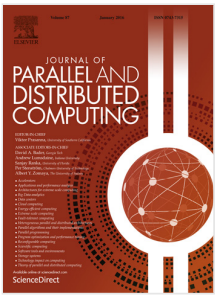
Accepted Manuscript



Grasping the gap between blocking and non-blocking transactional memories

Petr Kuznetsov, Srivatsan Ravi

PII: S0743-7315(16)30126-5
DOI: http://dx.doi.org/10.1016/j.jpdc.2016.10.008
Reference: YJPDC 3546

To appear in: *J. Parallel Distrib. Comput.*

Received date: 1 February 2016
Accepted date: 4 October 2016

Please cite this article as: P. Kuznetsov, S. Ravi, Grasping the gap between blocking and non-blocking transactional memories, *J. Parallel Distrib. Comput.* (2016), http://dx.doi.org/10.1016/j.jpdc.2016.10.008

Grasping the Gap between Blocking and Non-blocking Transactional Memories

Petr Kuznetsov[a], Srivatsan Ravi[b]

[a]*Telecom ParisTech, FRANCE, Email: petr.kuznetsov@telecom-paristech.fr*
[b]*Dept. of Computer Science, Purdue University, Email: srivatsan@srivatsan.in*

Abstract

Transactional memory (TM) is an inherently optimistic abstraction: it allows concurrent processes to execute sequences of shared-data accesses (transactions) speculatively, with an option of aborting them in the future. Early TM designs avoided using locks and relied on non-blocking synchronization to ensure *obstruction-freedom*: a transaction that encounters no step contention is not allowed to abort. However, it was later observed that obstruction-free TMs perform poorly and, as a result, state-of-the-art TM implementations are nowadays *blocking*, allowing aborts because of data *conflicts* rather than step contention.

In this paper, we explain this shift in the TM practice theoretically, via complexity bounds. We prove a few important lower bounds on obstruction-free TMs. Then we present a *lock-based* TM implementation that beats all of these lower bounds. In sum, our results exhibit a considerable complexity gap between non-blocking and blocking TM implementations.

Keywords: Transactional memory, Obstruction-freedom, Memory stalls, Expensive synchronization, Lower bounds, Invisible reads, Disjoint-access parallelism, Perturbability, Blocking, Non-blocking