COP 4520, Spring 2015

Problem 3

Design and Implementation of Concurrent Queues:

Design a bounded lock-based queue implementation using an array instead of a linked list. Allow parallelism by using two separate locks for head and tail. Save this version of the queue and call it BLOCKING_QUEUE. Transform your algorithm into a lock-free algorithm and call it NONBLOCKING_QUEUE. Name the enqueue and dequeue operations *add* and *remove*. Describe briefly where do you run into difficulties in the construction of your lock-free solution.

Evaluate your approach by executing performance comparison of your BLOCKING_QUEUE, NONBLOCKING_QUEUE and the Intel TBB concurrent_queue. In your benchmark tests, vary the number of threads from 1 to 32 and produce graphs where you map the total execution time on the y-axis and the number of threads on the x-axis. Produce at least 3 different graphs representing different ratios of the invocation of enqueue and dequeue.

Make sure to use counted pointers to avoid the ABA problem. To avoid "polluting" your results with the overhead of memory management (the standard malloc and free don't scale well), you should have each thread pre-allocate its own supply of nodes, which it can keep on a private list when they are not in the queue. Finally, to avoid pollution from calls to the pseudo-random (the standard rand isn't even thread-safe), you should pre-generate enough random bits to drive the choice between enqueueing and dequeueing for the entire test run.

Provide a brief summary of your approach and an informal statement reasoning about the correctness and efficiency of your design.

Use either C or C++ for this assignment and provide a ReadMe file with instructions explaining how to compile and run your program.

Grading policy:

General program design and correctness: 65%

Efficiency: 15%

Documentation including statements and proof of correctness, efficiency, and experimental evaluation: 20%

Additional Instructions:

Cheating in any form will not be tolerated.

Please, submit your work via webcourses. Submissions by e-mail will not be accepted.

Due date: Monday, April 20th by 11:59 PM. Best of luck!!