TAOBench

Audrey Cheng, Xiao Shi, Aaron Kabcenell, Shilpa Lawande,
Hamza Qadeer, Jason Chan, Harrison Tin, Ryan Zhao,
Peter Bailis, Mahesh Balakrishnan, Nathan Bronson, Natacha Crooks, Ion Stoica

riselab UC Berkeley

Meta

Introduction

Audrey Cheng, PhD student in riselab UC Berkeley
Advised by Natacha Crooks and Ion Stoica

Research on transaction processing for databases

- RAMP-TAO in VLDB ‘21 (Best Industry Paper Award)

TAOBench: a new benchmark for social networks based on Meta production workloads (VLDB ‘22)!

How can TAOBench be **useful** to LDBC?

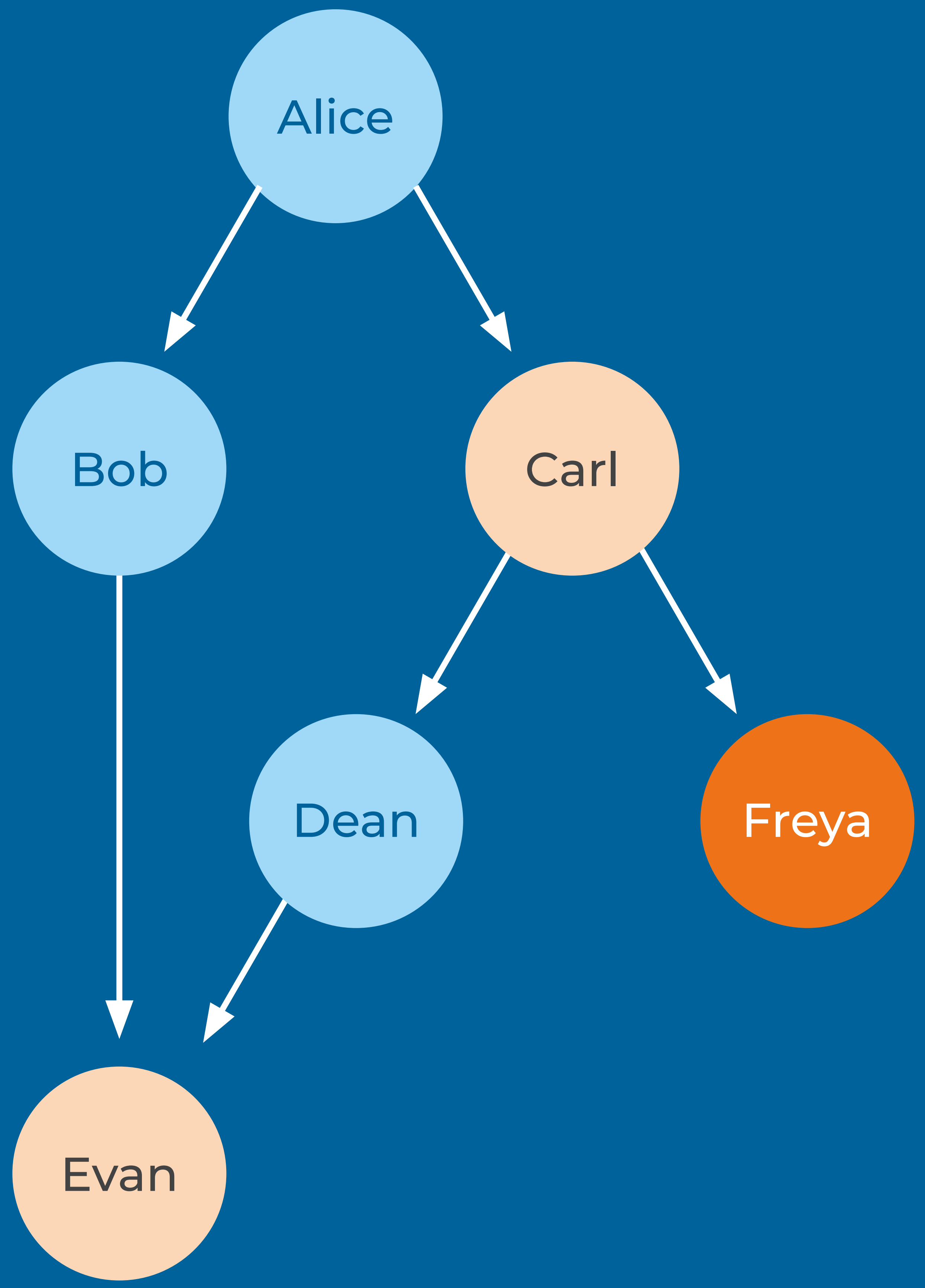
Social Networks

Ubiquitous!

