Jepsen: quick intro

Vojtěch Juránek

Red Hat

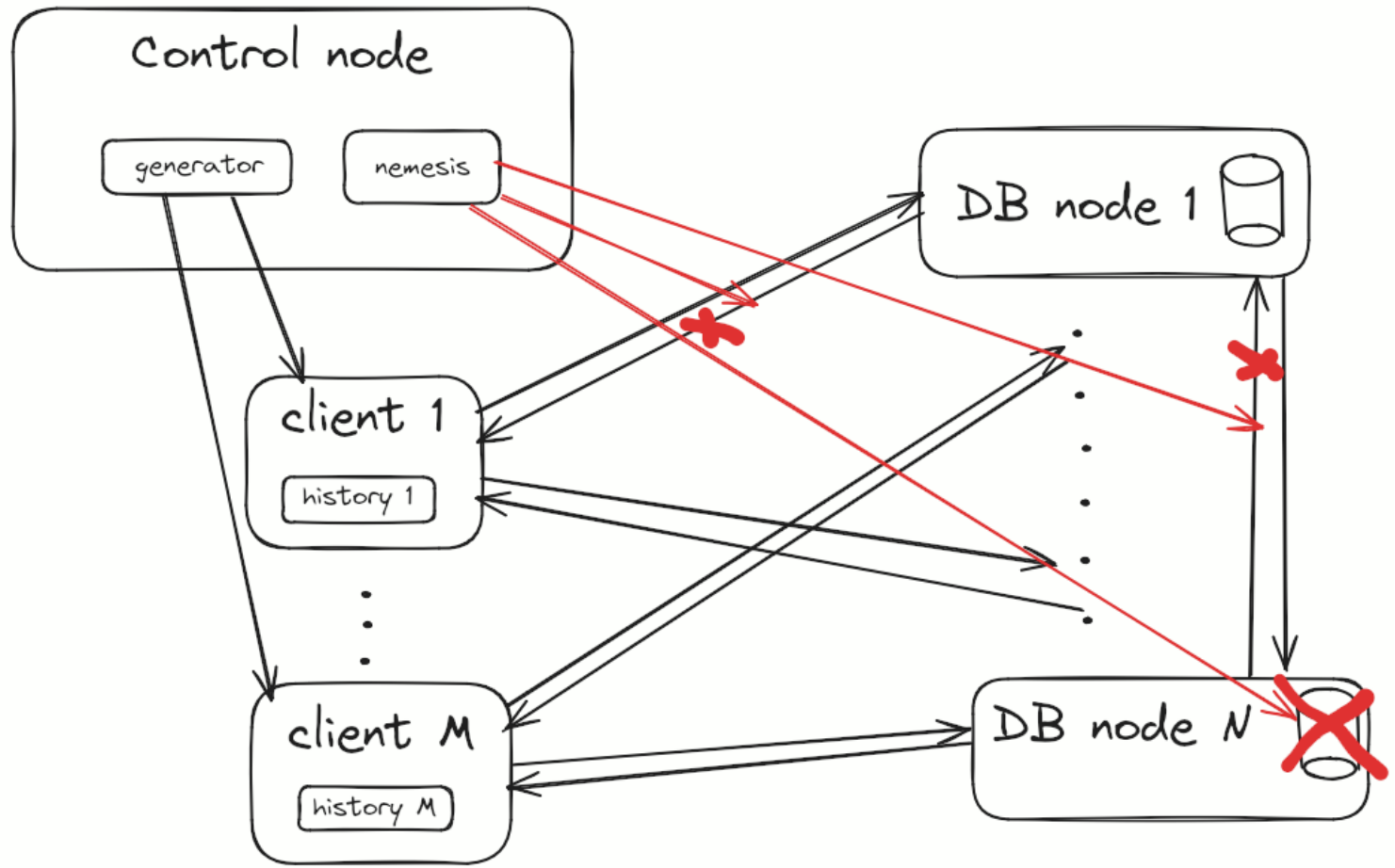
May 23rd 2024, Debezium F2F meeting, Brno

Jepsen

- *Breaking distributed systems so you don't have to.*
- *Jepsen is an effort to improve the safety of distributed databases, queues, consensus systems, etc.*

- A test library written in Clojure.
- Makes easy to setup and test distributed system in a test.
- Developed by Kyle Kingsbury, a.k.a "Aphyr".

Jepsen architecture



- Transaction histories are analyzed by the checker once the test finishes.
- Main work of the checker is done by Elle.

Elle

- Elle is a transaction consistency checker.
- Search for cycled in transaction dependencies graph.

