Quantifying Integrity Violations

Nikolay Amiantov,
Original work
"Coordination Avoidance in Distributed Databases"
by Peter David Bailis, Ph.D.

February 28, 2018

Concurrency control in Ruby on Rails

- Transactions;
- Locking;
- Validations and associations

"I consider stored procedures and constraints vile and reckless destroyers of coherence. No, Mr. Database, you can not have my business logic. Your procedural ambitions will bear no fruit and you'll have to pry that logic from my dead, cold object-oriented hands." D.H.H.

Uniqueness validation problem

1. $i.\alpha = k.\alpha$;
2. SELECT 1 FROM T_m WHERE $C_\alpha = i.\alpha$ LIMIT ONE;

5. INSERT INTO T_m VALUES i;

Uniqueness validation problem

1. $i.\alpha = k.\alpha$;
2. SELECT 1 FROM T_m WHERE $C_\alpha = i.\alpha$ LIMIT ONE;
3. SELECT 1 FROM T_m WHERE $C_\alpha = k.\alpha$ LIMIT ONE;
4. INSERT INTO T_m VALUES k;
5. INSERT INTO T_m VALUES i;

Association validation problem

1. SELECT 1 FROM T_m WHERE $C_\alpha = i.\alpha$ LIMIT ONE;

3. INSERT INTO T_m VALUES i;

Association validation problem

1. SELECT 1 FROM T_m WHERE $C_\alpha = i.\alpha$ LIMIT ONE;
2. DELETE FROM T_m WHERE $C_\alpha = i.\alpha$;
3. INSERT INTO T_m VALUES i;

Experiment setup

Configuration: 8 CPU cores, 68.4 GB RAM

- Application server:
  - Ruby VM pool (Unicorn), varying number of workers;
  - NGINX web frontend.
- Database server: PostgreSQL
- Stress testing server

Experiments:

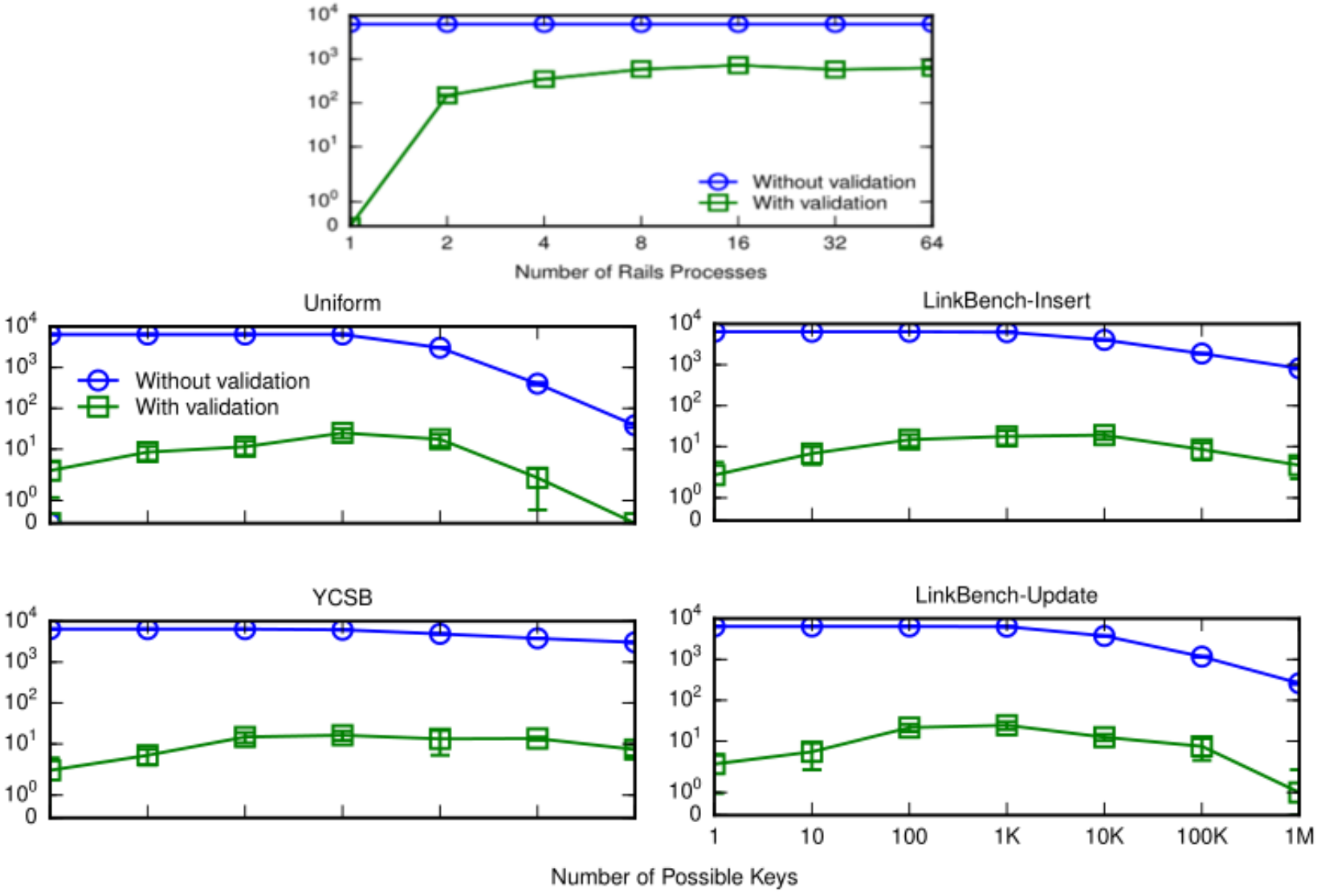
- Uniqueness: try to insert duplicate keys
- Association: insert association while deleting associated entry

