

OS 2 - Programming Assignment 3

Implementing TAS, CAS and Bounded Waiting
CAS Mutual Exclusion Algorithms

CS21BTECH11051
Rajiv Shailesh Chitale

This report compares the implementation of mutual exclusion with TAS, CAS and bounded waiting with CAS. An overview of the implementation is as follows:

The input() function reads parameters from an input file. n is the number of threads that enter the critical section, doing so k times each. The time spent in the critical section and remainder section are given by exponential distributions with average values l1 and l2. These samples are generated using the function genExpVariable().

The main function uses cpp thread library to create threads to run the function testCS().

The testCS() function takes note of the time when requesting, entering and exiting from the critical section with the help of the getSysTime() function which internally uses ctime library. It also measures waiting time.

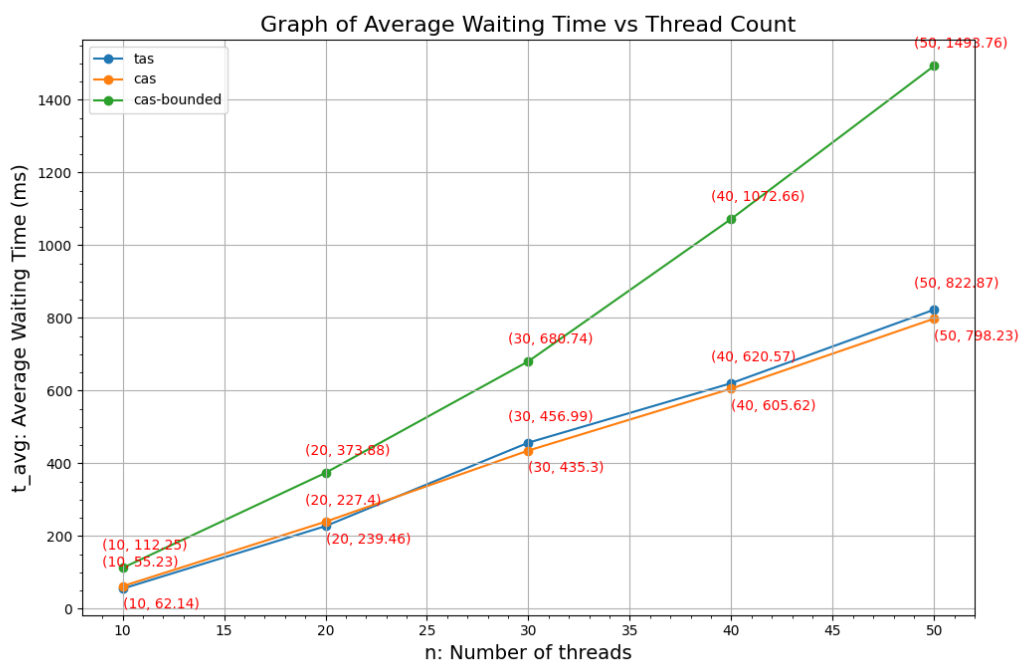
The output() function is called to write these times for each thread and iteration into a file. The waiting time. The logPerformance() function keeps track of maximum and total waiting time (used to calculate average).

The functions used to implement mutual exclusion involve the atomic library.

Analysis for n varying from 10 to 50, k=10, l1= 5 and l2 = 20 is in the following pages.

1) Average Waiting Time

Threads (n)	Average Waiting Time (ms)		
	TAS	CAS	Bounded Waiting CAS
10	55.23	62.14	112.25
20	227.4	239.46	373.88
30	456.99	435.3	680.74
40	620.57	605.62	1072.66
50	822.87	798.23	1493.76

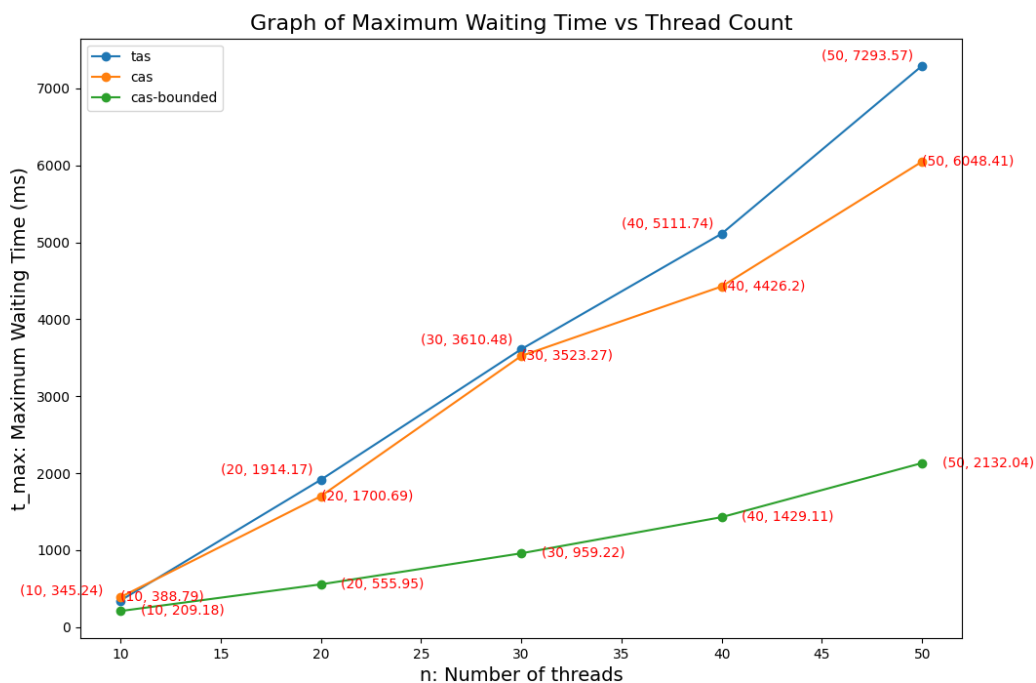


CAS and TAS have nearly equal average waiting times. (values of TAS are negligibly higher). This is because the code for TAS and CAS was nearly identical. The values 0 and 1 for the lock integer are equivalent to the boolean flag in TAS.

Bounded waiting with CAS had a greater average waiting time. This is because the thread exiting the critical section spends time checking for other requesting threads in a rotational manner. This is a big overhead as the number of threads increases.

2) Maximum Waiting Time

Threads (n)	Average Waiting Time (ms)		
	TAS	CAS	Bounded Waiting CAS
10	345.24	388.79	209.18
20	1914.17	1700.69	555.95
30	3610.48	3523.27	959.22
40	5111.74	4426.2	1429.11
50	7293.57	6048.41	2132.04



CAS and TAS have nearly equal average waiting times. TAS is slightly higher at the end.

As stated before, the code for TAS and CAS was nearly identical. The values 0 and 1 for the lock integer are equivalent to the boolean flag in TAS.

The maximum waiting time in bounded waiting with CAS is less than TAS and CAS. This is because the rotational checking in bounded CAS ensures that a thread will enter the critical section in definite time. There is no such guarantee in CAS and TAS. Other threads may repeatedly enter the critical section, keeping another thread waiting for very long.