Lock-Based Software Transactional Memory for Real-Time Systems

Catherine E. Nemitz and James H. Anderson

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL
Software Transactional Memory (STM)

Application Development

functionality + shared memory
Software Transactional Memory (STM)

- Application Development
  - functionality
  - + shared memory

- STM
  - shared memory

- Application Development
  - functionality

  separation of concerns
Software Transactional Memory (STM)

Application Development

functionality
+
shared memory

Real-time concern: bound priority-inversion blocking

Application Development

functionality

separation of concerns

STM

shared memory
A New Model: Lock-Based STM

Application

... begin <atomic> ...
... end <atomic> ...
...
A New Model: Lock-Based STM

Application

begin <atomic>
...
end <atomic>
...

Transaction
A New Model: Lock-Based STM

Traditional STM

1. transaction attempt
2. conflict detection
3. success
   3. rollback

Application

begin <atomic>
...
end <atomic>
...

Transaction

GOTO 1
A New Model: Lock-Based STM

Traditional STM

1. transaction attempt
2. conflict detection
3. success
3. rollback
   GOTO 1

Must bound number of retries.

Application

Transaction

... begin <atomic>
... end <atomic>
...
A New Model: Lock-Based STM

Traditional STM

1. transaction attempt
2. conflict detection
3. success
3. rollback

Lock-Based STM

0. offline conflict detection
1. lock resources
2. execute transaction
3. unlock resources

Application

... begin <atomic>
... end <atomic>
...

Must bound number of retries.
A New Model: Lock-Based STM

Traditional STM

1. transaction attempt
2. conflict detection
3. success
3. rollback

Must bound number of retries.

Application

... begin <atomic>
... end <atomic>
...

Lock-Based STM

0. offline conflict detection
1. lock resources
2. execute transaction
3. unlock resources

Transactions are guaranteed to commit with no retries.
A New Model: Lock-Based STM

Additional Research Directions

- Determine lock granularity
- Allow additional resource types
- Explore optimizations based on critical-section length or other parameters

Transactions are guaranteed to commit with no retries.

Lock-Based STM

0. offline conflict detection
1. lock resources
2. execute transaction
3. unlock resources