Software Transactional Memory

Marek Kubica

TU München

19.07.2011

Problem?

Concurrent writes in shared memory

- T1 reads value
- T1 computes new value
- T2 reads value
- T1 writes new value
- T2 computes new value
- T2 overwrites value of T1

How bad is it?

An example program in C using pthreads

```
$ ./broken 1 1000
Number = 1000
$ ./broken 2 1000
Number = 997
$ ./broken 15 1000
Number = 2408
$ ./broken 15 1000
Number = 3228
```

In short: computation results are nearly random.

Solutions

Typical real-world solutions

- Locking (Mutexes, Semaphores, Spinlocks)
- Message Passing (MPI)

Problems

- Deadlocks (Dining philosopher problem)
- Complex programs

Software Transactional Memory

Concept

- All writes run in transactions
- If a value in the transaction was modified, the transaction is rolled back
- Transactions get repeated if they don't succeed
- Optimistic model, efficient with infrequent writes

Fixed version

```
(def number (ref 0))
(def add1 (partial + 1))
(def num-threads (Integer/parseInt (first *command-line-args*)))
(def increments (Integer/parseInt (second *command-line-args*)))

(defn add-number [field times]
  (dorun (repeatedly times
                     (fn []
                       (dosync
                         (alter field add1))))))

(def thread (partial add-number number increments))

(printf "Starting %d threads, each with %d increments\n"
        num-threads increments)
(dorun (apply pcalls (repeat num-threads thread)))
(printf "Number = %d\n" @number)
(shutdown-agents)
```

Why does it work?

Optimistic model

- Runs computation, tries to write
- Succeeds or gets rolled back
- Efficient implementation: use data structures with O(1) rollback
- Increased concurrency

Pessimistic model

- Wait for lock, no computation
- Decreased concurrency

Drawbacks

Problems of STM

- Transactions sometimes hard to get right (inconsistent state)
- Language support not always great
- Increased bandwidth usage due to retried transactions

Sources & Questions

Source code

Minified C code on the left. Compile with `gcc -o broken broken.c -pthread`

Questions?

And thank you for listening.