On Breaching the Wall of Impossibility Results on Disjoint-Access Parallel STM

Sebastiano Peluso
Virginia Tech
peluso@vt.edu

Roberto Palmieri
Virginia Tech
robertop@vt.edu

Paolo Romano
INESC-ID / IST
romano@inesc-id.pt

Binoy Ravindran
Virginia Tech
binoy@vt.edu

Francesco Quaglia
Sapienza University of Rome
quaglia@dis.uniroma1.it

Over the last decade, Software Transactional Memory (STM) has emerged as an attractive paradigm for simplifying parallel programming. A property that is deemed as central for the scalability of a STM implementation is its ability to avoid any interference between transactions that access disjoint data sets – a property called *disjoint-access parallelism* (DAP).

Initial works [2, 8, 3, 4] in the area of DAP and STM addressed a number of fundamental questions, including impossibility results, on the cost of building STM systems achieving DAP, while ensuring popular consistency criteria such as Opacity and Snapshot Isolation. This body of literature concludes that an STM cannot be DAP, ensure wait-free and invisible read-only transactions (WFIRO), and guarantee Strict Serializability [2]. Since these works established a number of impossibility results related to the implementation of DAP STMs [2, 5, 8], in this talk, we intend to present a possibility result: an algorithm that achieves a set of desirable properties, i.e. DAP and WFIRO transactions, by embracing consistency criteria different from those typically targeted by existing STM algorithms (such as SI or Opacity).

More in detail, we seek an answer to the following question: considering Adya's consistency hierarchy [1], what is the strongest criterion that allows for achieving DAP and WFIRO? This criterion is EPL-3U, also known as Extended Update Serializability (EUS) [7]. EUS ensures that: *i)* every read operation always observes a consistent transactional state, and *ii)* write-committed transactions are serializable. In addition, due to the impossibility result in [5], our algorithm guarantees weakly progressive write transactions [6].

By doing so, we also take into account the real-time order property and we reveal an impossibility result: the real-time order cannot be ensured in an STM that is weakly DAP, obstruction-free, and guarantees wait-free read-only transactions, irrespective of the visibility of read operations and the consistency level. Leveraging this impossibility result, we show that the lower bound on the visibility of read-only transactions in [2] is not a sufficient condition to guarantee Strict Serializability and hence Opacity. We prove also that, ensuring the real-time order is still impossible in a weakly DAP and weakly progressive STM that ensures WFIRO. Therefore, in our algorithm, we enforce real-time order to only conflicting transactions.

Finally, we show that, in the proposed algorithm, the spatial cost of each object version is optimal, i.e. it is a necessary cost for any algorithm that ensures EUS, DAP and WFIRO.

References

[1] Atul Adya. *Weak Consistency: A Generalized Theory and Optimistic Implementations for Distributed Transactions*. PhD thesis, 1999. AAI0800775.

[2] Hagit Attiya, Eshcar Hillel, and Alessia Milani. Inherent limitations on disjoint-access parallel implementations of transactional memory. *Theory Comput. Syst.*, 49(4):698–719, 2011.

[3] Victor Bushkov, Rachid Guerraoui, and Michal Kapalka. On the liveness of transactional memory. In *PODC*, pages 9–18, 2012.

[4] Faith Ellen, Panagiota Fatourou, Eleftherios Kosmas, Alessia Milani, and Corentin Travers. Universal constructions that ensure disjoint-access parallelism and wait-freedom. In *PODC*, pages 115–124, 2012.

[5] Rachid Guerraoui and Michal Kapalka. On obstruction-free transactions. In *SPAA*, pages 304–313, 2008.

[6] Rachid Guerraoui and Michal Kapalka. The semantics of progress in lock-based transactional memory. In *POPL*, pages 404–415, 2009.

[7] Sebastiano Peluso, Pedro Ruivo, Paolo Romano, Francesco Quaglia, and Luis Rodrigues. When scalability meets consistency: Genuine multiversion update-serializable partial data replication. ICDCS '12, pages 455–465, 2012.

[8] Dmitri Perelman, Rui Fan, and Idit Keidar. On maintaining multiple versions in STM. PODC '10, pages 16–25, 2010.