Programming Assignment 2

Implementing TAS, CAS and Bounded Waiting CAS Mutual Exclusion Algorithms

Name : Shivashish Suman
Roll No. : CS17BTECH11037

In this assignment we have implemented TAS,CAS and CAS with Bounded Waiting Time Algorithms.

In this assignment,the design pattern is almost same for all the three program. We have defined a variable atomic<int> lock / atomic_flag lock which ensures mutual exclusion by allowing only one thread to enter in critical section. First we take input from “inp-params.txt” and then store the input in global variable which is shared among all the threads.Then we have created n pthreads and called the testCS function with parameter as thread-id.

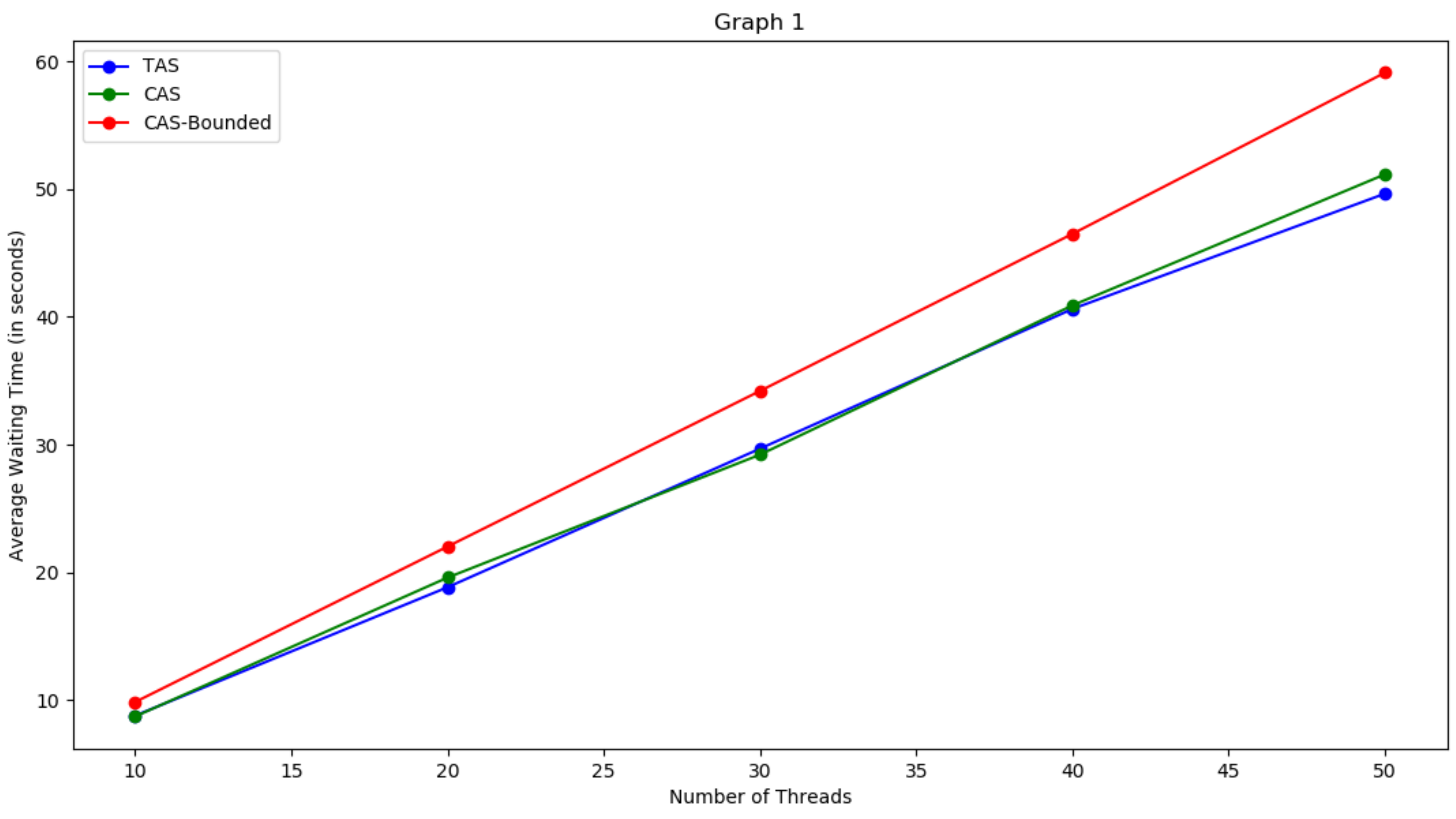
In TAS program, we have used “atomic_flag_test_and_set_explicit” which atomically sets the lock to be true and returns the previous value of lock.We have used ”atomic_flag_clear_explicit” to atomically change the state of lock to be false so that other threads waiting to enter critical section can now enter their critical section.We note the time at which thread makes a request to enter the critical section and time at which it enters. The difference of both the times give us the waiting time.
Then we have used exponential distribution to find sleep times in critical section and remainder section using the given average time.

