Exploiting Speculation in Partially Replicated Transactional Data Stores

Zhongmiao Li[†⋆], Peter Van Roy[†] and Paolo Romano[⋆]
[†]Université catholique de Louvain [⋆]Instituto Superior Técnico, Lisboa & INESC-ID

CCS CONCEPTS

• **Information systems** → **Distributed database transactions**; **Storage replication**;

Modern online services are increasingly deployed over geographically-scattered data centers (geo-replication), which allows services to be highly available and reduces access latency. On the down side, to provide ACID transactions, global (i.e., inter-data center) certification is needed to detect conflicts between concurrent transactions executing at different data centers. The global certification phase leads not only to impairing throughput, as transactions need to hold pre-commit locks during their global certification phase; it also directly impacts the client-perceived latency, as global certification lies in the critical path of transaction execution.

Internal and external speculation. This work investigates the use of two speculative techniques to alleviate the above problems: *speculative reads* and *speculative commits*.

Speculative reads allow transactions to observe the data item versions produced by pre-committed transactions, instead of blocking until they are committed/aborted. Speculative reads can reduce the "effective duration" of pre-commit locks, thus enhancing throughput by maximizing the degree of achievable parallelism. Speculative reads are an *internal speculation* technique, as misspeculations caused by it never surface to the clients.

Speculative commits remove the global certification phase from the critical path of transaction execution. Therefore, speculative commits can drastically reduce user-perceived latency, but also expose to external clients the results produced by transactions still undergoing global certification. Thus, speculative commits require programmers to define compensation logic to deal explicitly with misspeculations.

Avoiding the pitfalls of speculation. Past work has shown that the use of speculative reads and speculative commits [2–4] can enhance the performance of transactional systems. However, these approaches suffer from several limitations:

1. Unfit for geo-distribution/partial replication. Some existing works in this area were not designed for partially replicated geo-distributed data stores, as they either target full replication [4] or rely on centralized sequencer that imposes prohibitive costs in WAN environments [3].

2. Subtle concurrency anomalies. Existing geo-distributed transactional data stores that support speculative reads [2] expose applications to anomalies, e.g., data snapshots that reflect only partial updates of transactions or include versions created by conflicting concurrent transactions. Such anomalies can be potentially quite dangerous as they can lead applications to exhibit unexpected behaviors (e.g., crashing or hanging in infinite loops) and/or can externalize erroneous states to clients.

3. Performance robustness. In adverse scenarios, the injudicious use of speculative techniques can actually significantly penalize performance, rather than benefitting it.

Contributions. We propose Speculative Transaction Replication (STR), a novel speculative transactional protocol for partially replicated geo-distributed data stores. The key contribution of STR is its highly scalable and efficient distributed concurrency control scheme that supports speculative execution, while providing intuitive and stringent consistency guarantees via the Speculative Snapshot Isolation (SPSI) criterion. Besides guaranteeing the familiar Snapshot Isolation (SI) [1] for *committed transactions*, SPSI allows an *executing transaction* to read data item versions committed before it started (as in SI), and to atomically observe the effects of non-conflicting transactions that originated on the same node and pre-committed before it started. This shelters programmers from having to reason about complex concurrency anomalies that can otherwise arise in speculative systems.

To enhance performance robustness, STR employs a light-weight self-tuning mechanism that dynamically adjusts the aggressiveness of the speculative mechanisms employed by the system based on the workload characteristics.

Our evaluation shows that the use of internal speculation (speculative reads) yields 6× throughput improvements and 10× latency reduction in a fully transparent way. Further, applications that exploit external speculation (speculative commits) can achieve a reduction of user-perceived latency by up to 100× and to boost throughput by up to 4.7×.

ACKNOWLEDGEMENT

This work has been partially funded by the H2020 project 732505 LightKone, by FCT via projects UID/CEC/50021/2013 and PTDC/EEISCR/1743/2014 and by EACEA award 2012-0030.

REFERENCES

[1] Sameh Elnikety et al. 2005. Database replication using generalized snapshot isolation. In *SRDS*. IEEE.
[2] Goetz Graefe et al. 2013. Controlled lock violation. In *SIGMOD*.
[3] Evan Jones et al. 2010. Low overhead concurrency control for partitioned main memory databases. In *SIGMOD*. ACM.
[4] Paolo Romano et al. 2014. On Speculative Replication of Transactional Systems. *J. Comput. Syst. Sci.* 80, 1 (Feb. 2014), 20.

SoCC '17, September 24–27, 2017, Santa Clara, CA, USA

ACM ISBN 978-1-4503-5028-0/17/09.
https://doi.org/10.1145/3127479.3132692