Implementing Filter and Peterson based Tree Lock Algorithms
CS19B1027 - Vibhanshu Jain

Design:

- For the input, I am using ifstream and taking the inputs from “inp-params.txt’.
  For output also, I tried using outstream, but it was not able to print correctly, so I used fopen to open “output.txt” and used fprintf every time to print the logs.
- The main function creates n threads, firstly called the testCS function for testing Filter Lock followed by Peterson Lock.
- I am passing, thread id, lock details (which lock to use), output log file, k, lamda 1 and lamda 2 both the times followed by joining them again.

Filter Lock:

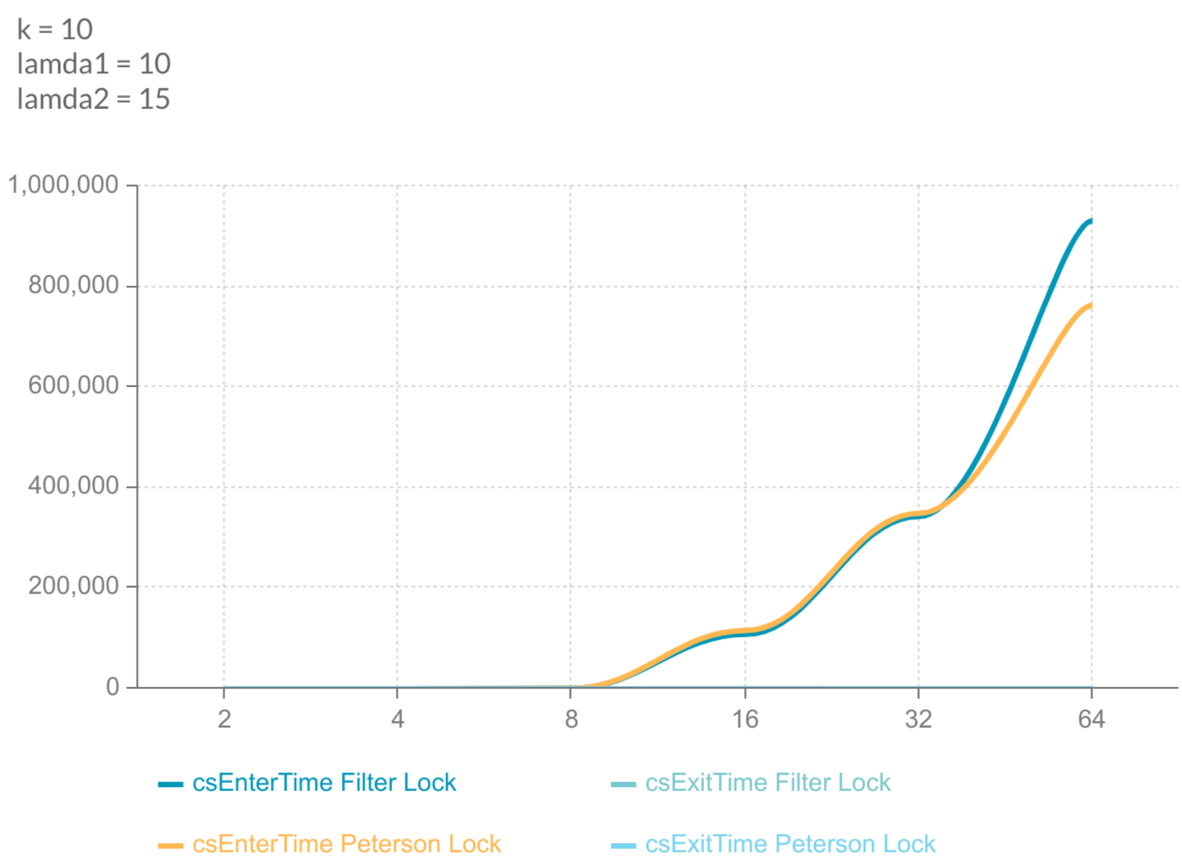
- The class has three elements and two functions. One is n, second is vector for level and third is vector for victim. Both these vectors are resized upon contruction.
- The complete function is implemented in the same way, as discussed in the book.

Peterson Tree Based Lock:

- In the tree, there is 2*n - 1 node, among which the first n-1 are two-threads Peterson locks and the remaining are the threads. The threads id are starting from 0 and thus we can say that the node representing the threads(with id a) is at n+a.
- Each thread is represented as the leaf in the tree at level l, then a two-threaded Peterson lock will be applied.
- n/2 lock for n,n+1 threads
  n/2+1 lock for n+2,n+3 threads
  All these locks will be acquired in step1, (n/2 threads aquired). Similarly in the next steps, n/4 and so on. In the last where it reaches root, the thread that acquires the last Peterson lock(at root only one lock will remain) can enter the CS.
- Acquiring a lock
  To aquire the lock, we must for the locks in the upper level to be released. This will continue until the root, where the last thread will go into CS.
  The lock for node i will be (i/2)-1
- Release a lock
  Release the lock acquired in the path from the leaf to the node.

