Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Master's Thesis Proposal

Mechanized Consistency Models for Distributed Database Transactions

Shabnam Ghasemirad

Supervisors: Christoph Sprenger and Si Liu
Professor: Prof. David Basin
Issue date: April 2022

Introduction

Cloud database systems use transactions for synchronization. Depending on the desired scalability and availability, different systems provide different transactional consistency guarantees, aka. consistency models in the literature. Prevalent consistency models include read atomic (RA) [BFG+16], causal consistency (CC) [LFKA11], parallel snapshot isolation (PSI) [SPAL11], snapshot isolation (SI) [BBG+95], and serializability (SER) [Pap79]. Different concurrency control mechanisms have been proposed that implement these consistency notions. For example, RAMP [BFG+16] satisfies RA, COPS [DBC+00] and Eiger-PORT [LSL20] satisfy CC, and Two-Phase Locking implements SER.

Motivation

The complex concurrent behaviors of these protocols call for formal verification that they satisfy the desired consistency model. Moreover, a formal analysis of transactional client programs is also desirable. Different types of formal semantics have been studied for these models.

Declarative semantics have been previously introduced for these models, using dependency graphs [AL99] and abstract executions [CBG15]. This type of semantics is quite concise, but also quite abstract. While it is possible to use abstract executions to prove that a concurrency control protocols implementing its intended consistency model, the complete lack of state information seems to make this type of model unsuitable for proving properties of client programs.

Xiong et al. in [XCRG20] define an *operational semantics* for representing different consistency models in a unified way. This model is a centralized one: the database is represented as a single, multi-versioned key-value store. In reality, the database may be sharded and replicated and each client may have a different view on its current content. These *client views* are explicitly represented in the centralized model as subsets of versions of each key that a given client sees. Transactions are executed atomically and the desired notion of consistency is specified as a so-called execution test, which consists of a condition that must be satisfied for a commit to take place and a constraint on the possible updates of the committing client's view on the database. The commit condition is formulated in terms of the Write-Read (WR) and Write-Write (WW) dependency relations and the Read-Write (RW) anti-dependency relation, first introduced in Adya's PhD thesis [AL99]. This model can be used both for verifying that concurrency control protocols implement their claimed notion of consistency and for analysing client programs' robustness under different consistency models.

Objectives

The main objective of this thesis is to set up a framework for studying the correctness of concurrency control protocols and of client programs using databases based on these protocols. Concretely, we will

- formalize the operational framework for consistency models proposed in [XCRG20] and
- use it to model and prove the correctness of a concurrency control protocol in Isabelle/HOL.

Such a correctness proof must establish that at the time of committing a transaction, the protocol state can be related to an abstract state of the centralized model, which satisfies the execution test of the desired notion of consistency. We will put particular attention on the proof technique for establishing the correctness of such protocols. While the centralized model executes transactions atomically, the concrete protocol executes the constituting read and write operations one-by-one. It is not a priori clear whether standard simulation proof techniques can relate such concrete and abstract executions. We will investigate whether more advanced techniques such as non-atomic refinement [DW03] or vertical implementation [RG01] better suit this purpose.

We propose the following candidate protocols for such a verification effort:

1. The RAMP algorithms [BFG+16], which provide RA (Read Atomicity) and the RYW (Read Your Writes) session guarantee [TDP+94].
2. The Eiger-PORT protocol [LSL20], which satisfies CC (Causal Consistency). This is a recent and novel algorithm published at a top conference.

Both RAMP and Eiger-PORT are state-of-the-art representative implementations of their respective consistency models in the literature.

To our knowledge, this will be the first formalization of a general framework of consistency models and the first general and fully mechanized correctness proof of a concurrency control protocol in such a framework. Depending on our investigation, we might also be able to contribute novel proof techniques for establishing the correctness of such protocols.

Tasks

Our development will be done in Isabelle/HOL, based on definitions, data consistency models, and execution test infrastructure of the ECOOP paper [XCRG20].

