CS 6V81: Special Topics in Computer Science—Concurrent Data Structures for Multi-Core Systems
Section 002
Programming Assignment 1

Instructor: Neeraj Mittal

Assigned on: Monday, February 20, 2012
Due date: Monday, March 5, 2012

You can work on this assignment in a group of at most two students.

1 Part 1

Write a program that spawns n threads, where n is a program argument. These threads access a shared counter (initialized as 0) in a loop. In each iteration, they read the counter to a local (stack) variable, increment it, and store it back to the counter.

When all threads complete 10000 iterations each, the program stops and prints the value of the shared counter. Note that the final value may be smaller than the total number of iterations.

Run the program on 1, 4, 8 and 16 threads and report the results in a table in a file. Also, report the run time of the program.

2 Part 2

Write a program doing the same task as in Part 1, but protect the shared counter using a lock, so no two threads modify the counter at the same time. (Use java.util.concurrent package for Java programs and Pthreads library for C/C++ programs.)

Run the program on 1, 4, 8 and 16 threads and report the results in a table in a file. Also, report the run time of the program.

3 Part 3

Experimentally compare the performance of lock-based and lock-free FIFO queue implementations described in Chapter 3 for two threads: one thread always adding an item to the queue and the other thread always removing an item from the queue.