- Meta, Twitter, LinkedIn, WeChat, ByteDance (TikTok)

Supported by large-scale, geo-distributed **data stores**

- TAO, Manhattan, Voldemort, PaxosStore, ByteGraph



Social Network Benchmarks?

Lack of **publicly available, realistic** workloads

- Difficult to understand limits of existing systems
- Challenging evaluate new features and mechanisms

What are the **properties** should be captured by the workloads of a social network benchmark?

Desired Properties

Derived from production traces

To the best of our knowledge, only 1 exists: LinkBench from Meta

Captures any transactional requirements

Single-shot, multi-key semantics for improved performance and scalability

Expresses colocation constraints

Sharding can reflect user intent, privacy constraints, or regulatory compliance

Models request distributions without prescriptive query types

Represent workloads via distributions for adaptability and flexibility

Exhibits behavior of multiple tenants

Product groups can exhibit coordinated behavior

facebookarchive/ **linkbench**

Facebook Graph Benchmark

Benchmark released in 2014

- Derived from partial production trace (**excluding requests that hit cache**)
  - Single MySQL instance
- No **graph-level** transactions
- No information about **colocation** preferences and constraints

LDBC Social Network Benchmark

Important workload for graph databases

- More processing-intensive rather than serving

How can we **supplement** this workload with TAOBench?

