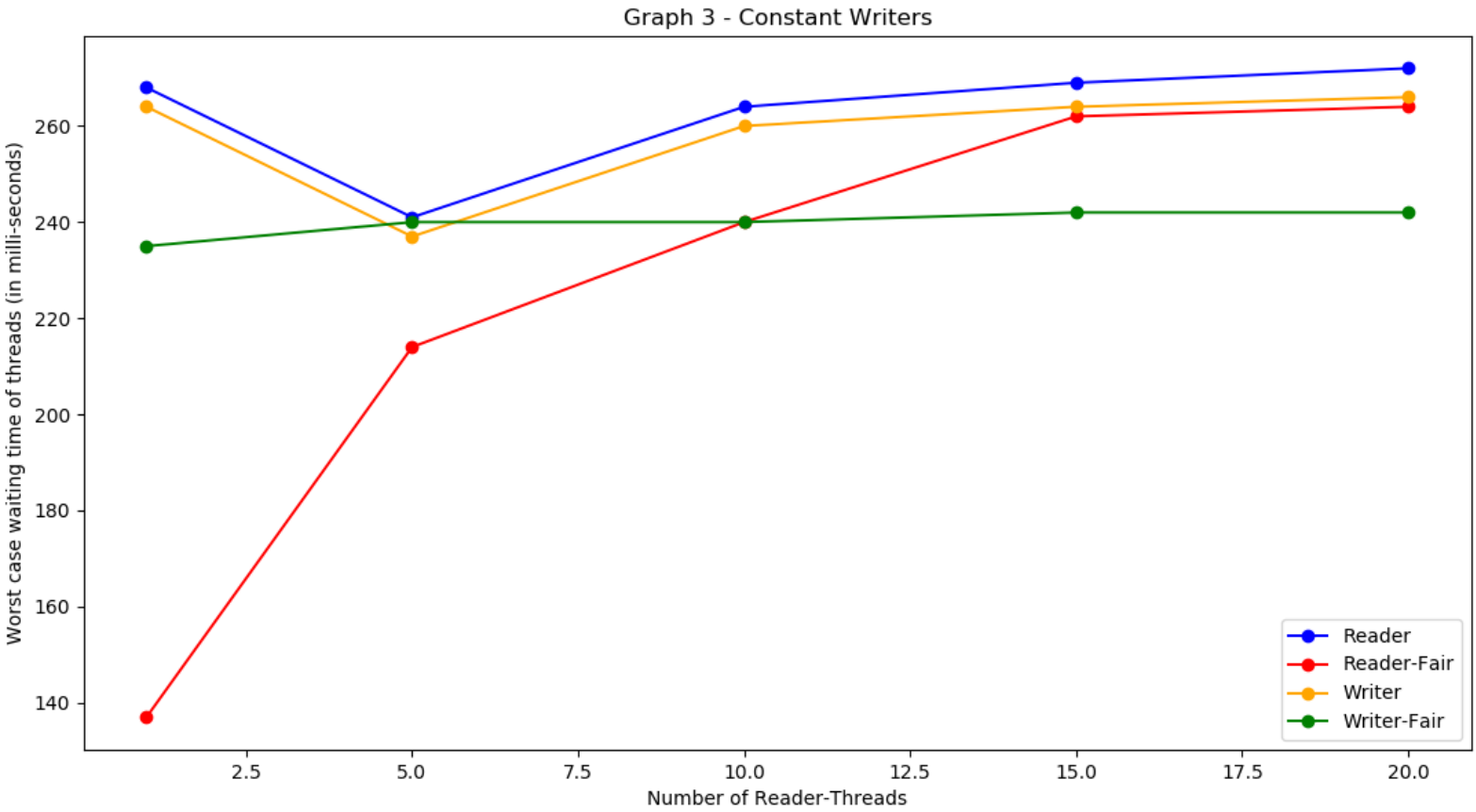
In this graph, we observe that average waiting time for readers is better in reader-writer solution as compared to fair reader-writer solution. Average waiting time for writers is almost the same in reader-writer solution as well as fair reader-writer solution