In the CAS program, we have used “comapare_exchange_strong” which compares the lock value with expected value i.e. x. If both are same, then it replaces the lock value with y, else it replaces expected value i.e. x with y. So everytime we have initialised x=0 and y=1. We break from waiting loop if lock = 0 and it makes lock = 1 so that no other thread can enter the critical section and after it exits its critical section then we make lock = 0.

In the CAS with bounded waiting program, we have created ‘waiting’ array of n bool variables denoting whether the i-th thread is waiting to enter the critical section or not. After a thread finishes it finds a thread who is willing to enter the critical section and makes its waiting value to be false. Rest part is same as in CAS.

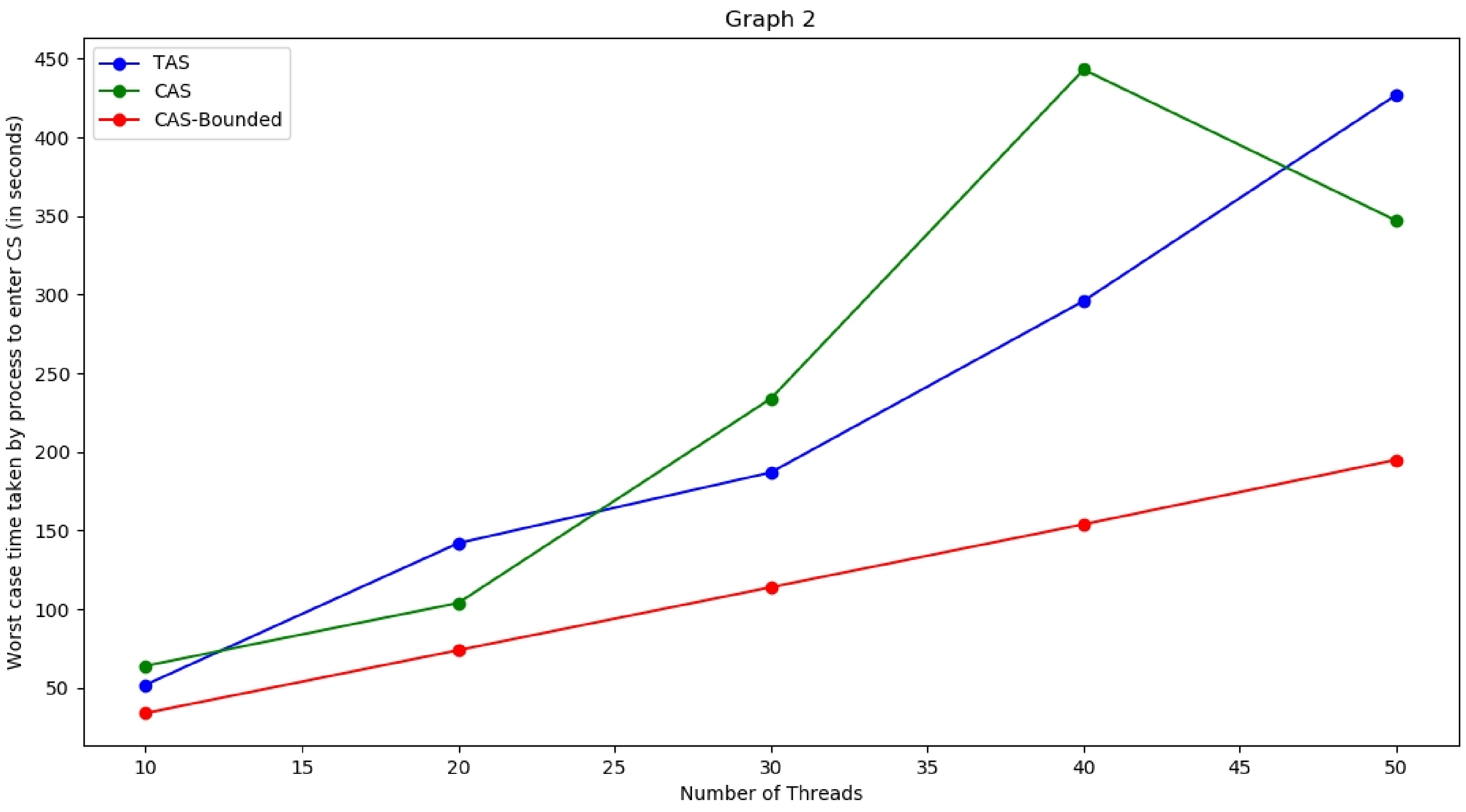
Graph :

(i) Average Waiting Time vs Number of Threads



In this graph, we observe that average waiting time for TAS and CAS is nearly same but that for CAS with Bounded Waiting time is much higher. Although CAS-Bounded gives every thread fair chance to enter critical section but it has much higher waiting time.

(ii) Worst Case Time taken by Process to enter CS vs Number of Threads



In this graph, we observe that worst case time for a process to enter CS is much lower in CAS-Bounded as compared to CAS and TAS as it gives fair chance to every thread to enter CS.