Appears in the *Proceedings of the IEEE International Conference on Data Engineering*, San Diego, CA, March 2000

Generalized Isolation Level Definitions

Atul Adya
Microsoft Research,
1 Microsoft Way,
Redmond, WA 98007
adya@microsoft.com

Barbara Liskov
Laboratory for Computer Science,
Massachusetts Inst. of Technology,
Cambridge, MA 02139
liskov@lcs.mit.edu

Patrick O'Neil
Univ. of Massachusetts,
Boston, MA 02125-3393
poneil@cs.umb.edu

Abstract

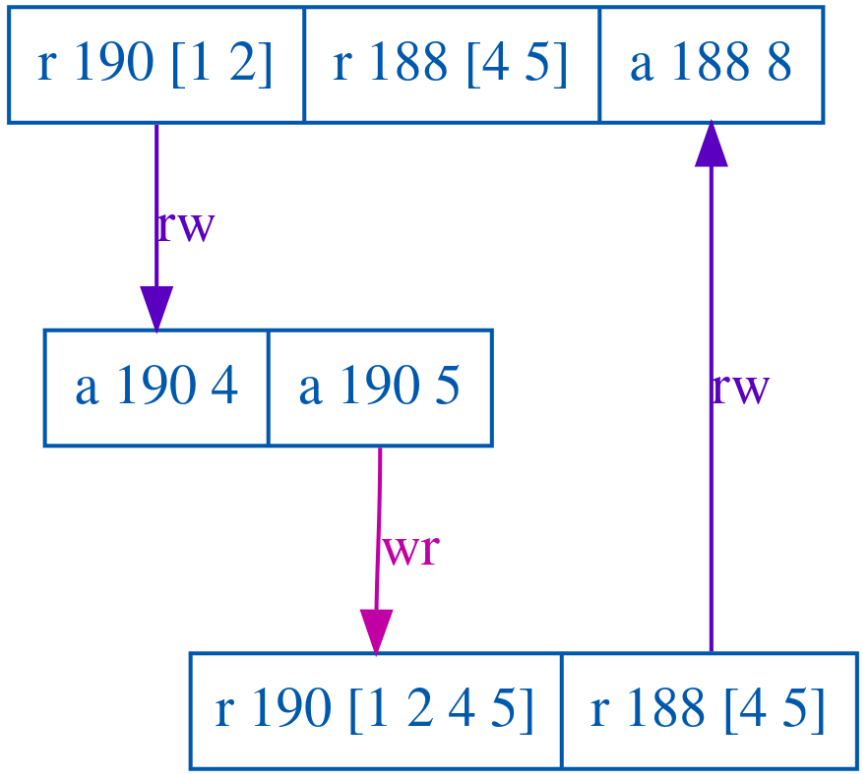
Commercial databases support different isolation levels to allow programmers to trade off consistency for a potential gain in performance. The isolation levels are defined in the current ANSI standard, but the definitions are ambiguous and revised definitions proposed to correct the problem are too constrained since they allow only pessimistic (locking) implementations. This paper presents new specifications for the ANSI levels. Our specifications are portable*; they apply not only to locking implementations, but also to optimistic and multi-version concurrency control schemes. Furthermore, unlike earlier definitions, our new specifications handle predicates in a correct and flexible manner at all levels.*

in [13] and some refinements suggested by [11] set the stage for the ANSI/ISO SQL-92 definitions for isolation levels [6], where the goal was to develop a standard that was implementation-independent. However, a subsequent paper [8] showed that the definitions provided in [6] were ambiguous. That paper proposed different definitions that avoided the ambiguity problems, but, as stated in [8], these definitions were simply "disguised versions of locking" and therefore disallow optimistic and multi-version mechanisms. Thus, these definitions fail to meet the goals of ANSI-SQL with respect to implementation-independence.

Thus, we have a problem: the standard is intended to be implementation-independent, but lacks a precise definition that meets this goal. Implementation-independence is important since it provides flexibility to implementors, which can lead to better performance. Optimism can outperform

Example: Postgres G2-item

- G2-item in Postgres, violates repeatable-read isolation level.
- See PostgreSQL 12.3 jepsen report and Kleppmann's Hermitage: Testing transaction isolation levels



Source: `https://jepsen.io/analyses/postgresql-12.3`

Practical

- Jepsen is supported on Linux bare metal, VMs and LXC.
- Docker is available, but not supported.
- Available very good tutorial.
- Clojure makes the development harder, but the actual hard thing is to nail down the right testing scenarios which reveal the issues.

Jepsen tests for Debezium

- Could find issues like recent DBZ-7816.
- There are tests for Postgres and MySQL- we should be able to reuse at least some parts of the code.

Resources

- https://jepsen.io/
- Black-box Isolation Checking with Elle
- Hermitage: Testing transaction isolation levels
- Maelstrom a workbench for learning distributed systems by writing your own

Thank you!

Questions?