Graph 2:
Average waiting time of threads with constant reader threads



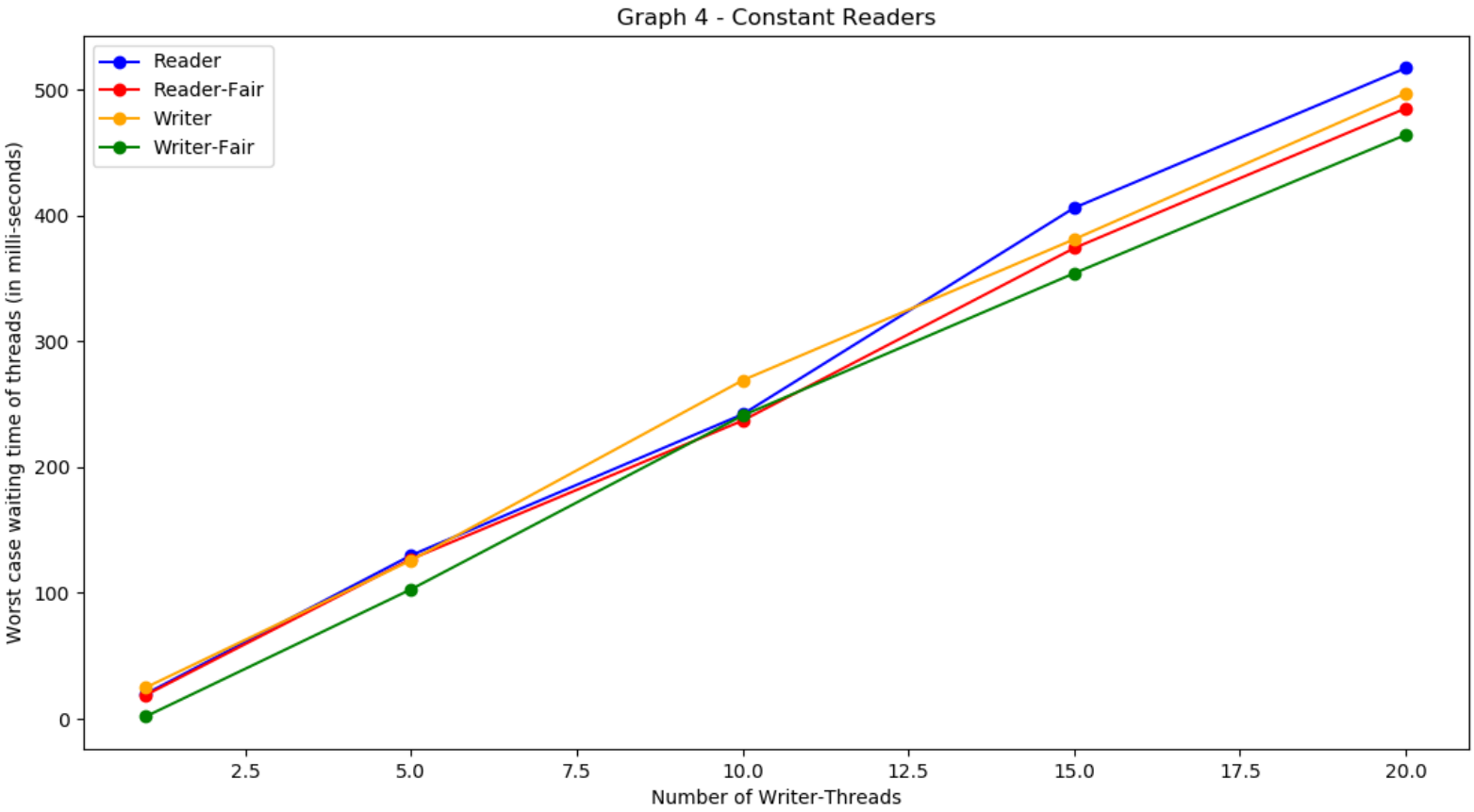
In this graph, we observe that average waiting time for readers is better in reader-writer solution as compared to fair reader-writer solution. Average waiting time for writers is almost the same in reader-writer solution as well as fair reader-writer solution.

Graph 3:
Worst Case Waiting Time of Threads with constant writer threads



In this graph, we observe that worst case waiting time for readers is better in fair reader-writer solution as compared to reader-writer solution.Also, worst case waiting time for writers is better in fair reader-writer solution as compared to reader-writer solution.

Graph 4:
Worst Case Waiting Time of Threads with constant reader threads



In this graph, we observe that worst case waiting time for readers is better in fair reader-writer solution as compared to reader-writer solution.Also, worst case waiting time for writers is better in fair reader-writer solution as compared to reader-writer solution.

Average Waiting Time:

Reader-Writer < Fair Reader-Writer (Reader Threads)

Reader-Writer ≈ Fair Reader-Writer (Writer Threads)

Worst Case Waiting Time:

Reader-Writer > Fair Reader-Writer

(Reader and Writer Threads)

As Fair Reader-Writer Solution gives fair chance to every thread so its worst case waiting time is low.