

# How to Get Online Transactional Schema Evolution (almost) for Free in Snapshot Databases

Tianzheng Wang



Joint work with Tianxun Hu (SFU) and Qingqing Zhou (Tencent America)

To appear at VLDB 2023

# Modern DB apps (+ their schemas) evolve continuously



Social network

**NETFLIX**

Online streaming



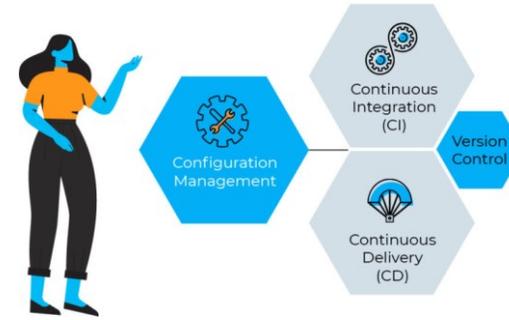
Image hosting/  
social media



**OANDA**

Finance

# Modern DB apps (+ their schemas) evolve continuously



New code → Testing/code review → Release engineering → Actual deployment



DB schema changes often needed, frequently

Via “data definition language” (DDL) statements:  
ALTER TABLE t\_name DROP COLUMN c\_name...  
ALTER TABLE t\_name MODIFY COLUMN c\_name SET  
DEFAULT value...  
CREATE TABLE...AS

# But modern DBMSs don't support DDL well

a concerted engineering effort. However, OANDA's database schema changes always were **always ad hoc and full of fear.**



## As a result...

- Schema evolution often involves service downtime

## But...

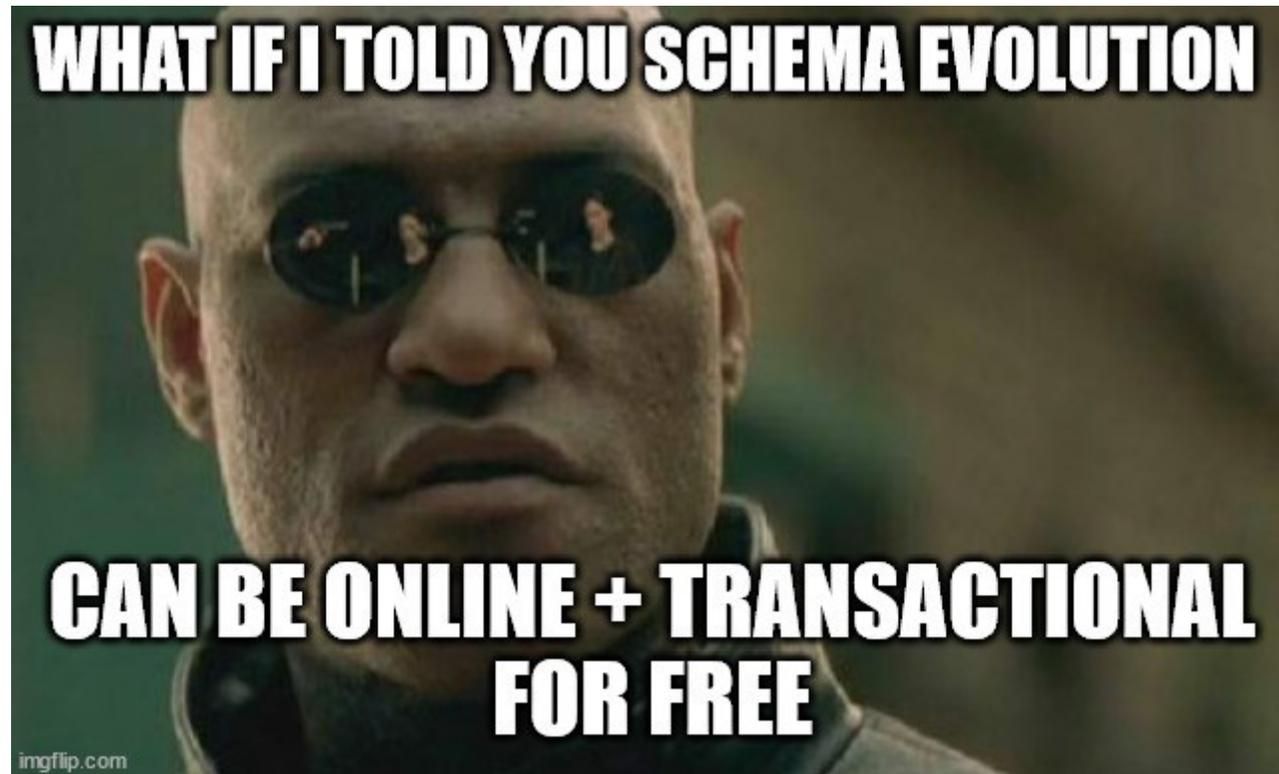
- They are frequent
- Things could go wrong
  - E.g., VARCHAR to INT
    - Must be able to rollback in case of “illegal” characters

