Concurrent Project Proposal

Evan Schlomann

Andrew Kee

February 15, 2015

PROJECT TOPIC

When programming for multiprocessor/multithread computers, it is necessary to implement locking systems to prevent from overwriting shared resources. During our class we have already seen multiple locking mechanisms. We see that there are many ways to provide security for objects in our programming.

We plan to write and test multiple locking systems to discover where some locks perform well and where others do not. We will be testing how they perform over the number of cores available, the amount of memory the system has, the number of threads being used, etc. These statistics will allow us to determine which locks we should use when, and to help better understand locking systems.

CONCURRENCY ISSUES

The main issue with concurrent programming is the manipulation of shared variables. To protect these shared resources from being overwritten by separate threads, locks are used. Implementing locks requires the design of locking systems and the construction of these systems as well.

One of the more common locking systems is the Peterson Lock. This lock uses a simple idea, and is used for two threads. The lock has two variables, flag (used to signal the desire to go to the critical section), and the victim (when the lock is called, the calling thread declares itself as the victim, and allows the other thread to go first).

The Peterson Lock can be expanded to be used for n-threads. Other systems, such as the Bakery Lock and the Filter Lock are designed for n-thread systems. We will have to implement these locks and write tests to acquire statistics for analysis. With the analysis, we will then develop a performance model that will explain the data.

Proposed Solution

Timeline