PostgreSQL bug

Found during testing of serializable transactions isolation level

1. "When issuing a high-contention workload under SERIALIZABLE isolation, I am able to produce non-serializable outcomes."
2. "I have further evidence, and have narrowed the area of investigation, but I haven't been able to pin down the cause yet." PostgreSQL contributor

Uniqueness stress-testing



Association stress-testing

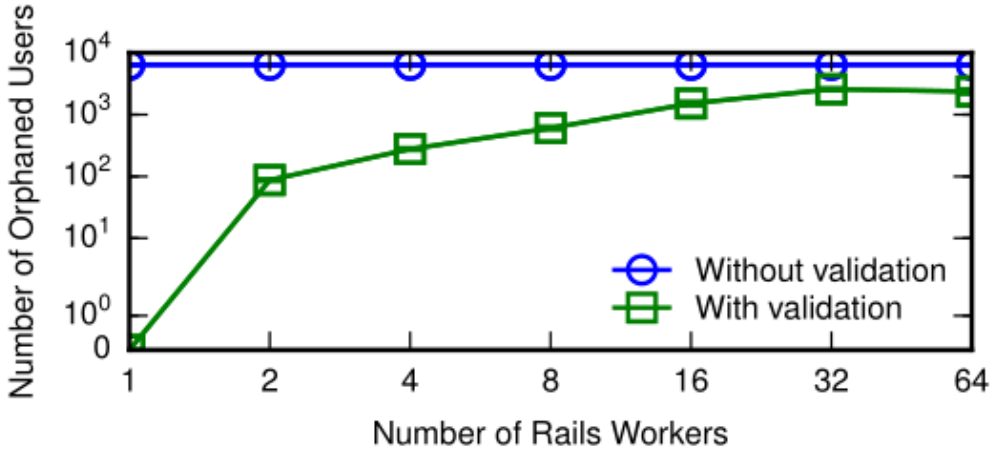
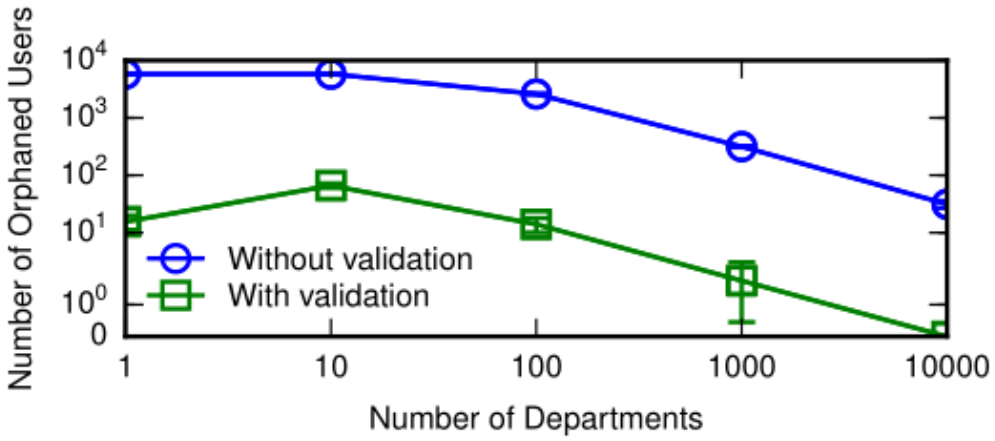


Figure 6.8: Foreign key stress association anomalies.



Not a bug

- Database constraints and/or stored procedures make the validation mechanisms database-dependent and can make testing and maintenance more difficult. However, if your database is used by other applications, it may be a good idea to use some constraints at the database level. Additionally, database-level validations can safely handle some things (such as uniqueness in heavily-used tables) that can be difficult to implement otherwise.

why-el commented on Dec 9, 2013 — Contributor

This is not a bug but documented and inherent behavior of validates_uniqueness_of.

pixeltrix commented on Dec 9, 2013 — Owner

@daveroberts this isn't a thread safety issue, more of a general concurrency issue and it's not limited to threaded servers - it can happen with any server setup. As **@arkiver** points out, the only way to handle this properly is at the database layer with a unique constraint on the column. What you then do is to rescue `ActiveRecord::RecordNotUnique` errors and handle them in a way that's suitable for your application.

 1

Consistency enforcement in other frameworks

Framework	Database level	Application level ("feral" 🐯)
Hibernate 4.3.1	Yes	Yes
CakePHP 2.5.5	No	Yes
Laravel 4.2	Manual	Yes
Waterline 0.10	Yes	Yes

Conclusion and advice

- From database developers:
  - Idiomatic invariants in programming language of choice;
  - Domestication of feral mechanisms;
  - Better defaults for coordination levels.
- From ORM developers:
  - Use database-level invariants where possible.

Questions?