**Wanted: online + transactional schema evolution**

\* Continuous Deployment at Facebook and OANDA, ICSEC 2016.

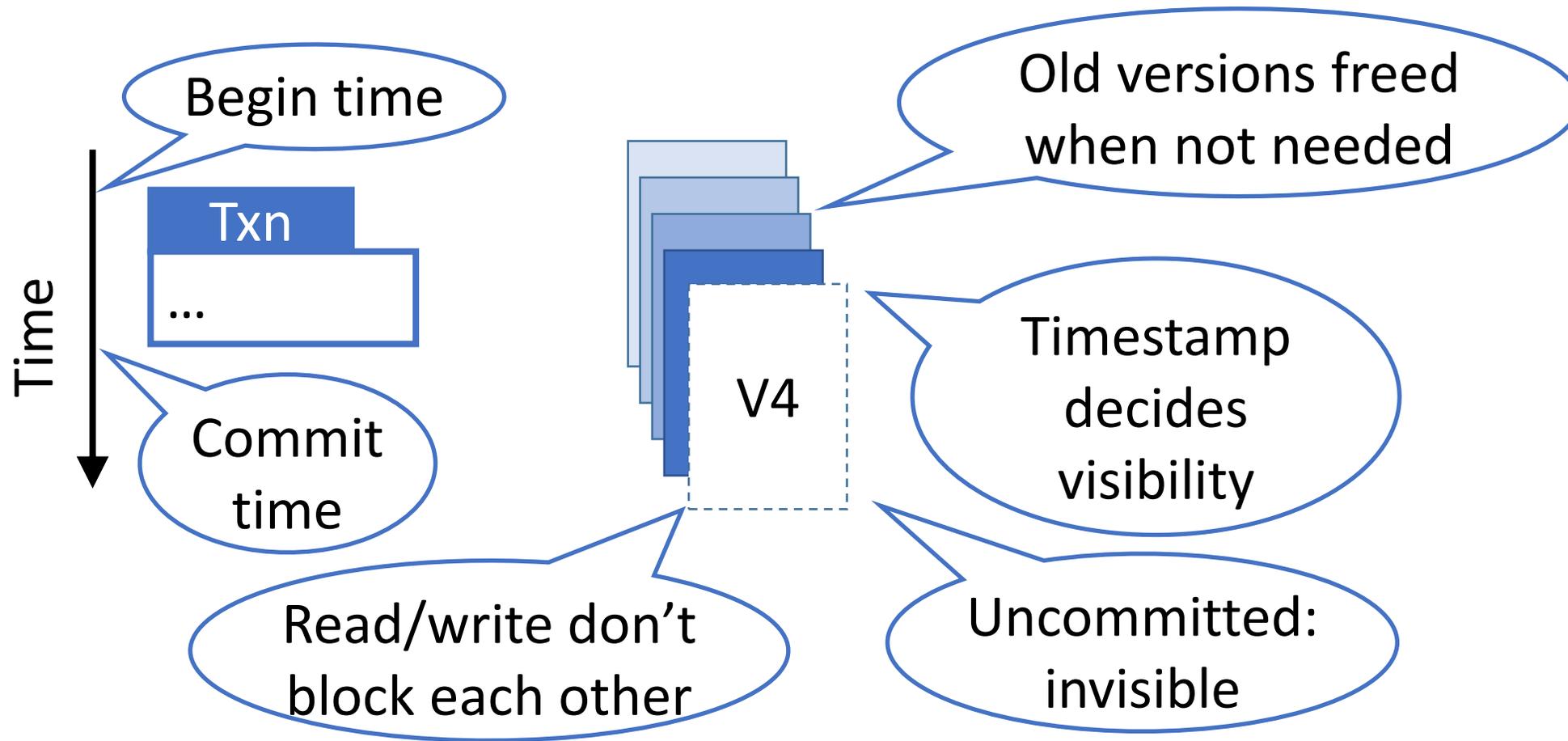
# But can't I just **ALTER TABLE**...?