1. Study the definitions and semantics as described in [XCRG20], in particular the semantics of transactional and sequential commands and programs, consistency models, and execution tests.
2. Formalize key-value stores, client views, and the semantics of transactional commands and programs in Isabelle/HOL.
3. Formalize the execution tests of the most widely used consistency models for further usage in the formal verification of database concurrency control protocols.
4. Formalize a candidate state-of-the-art protocol and establish its correctness with respective to the corresponding centralized consistency model.
5. **(optional)** Establish the correctness of some instances of the centralized model of [XCRG20] with respect to the corresponding notion of abstract execution, showing that these model instances indeed correctly reflect the intended notion of data consistency.
6. **(optional)** Investigate whether it is possible to refine the centralized model of [XCRG20] into a generic distributed model (similar to [CBG15]) that could be used to simplify the correctness proofs of concrete concurrency control protocols.

Deliverables

Final report The final report must be written in English and should include an introduction, an analysis of related work, and a detailed report of formalizations and proofs.

Isabelle theories for the case study.

Presentation At the end of the thesis a presentation of 30 minutes must be given during an Information Security group seminar. It should give an overview as well as the most important details of the work.

References

[AL99] A Adya and BH Liskov. Weak consistency: A generalized theory and optimistic implementations for distributed transactions (phd thesis). *Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science*, 1999.

[BBG+95] Hal Berenson, Phil Bernstein, Jim Gray, Jim Melton, Elizabeth O'Neil, and Patrick O'Neil. A critique of ansi sql isolation levels. *ACM SIGMOD Record*, 24(2):1–10, 1995.

[BFG+16] Peter Bailis, Alan Fekete, Ali Ghodsi, Joseph M Hellerstein, and Ion Stoica. Scalable atomic visibility with ramp transactions. *ACM Transactions on Database Systems (TODS)*, 41(3):1–45, 2016.

[CBG15] Andrea Cerone, Giovanni Bernardi, and Alexey Gotsman. A framework for transactional consistency models with atomic visibility. In Luca Aceto and David de Frutos-Escrig, editors, *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, volume 42 of *LIPIcs*, pages 58–71. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

[DBC+00] David Durham, Jim Boyle, Ron Cohen, Shai Herzog, Raju Rajan, and Arun Sastry. The cops (common open policy service) protocol, 2000.

[DW03] John Derrick and Heike Wehrheim. Using coupled simulations in non-atomic refinement. In Didier Bert, Jonathan P. Bowen, Steve King, and Marina Waldén, editors, *ZB 2003: Formal Specification and Development in Z and B, Third International Conference of B and Z Users, Turku, Finland, June 4-6, 2003, Proceedings*, volume 2651 of *Lecture Notes in Computer Science*, pages 127–147. Springer, 2003.

[LFKA11] Wyatt Lloyd, Michael J Freedman, Michael Kaminsky, and David G Andersen. Don't settle for eventual: Scalable causal consistency for wide-area storage with cops. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pages 401–416, 2011.

[LSL20] Haonan Lu, Siddhartha Sen, and Wyatt Lloyd. {Performance-Optimal}{Read-Only} transactions. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, pages 333–349, 2020.

[Pap79] Christos H Papadimitriou. The serializability of concurrent database updates. *Journal of the ACM (JACM)*, 26(4):631–653, 1979.

[RG01] Arend Rensink and Roberto Gorrieri. Vertical implementation. *Inf. Comput.*, 170(1):95–133, 2001.

[SPAL11] Yair Sovran, Russell Power, Marcos K Aguilera, and Jinyang Li. Transactional storage for geo-replicated systems. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pages 385–400, 2011.

[TDP+94] Douglas B. Terry, Alan J. Demers, Karin Petersen, Mike Spreitzer, Marvin Theimer, and Brent B. Welch. Session guarantees for weakly consistent replicated data. In *PDIS*, pages 140–149. IEEE Computer Society, 1994.

[XCRG20] Shale Xiong, Andrea Cerone, Azalea Raad, and Philippa Gardner. Data consistency in transactional storage systems: A centralised semantics. In Robert Hirschfeld and Tobias Pape, editors, *34th European Conference on Object-Oriented Programming, ECOOP 2020, November 15-17, 2020, Berlin, Germany (Virtual Conference)*, volume 166 of *LIPIcs*, pages 21:1–21:31. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.