SPTCC June 25, 2017

Relaxed Concurrent Data Structures

Dan Alistarh

Problem 1. Recall the definitions of *bounded lock-freedom*, *wait-freedom*, and *stochastic scheduler* introduced in class. Prove that, for any fixed constant $B > 0$, any B-bounded lock-free is wait-free with probability 1 under any stochastic scheduler.

Problem 2. Recall the probabilistic analysis of the SprayList under the "perfect skiplist" assumption, given in class, and in particular the bound on the probability that a certain operation hits a specific element.

1. Fix a DeleteMin operation op, taking steps when $p - 1$ other operations might be in the system. Give a lower bound on the probability that the operation succeeds (does not collide) in walk to a bottom element.
2. Give bounds on the expected running time of a DeleteMin operation.

Problem 3. Consider an *approximate counter* implementation, which offers read and increment operations, with the usual semantics, except that the output of a read might be an approximation of the total number of increments which precede it in the linearization order. Our implementation works as follows:

- Each of the p threads has a local counter.
- To read, the thread picks one of the p counters at random, and returns its value multiplied by n as its output.
- To increment, the thread picks two of the p counters uniformly at random, compares their values, and increments by 1 the counter with the smaller value.

Assuming all operations are sequential, and using the analysis given in class, prove upper and lower bounds on the approximation properties of this counter implementation, i.e. bound how close to the true increment mean the returned values are. How can we modify the algorithm to improve these bounds?

Problem 4. Implement the above sequential process in e.g. python, and plot the difference between the true counter value and the returned counter values over 10000 steps alternating increment and read. Does the approximation factor depend on the number of steps we run? How does it depend on the number of distinct random trials a read / increment operation chooses from?

Problem 5. ($\star$) The priority queue analysis given in class assumed that the queues are never empty. Assume a scenario in which $m \gg n$ items of consecutive integer labels are inserted into the queues, and we then perform removals until no elements are left. Give bounds on the average cost of a removal, assuming that whenever two empty queues are chosen we pay a cost equal to the number of elements present in the system.