Brief Announcement: Enhancing Permissiveness in Transactional Memory via Time-Warping

Nuno Diegues and Paolo Romano

INESC-ID Lisboa / Instituto Superior Técnico
Universidade de Lisboa, Portugal
ndiegues@gsd.inesc-id.pt, romano@inesc-id.pt

Abstract. The notion of permissiveness in Transactional Memory (TM) translates to only aborting a transaction when it cannot be accepted in any history that guarantees a target correctness criterion. Achieving permissiveness, however, comes at a non-negligible cost. This desirable property is often neglected by state of the art TMs, which, in order to maximize implementation's efficiency, resort to aborting transactions under overly conservative conditions. We identify a novel sweet spot between permissiveness and efficiency by introducing the Time-Warp Multi-version algorithm (TWM), which allows for drastically minimizing spurious aborts with respect to state of the art, highly efficient TMs, while introducing minimal bookkeeping overheads. Further, read-only transactions are abort-free, and both Virtual World Consistency and lock-freedom are ensured.

1 Overview of Time-Warping

Typical MVCC algorithms for TM allow read-only transactions to be serialized "in the past", i.e., before the commit event of any concurrent write transaction. Conversely, they serialize a write transaction T committing at time t "in the present", by: (1) attempting to order the versions produced by T after all versions created by transactions committed before time t; and (2) performing what we call a "classic validation", which ensures that the snapshot observed by T is still up-to-date considering the updates generated by all transactions that committed before t. This results in aborting any write transaction T that missed the writes of a concurrent, committed transaction T', also called an anti-dependency in the literature. We note that this approach is a conservative one, as it guarantees serializability by systematically rejecting serializable histories in which T might have actually been safely serialized before T'.

The key idea of TWM is to allow a write transaction, which missed the write committed by a concurrent transaction T', to be serialized "in the past", namely before T'. This is in contrast with the approach taken by most practical TM algorithms (designed to minimize overhead), which only allow the commit of transactions "in the present". Unlike TMs that ensure permissiveness [2], TWM tracks solely direct anti-dependencies developed by a committing transaction, hence avoiding onerous validation of the entire conflicts' graph [3]. TWM's novel validation is sufficiently lightweight to ensure efficiency, while accepting far more histories than state of the art, practical TMs.

To efficiently implement the time-warp abstraction, TWM maintains two totally ordered time lines: $\mathcal{N}$ for the natural order of commit requests and $\mathcal{TW}$ for the time-warp

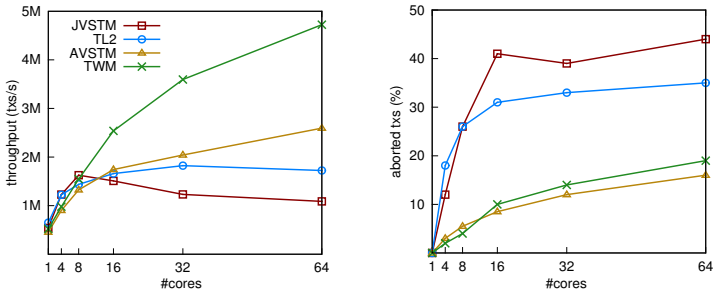

Fig. 1: Comparison of throughput (left) and aborts (right) in a skip-list.

commit order that results in the version order of data. In a conflict-free execution both orders coincide. Otherwise, a transaction B is time-warped when it anti-depends on a set of concurrent transactions A. In such case, B is serialized (along $\mathcal{TW}$) before the transaction in A with the least natural commit order. Note that if B has no anti-dependencies, then the natural and time-warp commit order coincide (as B is serialized in the present). To ensure only safe executions (virtual world consistent ones), the validation scheme of TWM detects a specific pattern that we call a *triad*. A triad exists whenever there is transaction T that is both the source and target of anti-dependency edges from two concurrent transactions T' and T'' (where, possibly, $T' = T''$). We call T a *pivot*, and define the TWM validation scheme as follows: A transaction fails its validation if, by committing, it would create a triad whose *pivot* time-warp commits.

Results. We conducted an experimental study against four representative TMs. TL2 and JVSTM use the *classic validation* (the latter is also multi-version), and AVSTM is probabilistic permissive. We provide a sample of the evaluation for a contended skip-list in Fig. 1. Our overall results in typical TM benchmarks yielded an average improvement of 81% in high concurrency scenarios, with gains extending up to $8\times$. Further, we observed limited overheads, even in worst-case scenarios entailing no contention or patterns that cannot be optimized using TWM. More details can be found in [1].

Acknowledgements. This work was supported by national funds through Fundação para a Ciência e Tecnologia under project PEst-OE/EEI/LA0021/2013 and Cloud-TM project (co-financed by the European Commission through the contract no. 257784).

References

1. Nuno Diegues and Paolo Romano. Enhancing Permissiveness in Transactional Memory via Time-Warping. Technical Report RT/10/2013, INESC-ID Lisboa, July 2013.
2. Rachid Guerraoui, Thomas A. Henzinger, and Vasu Singh. Permissiveness in transactional memories. In *Proceedings of the 22nd International Symposium on Distributed Computing*, DISC '08, pages 305–319, 2008.
3. Idit Keidar and Dmitri Perelman. On avoiding spare aborts in transactional memory. In *Proceedings of the 21th annual Symposium on Parallelism in Algorithms and Architectures*, SPAA '09, pages 59–68, 2009.