



IBM T. J. Watson Research Center

STM Conflict-Free Complexity Trade-Offs

Maged Michael

IBM T J Watson Research Center

Workshop on What Theory for TM?

22 September 2009

Conflict-Free Transactions

Transaction A is **conflict-free** iff

There is no transaction* $B \neq A$ such that

A and B overlap in time

AND

$(R_A \cap W_B) \cup (W_A \cap R_B) \cup (W_A \cap W_B)$ is not empty

R_A The set of shared variables read by transaction A

W_A The set of shared variables written by transaction A

*Assume that all shared locations are accessed only inside transactions

STM Conflict-Free Time Complexity

Conflict-free time complexity of an STM implementation:

The sum of times for shared accesses executed by a conflict-free transaction

§ Reads: Constant time per shared read

§ Writes: $O(f(n))$ time per shared write, where n is the number of concurrent writes to the target location

What are the trade-offs and bounds on conflict-free time complexity?

Under what conditions can an STM implementation have conflict-free time complexity of $O(|R|+|W|)$?

Opacity – Read Set Consistency

Under *opacity* each transaction must have a consistent read set

An STM that supports opacity must prevent the following case:

Initially $x == 2$ and $y == 1$

invariant $x > y$

T1:

$r1 = x$ $\# x == 2$

$\# T1$ is doomed

$r2 = y$ $\# y == 2$

$z = 1/(1-r1/r2)$ $\#$ division by zero

T2:

$x = 4$

$y = 2$

commit

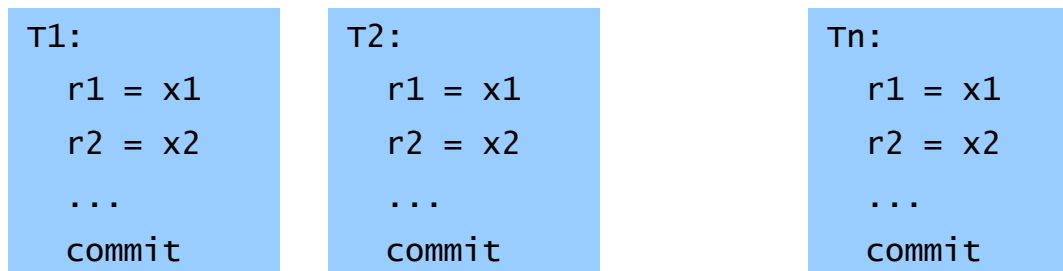
Can an STM implementation that satisfies opacity have conflict-free time complexity of $O(|R|+|W|)$?

Visible Reads

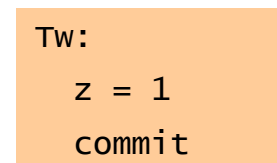
Readers write to some STM metadata

Writers must detect conflicts with already active readers

N concurrent conflict-free readers



Conflict-free writer



§ For each read location, the n readers write to k distinct locations

§ Cost of reads: $O(|R| * f(n/k))$

§ A writer will have to check k locations per write

§ Cost of writes: $O(|W| * k)$

§ Conflict-free time complexity: $O(|R| * f(n/k) + |W| * k)$

Invisible Reads – Full Validation

Validate prior reads on every read, e.g., DSTM

§ Cost of reads: $O(|R|^2)$

§ Conflict-free time complexity: $O(|R|^2 + |W|)$

Invisible Reads – Global Version Number

Writers update a global version number, e.g., TL2, TinySTM

N concurrent conflict-free writers

T1:

```
y[1] = 1
commit
```

T2:

```
y[2] = 1
commit
```

Tn:

```
y[n] = 1
commit
```

§ Each transaction issues a writing atomic operation to increment the global version number

CAS

CAS

CAS

§ Cost of commit: $O(f(n))$

§ Conflict-free time complexity: $O(|R|+|W|+f(n))$

Invisible Reads – Global Version Number

Using Special Atomic Ops

Try to increment. Use value set by winner (TL2 GV4)

CAS

CAS

CAS

§ Losing CAS operations take non-constant time to resolve

§ **Atomic <Op>OrLose primitives (e.g., using LL/SC)**

§ Try atomic op. If it is certain that some other concurrent operation of the same type will complete successfully, return failure.

IncOrLose

IncOrLose

IncOrLose

§ OpOrLose primitives can conceivably take constant time

§ Conflict-free time complexity: $O(|R|+|W|)$

Invisible Reads – Thread Local Clocks

Each thread has its own clock

Each thread maintain a private lazily updated list of other thread's clocks

§ An out-of-date private clock value can lead to a conflict-free abort

§ Each read with out-of-date clock of last updater can lead to an abort

§ Aborted executions that occur while a transaction is conflict-free are counted in conflict-free complexity

§ Conflict-free time complexity: $O(|R|^2 + |W|)$

```
T:  
  r1 = x1  
  abort  
  r1 = x1  
  r2 = x2  
  abort  
  r1 = x1  
  ...  
  commit
```

Invisible Reads – HW Watches

Readers set HW watches

HW invalidate watches if location is written

Readers check all watches in one step

§ Bounded HW watches

§ Conflict-free time complexity: $O(|R|^2 + |W|)$

Invisible Reads – HW Clocks

Imprecise clocks

- § Possible unnecessary abort/revalidation per reads
- § Conflict-free time complexity: $O(|R|^2 + |W|)$

Perfectly synchronized global clock

- § Is it reasonable to assign constant cost to reading a perfectly synchronized global clock?
- § Maybe, but with prohibitive HW cost.
- § Conflict-free time complexity: $O(|R| + |W|)$

Summary

- § **Under what conditions can an STM implementation that satisfies opacity have conflict-free time complexity of $O(|R|+|W|)$?**
- § **Can this be achieved using only common primitives?**
 - No constant time perfectly synchronized clock
 - No OpOrLose atomic primitives

THANK YOU

Questions?