- YMMV depending on the DBMS
- DBMS often miss transactional/online DDL infrastructure
- Existing solutions: “ad hoc patches”
  - E.g., using triggers, views. DB apps themselves!
  - Often partial support, many corner cases, need careful use
    - MySQL: any DDL in a transaction? I'm committing it for you first! → Correctness issues



\* fine print: for snapshot databases

# Snapshot Isolation: A Primer



Adopted by almost all major OLTP DBMSs

# (Basic) Data-Definition-as-Manipulation (DDaM)

- DDL operation === DML operation that happens to modify entire tables
- Leverage snapshot isolation for multi-schema-versioning
- A table = (many) data records + (one) schema record
  - Both multi-versioned, subject to snapshot isolation visibility rules

```
START TRANSACTION
ALTER TABLE ...
  foreach record in Table:
    update with new schema
COMMIT
```

Trivial, eh?

# DDaM Challenge: Conflicts

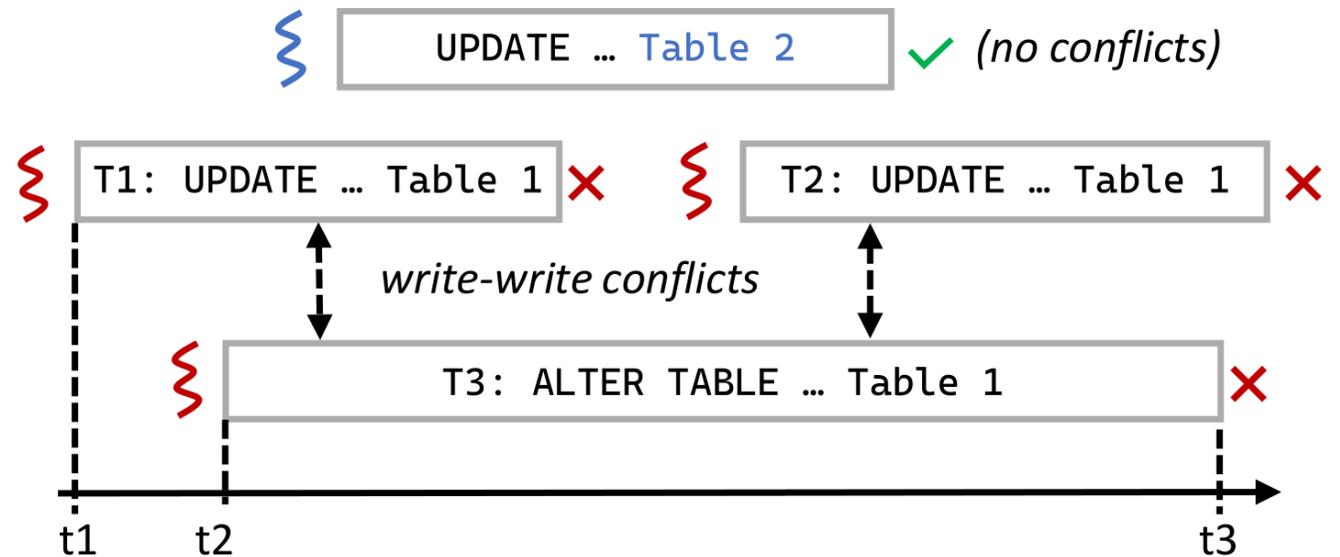
START TRANSACTION

ALTER TABLE ...

foreach record in Table:  
update with new schema

COMMIT

A very long  
transaction!



→ Write-conflict with everybody else, massive aborts

→ DDL write set is too big, hard (if not impossible) to maintain

Everybody loses!

