

A Sample Article for the LIPIcs series*

John Q. Open¹ and Joan R. Access²

- 1 Dummy University Computing Laboratory, Address/City, Country
open@dummyuniversity.org
- 2 Department of Informatics, Dummy College, Address/City, Country
access@dummycollege.org

Abstract

State-of-the-art *software transactional memory (STM)* implementations achieve good performance by carefully avoiding the overhead of *incremental validation* (i.e., re-reading previously read data items to avoid inconsistency) while still providing *progressiveness* (allowing transactional aborts only due to *data conflicts*). Hardware transactional memory (HTM) implementations promise even better performance, but offer no progress guarantees. Thus, they must be combined with STMs, leading to *hybrid TMs (HyTMs)* in which hardware transactions must be *instrumented* (i.e., access metadata) to detect contention with software transactions.

We show that, unlike in progressive STMs, software transactions in progressive HyTMs cannot avoid incremental validation. In fact, this result holds even if hardware transactions can *read metadata non-speculatively*. We then present *opaque* HyTM algorithms providing *progressiveness for a subset of transactions* that are optimal in terms of hardware instrumentation. We explore the concurrency vs. hardware instrumentation vs. software validation tradeoffs for these algorithms. Our experiments with Intel and IBM POWER8 HTMs seem to suggest that (i) the *cost of concurrency* also exists in practice, (ii) it is important to implement HyTMs that provide progressiveness for a maximal set of transactions without incurring high hardware instrumentation overhead or using global contending bottlenecks and (iii) there is no easy way to derive more efficient HyTMs by taking advantage of non-speculative accesses within hardware.

1998 ACM Subject Classification Dummy classification – please refer to <http://www.acm.org/about/class/ccs98-html>

Keywords and phrases Dummy keyword – please provide 1–5 keywords

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

References

- 1 Edsger W. Dijkstra. Letters to the editor: go to statement considered harmful. *Commun. ACM*, 11(3):147–148, 1968. doi:10.1145/362929.362947.
- 2 Jim Gray and Andreas Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.
- 3 John E. Hopcroft, Wolfgang J. Paul, and Leslie G. Valiant. On time versus space and related problems. In *16th Annual Symposium on Foundations of Computer Science, Berkeley, California, USA, October 13-15, 1975*, pages 57–64. IEEE Computer Society, 1975. doi:10.1109/SFCS.1975.23.
- 4 Donald E. Knuth. Computer Programming as an Art. *Commun. ACM*, 17(12):667–673, 1974. doi:10.1145/361604.361612.

* This work was partially supported by someone.