Agenda

1. Characterizing the Social Network
2. Benchmark Details
3. Distributed DB Evaluation

Social Network Workload

01

A benchmark is only as useful as the workloads from which it is derived

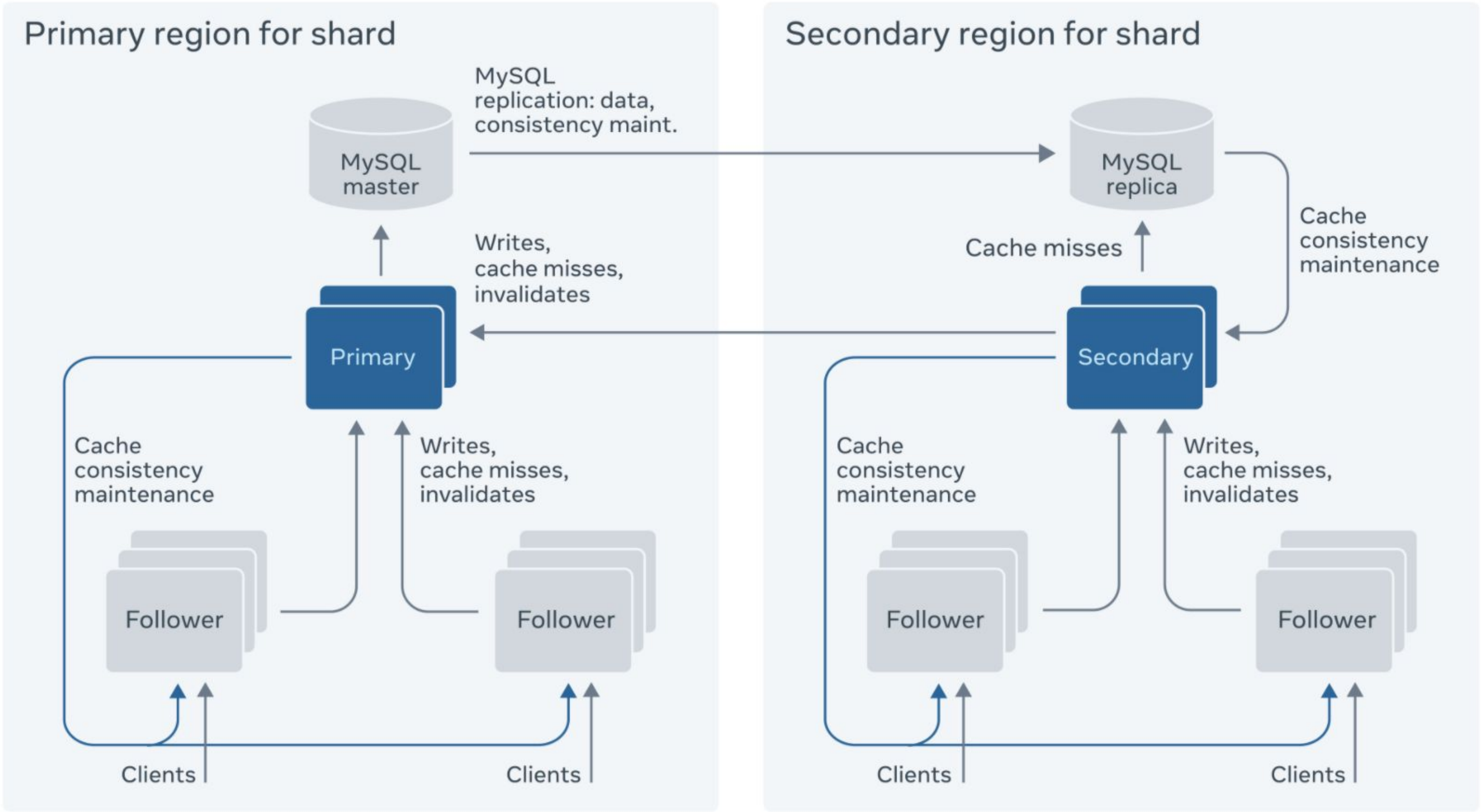
TAO @ Facebook

Diverse products: Underlies many applications

Huge scale: >10B reads and >10M writes per second

Simple graph API: Do a few things well as scale

Eventually consistent*: High availability and low latency



TAO's Workload Satisfies All 5 Properties

Derived from production traces

Support majority of social graph requests for Meta's 3.6 billion monthly active users

Captures any transactional requirements

Failure-atomic write transactions and read-only transactions (RAMP-TAO)

Expresses colocation constraints

Applications can choose to explicitly colocate data in the MySQL layer

Models request distributions without prescriptive query types

With 10K+ query types per day, distributions are needed to model the full workload

Exhibits behavior of multiple tenants

Many applications and other infrastructures layered on top of TAO

Collecting Production Data

Analyze traces collected over **3 days**

- Distributions do not vary significantly between different periods

Uniformly sample over objects (nodes) and associations (edges)

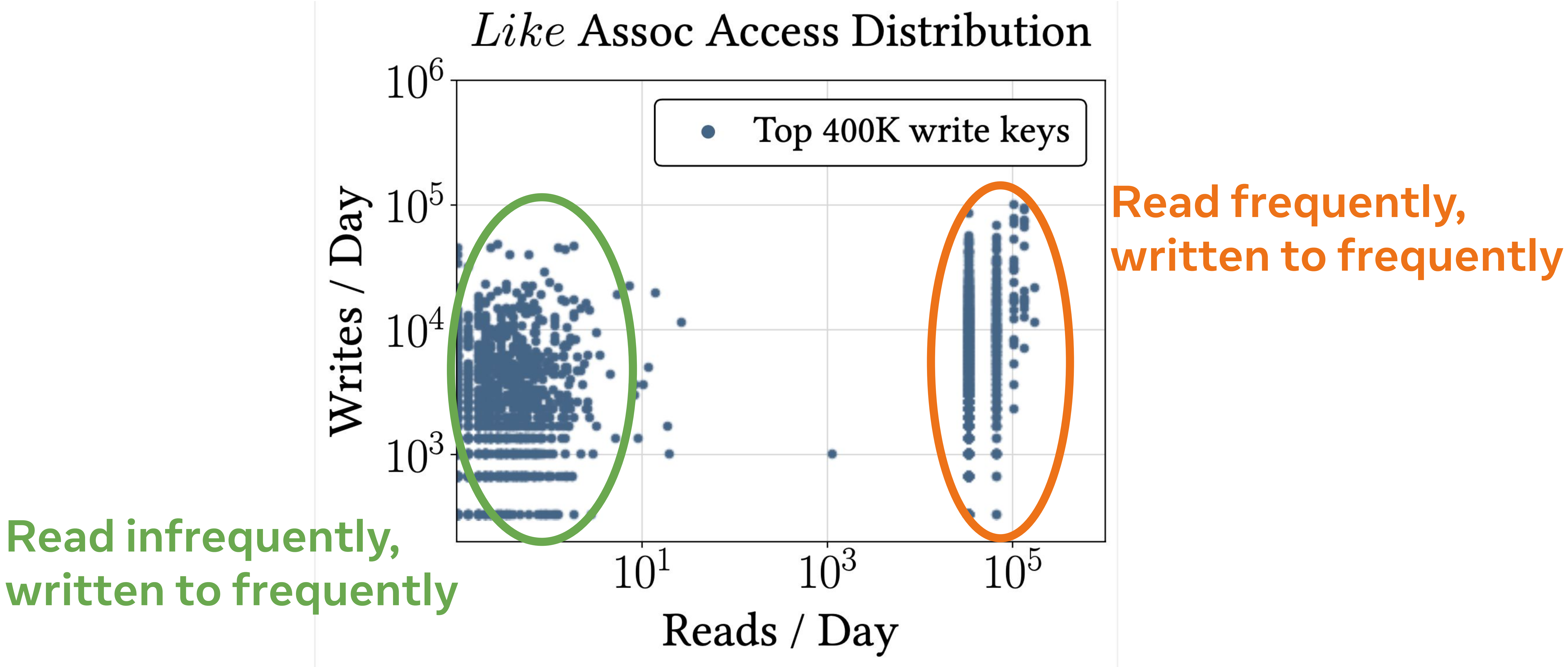
- Capture all requests that touch these items
- No conflicts on these keys are missed