# Tesseract to the Rescue

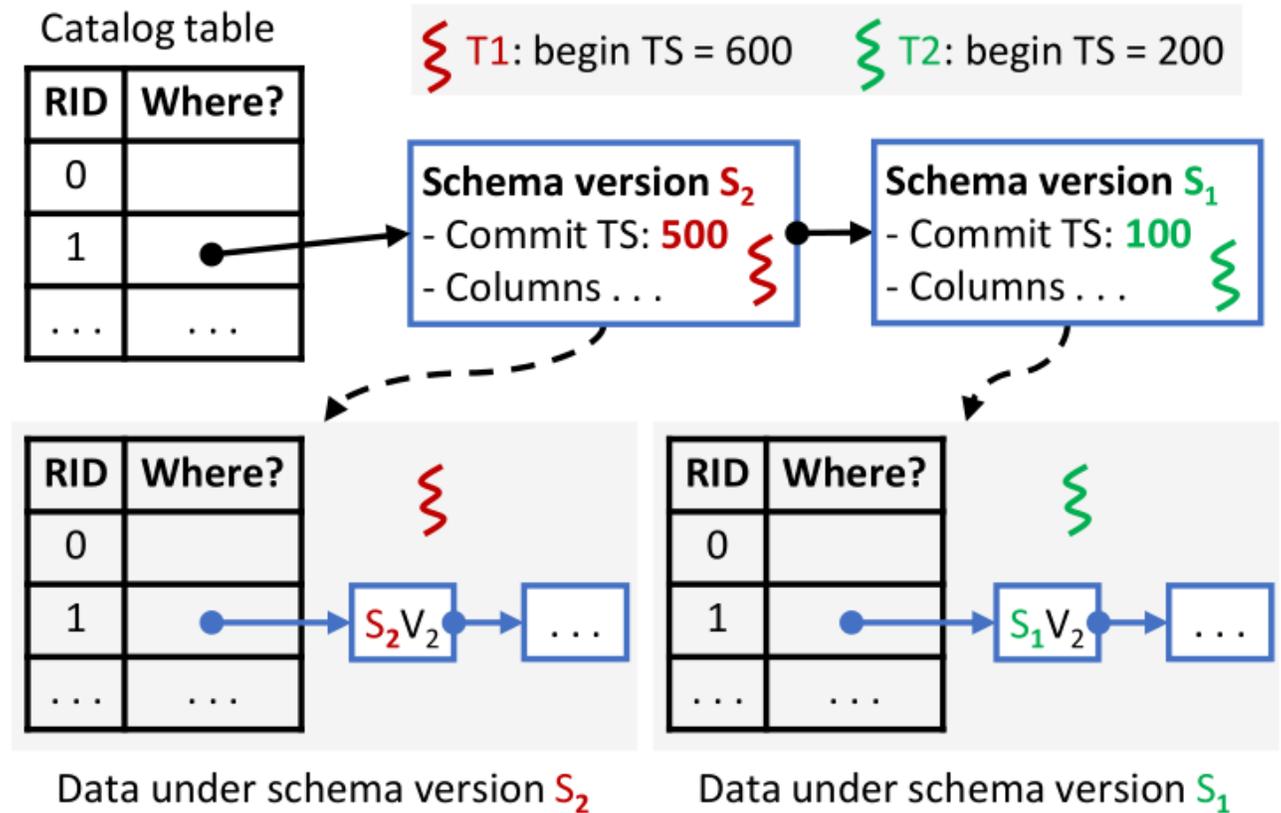
## Still DDaM, but +:

- Our-of-place migration
- Parallel change data capture (CDC)
- Relaxed snapshots

All at the engine level by tweaking concurrency control protocols (SI)

# Tesseract Schema Versioning

- DDL works on a new table
  - Invisible to concurrent DML
- Concurrent DML uses old table
  - But held up upon commit
  - (or allowed to commit if DML is prioritized → DDL may abort)
  - Resolve conflict later (CDC)