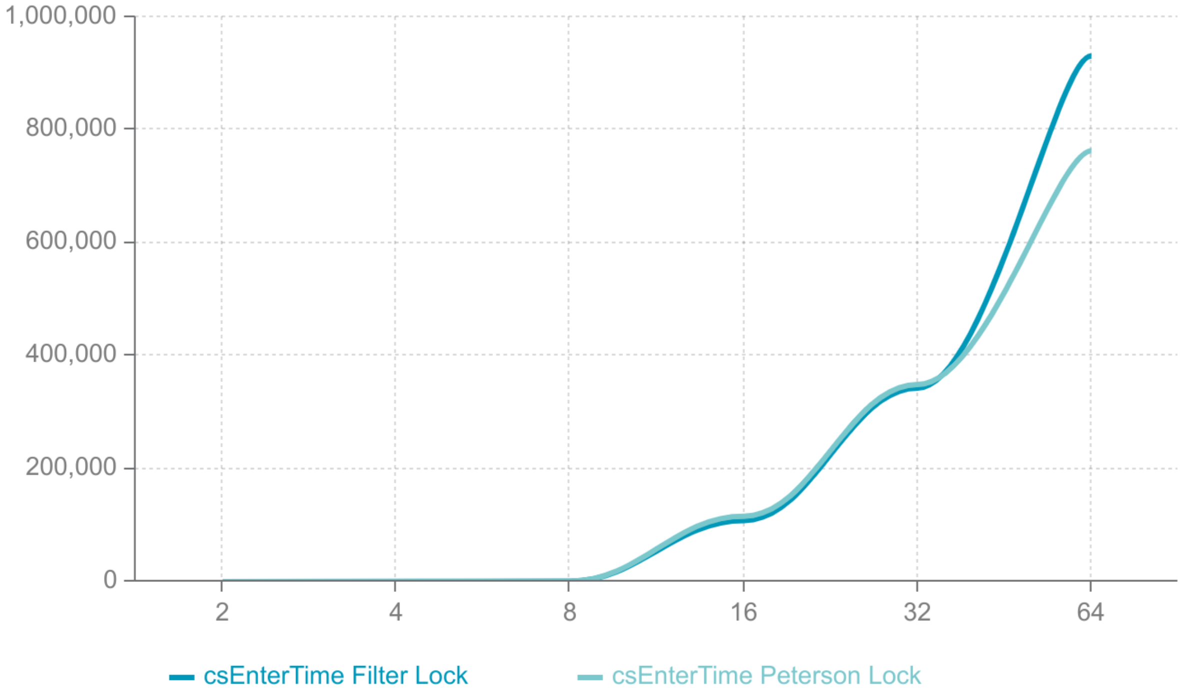
Number of threads vs Average time taken

Avg Time(in us)/No of threads	2	4	8	16	32	64
csEnterTime Filter Lock	37.85	220.65	872.612	108324	343079	931639
csExitTime Filter Lock	3.9	11.95	45.9375	205.219	5.45312	97.5359
csEnterTime Peterson Lock	54.5	174.25	786.375	115981	349212	763436
csExitTime Peterson Lock	6.4	6.625	3.65	25.9688	8.19375	14.0656

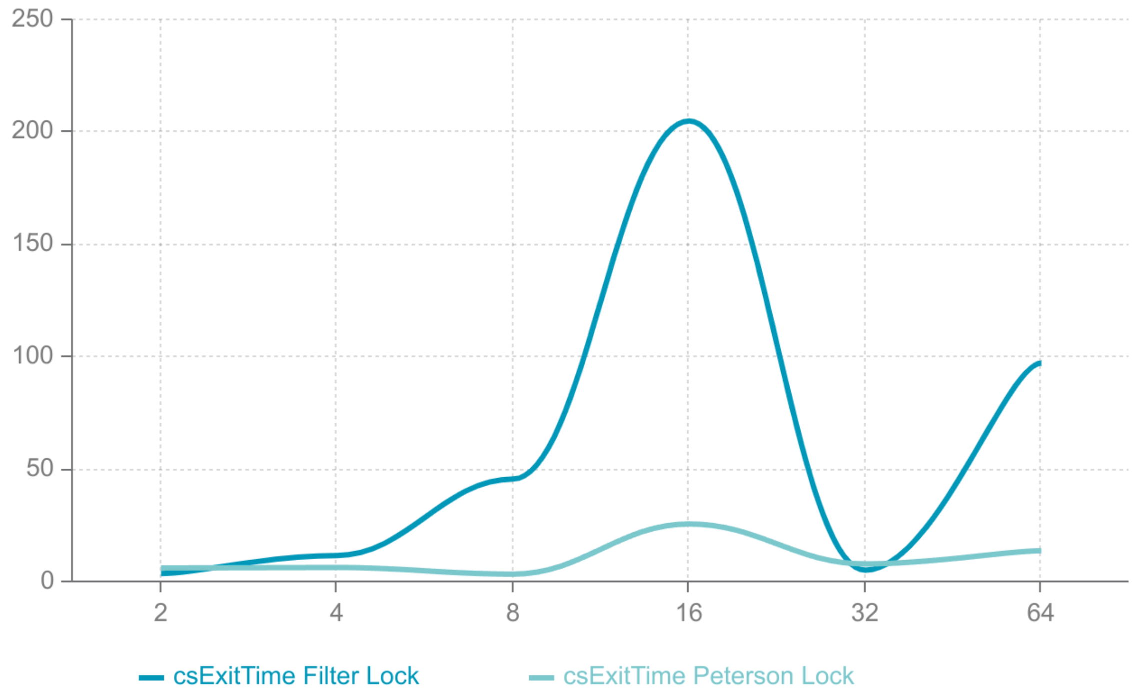


Some more charts:

k = 10
lamda1 = 10
lamda2 = 15



k = 10
lamda1 = 10
lamda2 = 15



Analysis:
The CS enter time for the Filter lock is slightly higher than the Peterson Lock, the reason behind this is the fact that the in filter lock each thread has to traverse n-1 levels, whereas in Peterson Lock on up to the height of the tree (log(n)).

It can also be observed that the csExitTime for Filter Lock is a little higher than the Peterson lock in general.

Conclusion:
The filter lock takes slightly more time than Peterson lock for entering the CS. Both takes approximately the same amount of time for exiting the CS.