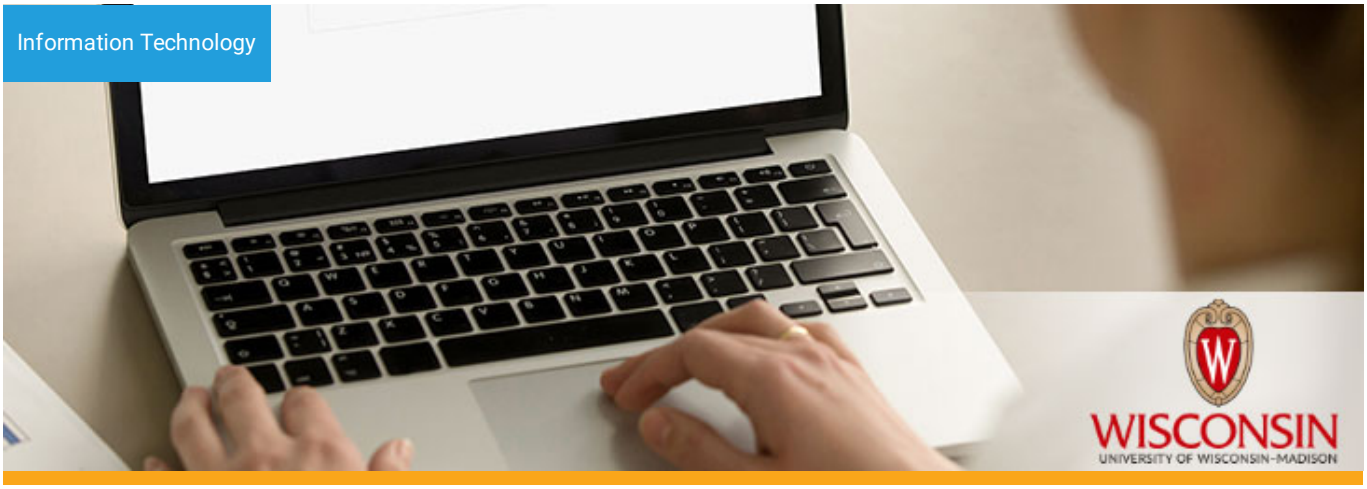
Information Technology



Computer Architecture Providing Transactional, Lock-Free Execution of Lock-Based Programs

View U.S. Patent No. 7,340,569 in PDF format.

WARF: P03229US

Inventors: Ravi Rajwar, James Goodman

The Wisconsin Alumni Research Foundation (WARF) is seeking commercial partners interested in developing a transactional lock removal (TLR) protocol that provides "lock-free" execution even when data conflicts are present.

Overview

Multi-threaded software provides multiple execution “threads” that act like independent (but cooperating) programs and can be assigned to different computer processors for parallel, and thus faster, execution. However, when different threads modify a common data source, lack of coordination can lead to errors. To avoid this, each “critical section,” consisting of a set of operations performed atomically (all at once), is typically protected by a lock, which a thread must acquire before executing the critical section.

While this technique prevents conflict, it also reduces the benefits of parallel execution. Based on the insight that different threads can often execute critical sections simultaneously without conflict, UW-Madison researchers previously developed a speculative lock elision (SLE) protocol in which each thread detects and then tentatively executes a critical section without first obtaining the lock. Execution is finalized only if no conflict is detected; otherwise execution is rolled back and reverts to acquiring the lock.

The Invention

The researchers have now extended this work to create a transactional lock removal (TLR) protocol that provides "lock-free" execution even when data conflicts are present. When data conflict is detected, TLR employs a conflict resolution scheme that determines which thread can retain ownership of the necessary data and execute the critical section first without causing errors. The scheme executes critical sections concurrently, but fairly, whenever possible, and efficiently when not.

Applications

- Multi-threaded computer software

Key Benefits

- By using the critical shared data itself to order any conflicting accesses to the data, TLR provides successful lock-free execution even in the presence of data conflicts.
- Provides a much more feasible and efficient programming environment for multi-threaded software because programmers can use locks freely to prevent conflict without fear of compromising software performance
- Provides hardware support for "lock-freedom" – the guarantee that a critical section will either be executed to completion or not at all, thus protecting against hardware (and some software) failures that might leave the database in an inconsistent state.
- By eliminating software overhead, it improves system-wide interactions and increases scalability.
- Designers can implement the conflict resolution scheme in hardware by building on existing cache coherence protocols.

Additional Information

Related Technologies

- See WARF reference number P02070US for information on the inventors' original speculative lock elision (SLE) protocol.

Tech Fields

- Information Technology : Computing methods, software & machine learning

For current licensing status, please contact Emily Bauer at emily@warf.org or 608-960-9842