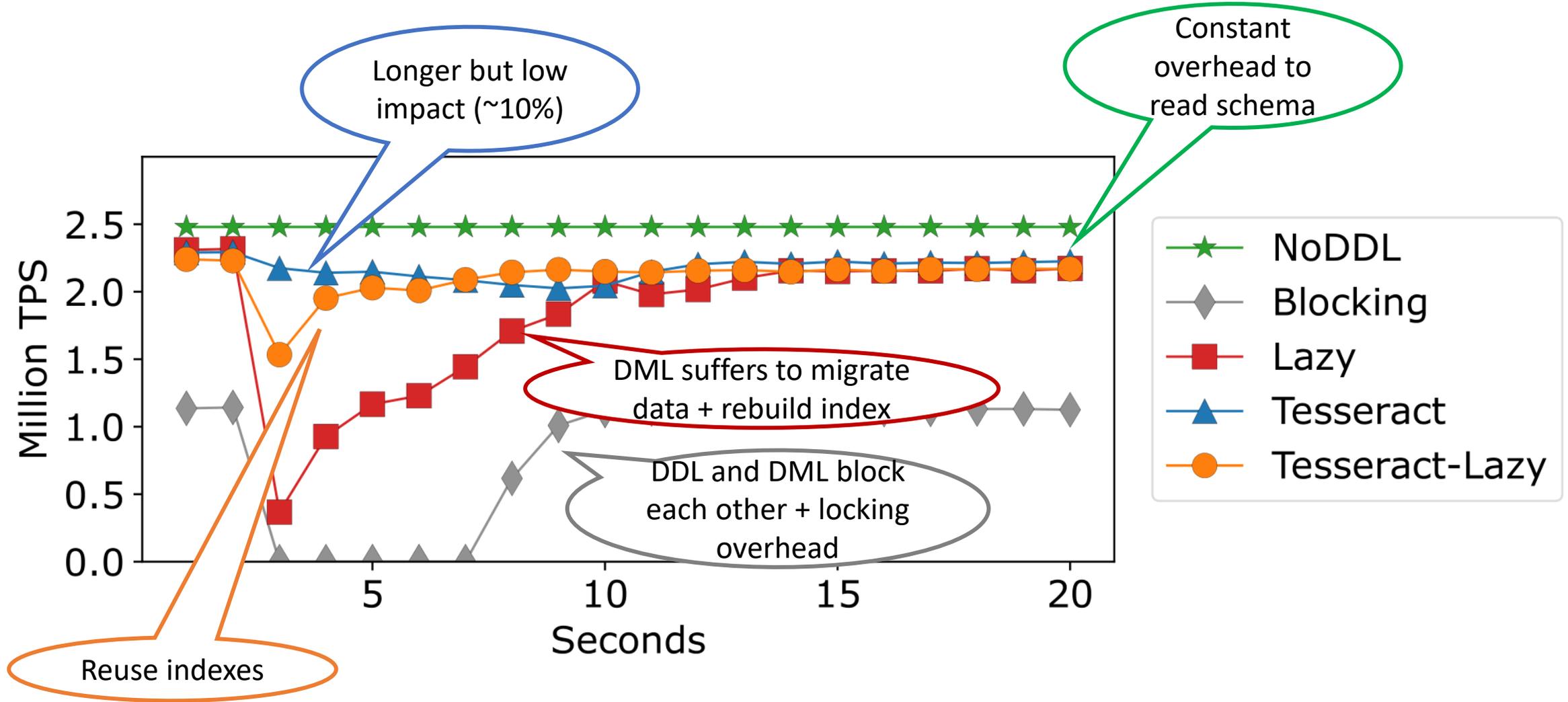
**Write-write conflicts delayed**

(most DDL changes are compatible anyway)

# Evaluation

- Hardware
  - 2 x Xeon Gold 6242R (40 cores across two sockets)
  - 384GB main memory
- Prototype based on ERMIA [SIGMOD 2016]
- Methodology
  - Run DML benchmarks as usual
  - Issue a DDL transaction after two seconds the benchmark started
- Benchmarks
  - YCSB + TPC-C with additional DDL transactions

# Microbenchmark (Add Column)



# Summary

- Modern database applications need transactional + online DDL
  - They often change schemas, >1 version may need to co-exist
  - Need CD/CI → The less human/DBA intervention, the better
- Key observation: DDL is a DML that modifies full tables
  - Data-Definition-as-Manipulation (DDaM)
- Tesseract: transaction + online DDL at the engine level
  - DDaM under snapshot isolation
  - Relaxed DDaM: tweak SI protocol to extract more concurrency
- Code: <https://github.com/sfu-dis/tesseract>
- Find out more in our paper
  - Online Schema Evolution is (Almost) Free for Snapshot Databases, VLDB 2023

*Thank you!*

# Two Final Plugs

Students volunteering opportunity  
for VLDB 2023

**Free registration!**



<https://forms.office.com/r/4HL2X72mQN>

**New**

Workshop on Simplicity in  
Management of Data (SiMoD) at  
SIGMOD 2023 (Jun 23)



<https://sfu-dis.github.io/simod>