ECE/CS 5510 Multiprocessor Programming
Homework 6

Mincheol Sung

November 18, 2018

Part I

Problem 1.
(1) A store-conditional by a thread fails the second linked-load of the other thread. (2) CPUs typically track the load-linked address at a cache-line or other granularity. Then any modification to any portion of the cache line causes the store-conditional to fail.

Problem 2.
(a) Linearizability: No. Can not pop(y) before push(y)
Sequentially consistency: Yes. A.push(x) A.push(y) B.pop(y) B.pop(x)

(b) Linearizability: No. pop(y) can not happen before push(y).
Sequentially consistency: No. pop(y) can not happen before push(y).

(c) Linearizability: Yes. A.push(x) B.pop(x) A.push(y) B.pop(y).
Sequentially consistency: Yes. A.push(x) B.pop(x) A.push(y) B.pop(y).

Problem 3.
Let there be a stack filled with elements of A and B. The top is currently A. When a thread T1 is about to do top.compareAndSet(A, B) in the pop(), it is scheduled out and another thread T2 is scheduled in. The T2 pops A, B and pushes D, C, and B sequentially. After the T2 finishes its job, the T1 comes in and finishes the top.compareAndSet(A,B) successfully. The status of stack is now B, C, D and the top is B which has NULL on its next as NULL. This may lead the GC to reclaim C and D to the memory because they are not referenced by no one.
The solution can be checking not only value, but also a time stamp with AtomicStampedReference. If the value is same but the time stame is different, the CAS should fail.

Problem 4.
(1)
enq() is wait-free because there is no loop or condition.

If the queue is empty, deq() is not lock-free since it will run forever.
If the queue is never empty, deq() is lock-free but not wait-free.
(2)
The linearization point of the enq() is line #7.
The linearization point of the deq() is line #14.

Problem 5.
Unless holding the lock, when a thread dequeues the last element and goes sleep before executing line #9, the other thread can see the queue is not empty.

Part II

I tested on the rlogin machine (spruce) with 40 CPUs and 100GB memory. I configured the duration as 10 sec and n as 5 for the SLQueue. Results are below.

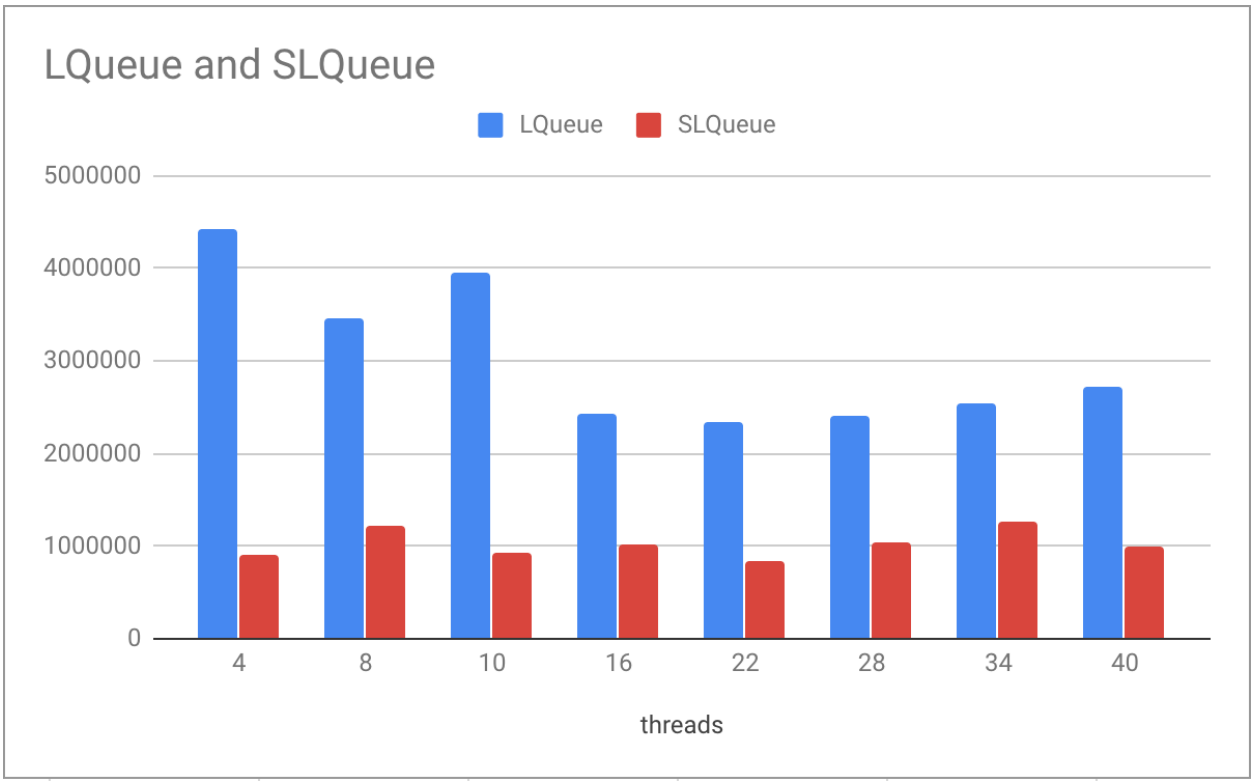

Figure 1: Throughput of LQueue and SLQueue