Automated Synthesis of Comprehensive Distributed Consistency Model Litmus Test Suites[*]

Xue Jiang
State Key Laboratory for Novel
Software Technology
Nanjing University
Nanjing, China
xxx@smail.nju.edu.cn

Hengfeng Wei[*]
State Key Laboratory for Novel
Software Technology
Nanjing University
Nanjing, China
hfwei@nju.edu.cn

Yu Huang
State Key Laboratory for Novel
Software Technology
Nanjing University
Nanjing, China
yuhuang@nju.edu.cn

. . .

. . .

ABSTRACT

PVLDB Reference Format:
Xue Jiang, Hengfeng Wei*, Yu Huang, . . . , and . . . . Automated Synthesis of Comprehensive Distributed Consistency Model Litmus Test Suites. PVLDB, 14(1): XXX-XXX, 2020.
doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at URL_TO_YOUR_ARTIFACTS.

1 INTRODUCTION

Motivations. Distributed consistency models are quite tricky to understand. Even worse, in the literature, there are often several variants of a specific consistency model. For example, Jiang et al. presents six variants of causal consistency in [2] for non-transactional databases. Crooks presents a hierarchy of eight variants of snapshot isolation [1] for transactional databases. It is difficult, even for experts,

Consistency Checking: to check whether a given history satisfies some consistency model or not;

Test-case Generating: to come up with histories under some constraints (e.g., on size) that satisfy or refuse some consistency model; and

Model Comparing: to tell the differences between two consistency models by presenting distinguishing histories that satisfy one consistency model but refuse the other.

All the three tasks are concerned about histories, which are easier for human to understand.

programs vs. litmus tests vs. histories vs. abstract executions

Challenges.

- How to formally express consistency models?
- generating too many (redundant) histories

Our Contributions.

- Alloy$\ast$ model of both non-transactional and transactional consistency models
- Two case studies

2 PRELIMINARIES

2.1 The Alloy and Alloy* Modelling Languages

2.2 The $(\mathrm{VIS}, \mathrm{AR})$ Specification Framework

2.2.1 *Non-transactional Consistency Models.*

2.2.2 *Transactional Consistency Models.*

3 OVERVIEW

4 APPROACH

5 CASE STUDIES

5.1 Causal Consistency Variants

5.1.1 *The Alloy* Model.*

5.1.2 *Consistency Checking.*

5.1.3 *Test-case Generating.*

5.1.4 *Model Comparing.*

5.2 Snapshot Isolation Variants

5.2.1 *The Alloy* Model.*

5.2.2 *Consistency Checking.*

5.2.3 *Test-case Generating.*

5.2.4 *Model Comparing.*

6 RELATED WORK

Consistency Checking. MemSAT [4]: "Given an axiomatic specification of a memory model and a multi-threaded test program containing assertions, MemSAT outputs a trace of the program in which both the assertions and the memory model axioms are satisfied, if one can be found. If it cannot find a trace, it outputs a minimal subset of the memory model and program constraints that are unsatisfiable."

Test-case Generating.

Model Comparing. [3]: "systematically comparing hardware memory models specified using both operational and axiomatic

[*] Corresponding author. Hengfeng Wei is also with Software Institute at Nanjing University.

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

styles. When the models differs, the tool finds a minimal "litmus test" program that demonstrates the difference."

7 CONCLUSION

REFERENCES

[1] Natacha Crooks, Youer Pu, Lorenzo Alvisi, and Allen Clement. 2017. Seeing is Believing: A Client-Centric Specification of Database Isolation. In *Proceedings of the ACM Symposium on Principles of Distributed Computing (PODC '17)*. Association for Computing Machinery, New York, NY, USA, 73–82. https://doi.org/10.1145/3087801.3087802

[2] Xue Jiang, Hengfeng Wei, and Yu Huang. 2020. A Generic Specification Framework for Weakly Consistent Replicated Data Types. In *2020 International Symposium on Reliable Distributed Systems (SRDS)*. 143–154. https://doi.org/10.1109/SRDS51746.2020.00022

[3] Sela Mador-Haim, Rajeev Alur, and Milo M. K. Martin. 2010. Generating Litmus Tests for Contrasting Memory Consistency Models. In *Computer Aided Verification*, Tayssir Touili, Byron Cook, and Paul Jackson (Eds.). Springer Berlin Heidelberg, 273–287.

[4] Emina Torlak, Mandana Vaziri, and Julian Dolby. 2010. MemSAT: Checking Axiomatic Specifications of Memory Models. In *Proceedings of the 31st ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '10)*. Association for Computing Machinery, New York, NY, USA, 341–350. https://doi.org/10.1145/1806596.1806635