Assessing the Suitability of Transactional Memory Synchronization in Concurrent GC

A concurrent garbage collection policy runs alongside, or concurrently, with the software application. However, problems can arise if synchronization between the application and the collector is not maintained.

Concurrent Garbage Collection

A concurrent garbage collection policy runs alongside, or concurrently, with the software application. However, problems can arise if synchronization between the application and the collector is not maintained.

Traditional Locking (a “pessimistic” approach)

Traditionally, this problem is solved using lock-based synchronization. A thread can only update an object if it is in possession of the object’s associated lock. This prevents concurrent updates.

Transactional Memory (an “optimistic” approach)

An alternative approach, which has just recently become supported in hardware, is called Transactional Memory (TM). Concurrent updates in a TM system can be thought of as automobiles approaching an intersection. Even though a given object may be affected by more than one update, it may still be possible multiple updates to proceed simultaneously.

Proposed Research

Transactional Memory is well suited for situations where the possibility of conflict exists, but the probability of it occurring is low.

Our research involves developing tools for profiling lock-based concurrent applications to identify key areas within the application where TM synchronization will be most beneficial. These tools will be used to profile the IBM J9 virtual machine to determine where its existing GC policies can be improved through TM synchronization.