CMPT 431: Distributed Systems (Fall 2021)

Assignment 5 - Report

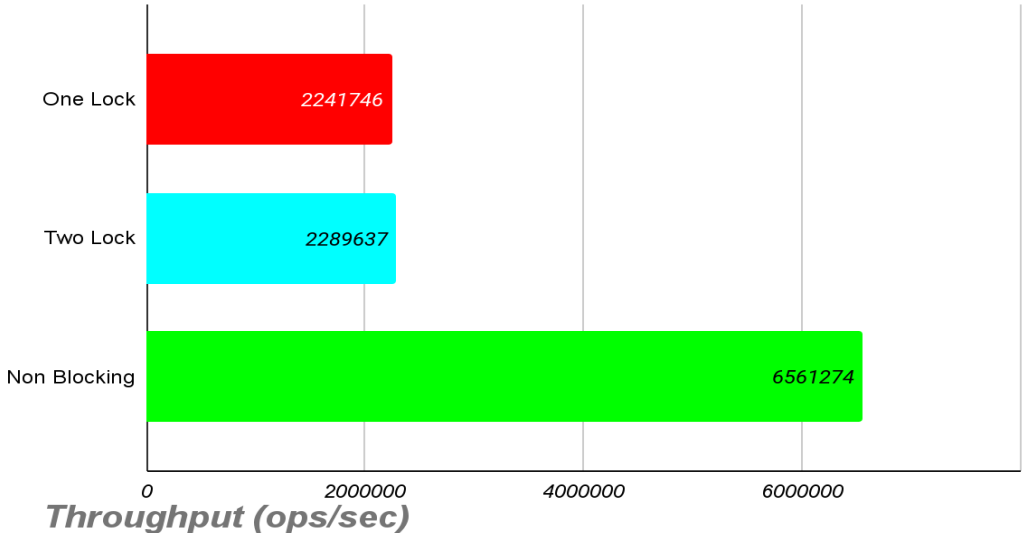
Name	Yu Ke
SFU ID	keyuk

Instructions:

- This report is worth 20 points.
- Answer in the space provided.
  Answers spanning beyond provided lines (11pt font) will lose points.
- All your answers must be based on the experiments conducted using the expected number of threads (as specified in the questions below) on fast nodes. Answers based on slow nodes and/or different numbers of workers than expected will result in 0 points.

---

1. [5 Points] Plot the total throughput for OneLock Queue, TwoLock Queue and Non-Blocking Queue with 4 producers and 4 consumers as a horizontal bar plot. The structure of your plot should be similar to the sample shown below: the x-axis has throughput in terms of number of operations ("**`Total throughput`**" from output) per second, and y-axis has the three queue implementations. The `a', `b', `c', `d', … on x-axis should be numbers (on linear scale).
   There may be some performance variation between runs, and hence you should take the average of 10 runs for each queue implementation. Remember to allocate one CPU per producer/consumer thread so that multiple threads do not have to fight for the same CPU (i.e., 4 producers + 4 consumers should be run using 8 CPUs).



2. [4 Points] Based on your plot from question 1, why does the throughput vary as it does across different queue implementations?

   One lock queue and two-lock queue have similar number of throughput, but the number of throughput for two-locks queue is slightly larger than the number of throughout for one-lock queue, because two-lock queue allows enqueue and dequeue operations at the same time. The number of throughput for non-blocking queue is much more larger than the other two queues, about 3 times. Because non-block queue doesn't use lock, then there is no race condition and privide more concurrency.

3. [7 Points] In Non-Blocking Queue pseudocode, there are two lines labeled as ELabel and DLabel. Why are those two lines present in the algorithm? Will the correctness and/or progress guarantees change if they are removed? If yes, which ones and why? If no, why? Support your argument using formal terms taught in class (e.g., linearizable, sequentially consistent, lock-free, wait-free, etc.).

   For the ELabel line, it is for if the tail is not pointing to the last node, swing tail to the next node; for the DLabel line, it tries to advance the tail if it is falling behind. The correctness and progress guarantees will not change if they are removed. Because the queue process is lock free, even in one queue operation, the tail is falling behind, other operation will fix this problem and then make this break loop.

4. [4 Points] In Non-Blocking Queue pseudocode, what is the purpose of the counter that is co-located with the next pointer? What can happen if the counter is removed? Please provide an example to support your argument.

   The purpose of the counter is to make the APA problem extremely unlikely. If the counter is removed, then there is a chance that ABA problem occurs, for example: There is a queue: Head->N1->N1, and two operations: Op1: dequeue and read N1; Op2: dequeue N1 and N2, then enqueue N1 into the queue. If Op1 goes to sleep because of some reason, and Op2 operates successfully and puts N1 and N2 into the freelist, then puts N1 back in the queue. Then Op1 awakes, and it regards N1 as the previous one, however, which is not. Then the ABA problem occurs.