

# Implementing Filter and Bakery Lock Algorithms

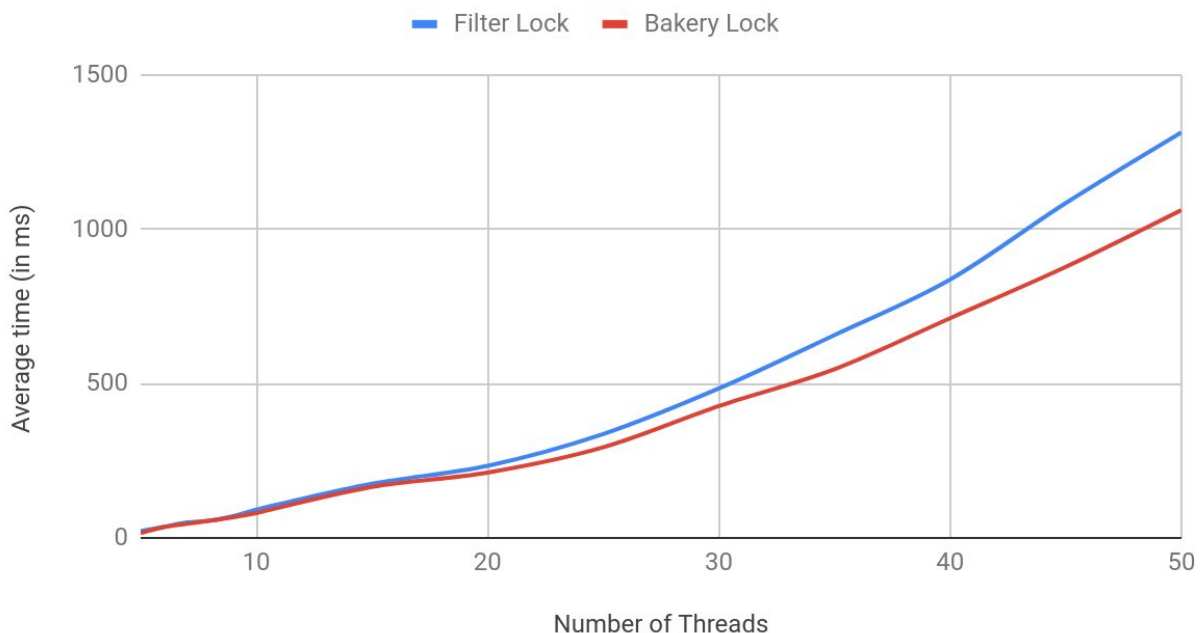
## Design

### Filter Lock and Bakery Lock

- A class was implemented with 2 methods, lock and unlock with specifications as given in the textbook(The Art of multiprocessor programming)
- A volatile Lock object Test, was declared which is accessible by all threads
- Within the class, objects which are to be shared by all threads were made volatile
  - Filter Lock -> level and victim array objects
  - Bakery Lock -> flag and label array objects
- In bakery lock, label array was made of long long type and an error message was displayed when the labels overflow and the program was terminated
- Average time was taken to enter the CS after the CS request was taken to evaluate the below graphs
- Only the Lock class was changed while keeping the rest of the boilerplate code same to maintain fairness in evaluating both the algorithms

## Graphs and Analysis

Average time vs Number of Threads



$\lambda_1 = 5$

$\lambda_2 = 20$

$N = 5$  to  $50$  (steps of  $5$ )

★ Each test was run 5 times and the average value was calculated to plot the graphs

- We can clearly see from the above graph that bakery lock perform better than filter lock
- In bakery lock algorithm, threads are served in FCFS manner, but in Filter Lock threads contend with each other
- Filterlock is starve free but a thread might be overtaken arbitrary number of times by other threads which results in more wait time for that thread
- In Bakery algorithm, when a thread try to acquire lock, it is given a label (1 higher than max of other threads) and it only enter in CS when there is no other thread with lower label.
- Thus the threads don't compete with each other and simply follows first come first serve and thus there is less wait time than in filter lock.

### **Filter Lock**

In filter lock algorithm, when a thread tries to acquire lock, It passes from level 1 to  $n$ . In each level it declares itself victim and waits till some other thread declares itself victim or threads in higher level finishes.

### **Bakery Lock**

In Bakery algorithm, when a thread tries to acquire lock , it is first given a label 1 higher than max of labels of all other threads. Then this thread busy waits till all the other threads with labels less than its label executes CS. When there is no thread left with lower label then it enters CS. We can clearly see that this algorithm follow First Come First Serve.