Brief Announcement: Breaching the Wall of Impossibility Results on Disjoint-Access Parallel TM

Sebastiano Peluso[1 2 3], Roberto Palmieri[1], Paolo Romano[2], Binoy Ravindran[1], and Francesco Quaglia[3]

[1] Virginia Tech, Blacksburg, VA, USA. {peluso,robertop,binoy}@vt.edu
[2] IST/INESC-ID, Lisbon, Portugal. {peluso,romanop}@gsd.inesc-id.pt
[3] Sapienza University, Rome, Italy. {peluso,quaglia}@dis.uniroma1.it

Abstract. Transactional Memory (TM) implementations guaranteeing disjoint-access parallelism (DAP) are desirable on multi-core architectures because they can exploit low-level parallelism. In this paper we look for a breach in the wall of existing impossibility results on DAP TMs, by identifying the strongest consistency and liveness guarantees that a DAP TM can ensure while maximizing efficiency in read-dominated workloads. Along the path of designing this protocol, we report two impossibility results related to ensuring real-time order in a DAP TM.

Keywords: Transactional Memory, Disjoint-Access Parallelism, Real-Time Order.

1 Overview of the Achieved Results

A property that is deemed as crucial for the scalability of a TM is its ability to avoid any contention on shared objects, also called *base objects*, among transactions that access disjoint data sets – *disjoint-access parallelism* (or DAP) [1]. Also, since many real-world workloads are often read-dominated, another aspect with strong impact on performance of TM algorithms is optimizing the processing of read-only transactions. In this sense, two main properties are regarded as particularly important for read-only transactions: wait-freedom, i.e. transactions are never blocked or aborted (WFRO), and invisible reads, i.e. read operations never update any datum or base object (IRO). We succinctly denote their union as WFIRO.

Given the set of impossibility results related to implementing TM algorithms that guarantee different variants of the DAP property, as well as alternative consistency and liveness criteria [1–3], in this paper we find a breach in this wall of impossibility results, seeking an answer to the following question: what are the strongest *consistency* and *liveness* guarantees that a TM can ensure while remaining scalable — by ensuring DAP — and maximizing efficiency in read-dominated workloads — by having WFIRO? Our search space considers the Cartesian product of the consistency criteria specified by Adya's hierarchy [4]

and of a set of liveness properties that comprises both TM-specific criteria [5], as well as classical progress criteria, i.e. obstruction-, lock- and wait-freedom.

Along the path that leads us to answer the above question, we also prove two novel impossibility results. If one selects *any* consistency criterion that ensures Real Time Order (RTO), i.e. by ensuring that transactions appear as executed without reversing the partial order defined by non-concurrent transactions, and independently of the isolation guarantees for concurrent transactions, it is impossible to ensure also WFRO, obstruction-free update transactions and the weakest form of DAP [1]. Further, even assuming weakly progressive update transactions [5], we are still faced with an impossibility result if we want IRO.

These results highlight the necessity of relaxing RTO to implement a scalable TM that maximizes the efficiency of read-only transactions by jointly guaranteeing DAP and WFIRO. This leads us to introduce a weaker variant of RTO, named *Witnessable Real Time Order* (WRTO), which demands that the RTO is enforced only among transactions exhibiting (transitive) data conflicts.

By adopting WRTO, we design a WFIRO TM that guarantees the strongest variant of DAP [2], strong progressiveness [5] and a consistency criterion whose semantics is very close to those provided by popular safety properties for TM, such as Opacity. This consistency criterion, known as Extended Update Serializability (EUS) [4, 6] guarantees the serializability of the history of committed update transactions. Further, EUS ensures that all transactions (also transactions that eventually abort) observe a snapshot producible by some equivalent serialization of the history of (committed) update transactions.

Acknowledgments. This work is supported in part by US National Science Foundation under grant CNS-1217385.

References

1. Attiya, H., Hillel, E., Milani, A.: Inherent Limitations on Disjoint-Access Parallel Implementations of Transactional Memory. J. Theory Comput. Syst. 49(4), 698–719 (2011)
2. Guerraoui, R., Kapalka, M.: On Obstruction-free Transactions. In: 20th Annual Symposium on Parallelism in Algorithms and Architectures, pp. 304–313. ACM, New York (2008)
3. Bushkov, V., Dziuma, D., Fatourou, P., Guerraoui, R.: The PCL Theorem. Transactions cannot be Parallel, Consistent and Live. In: 26th annual Symposium on Parallelism in Algorithms and Architectures, pp. 178–187. ACM, New York (2014)
4. Adya, A.: Weak Consistency: A Generalized Theory and Optimistic Implementations for Distributed Transactions. PhD Thesis, MIT (1999)
5. Guerraoui, R., Kapalka, M.: The Semantics of Progress in Lock-based Transactional Memory. In: 36th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 404–415. ACM, New York (2009)
6. Peluso, S., Ruivo, P., Romano, P., Quaglia, F., Rodrigues, L.: When Scalability Meets Consistency: Genuine Multiversion Update-Serializable Partial Data Replication. In: 32nd IEEE International Conference on Distributed Computing Systems, pp. 455–465. IEEE Computer Society, Washington DC (2012)