⇨ 99.7% reads, 0.2% writes, and 0.01% write transactions

Read and Write Hotspots

Read and write hotspots occur on **different keys**

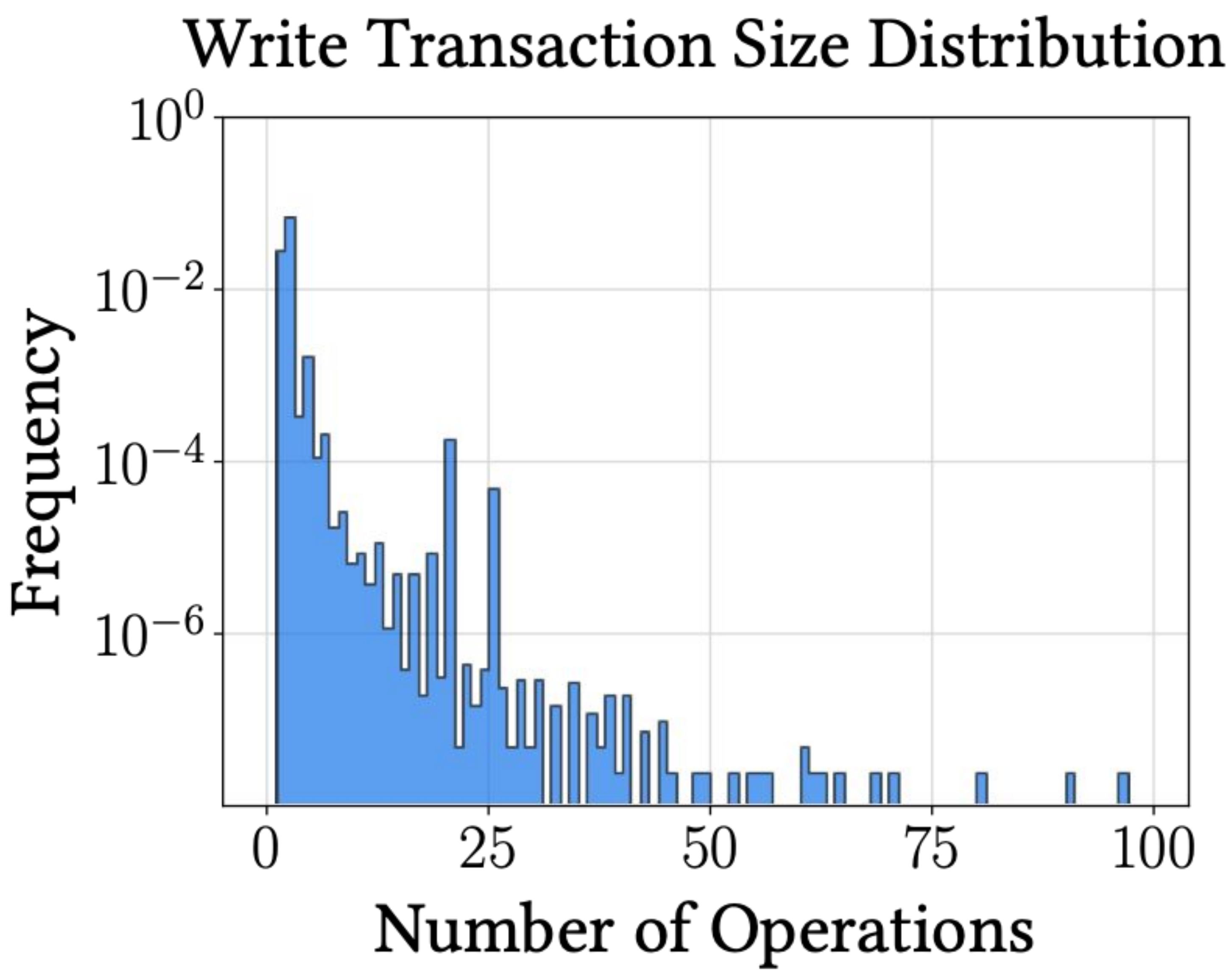
- Only 0.1% of the top 400K keys overlap

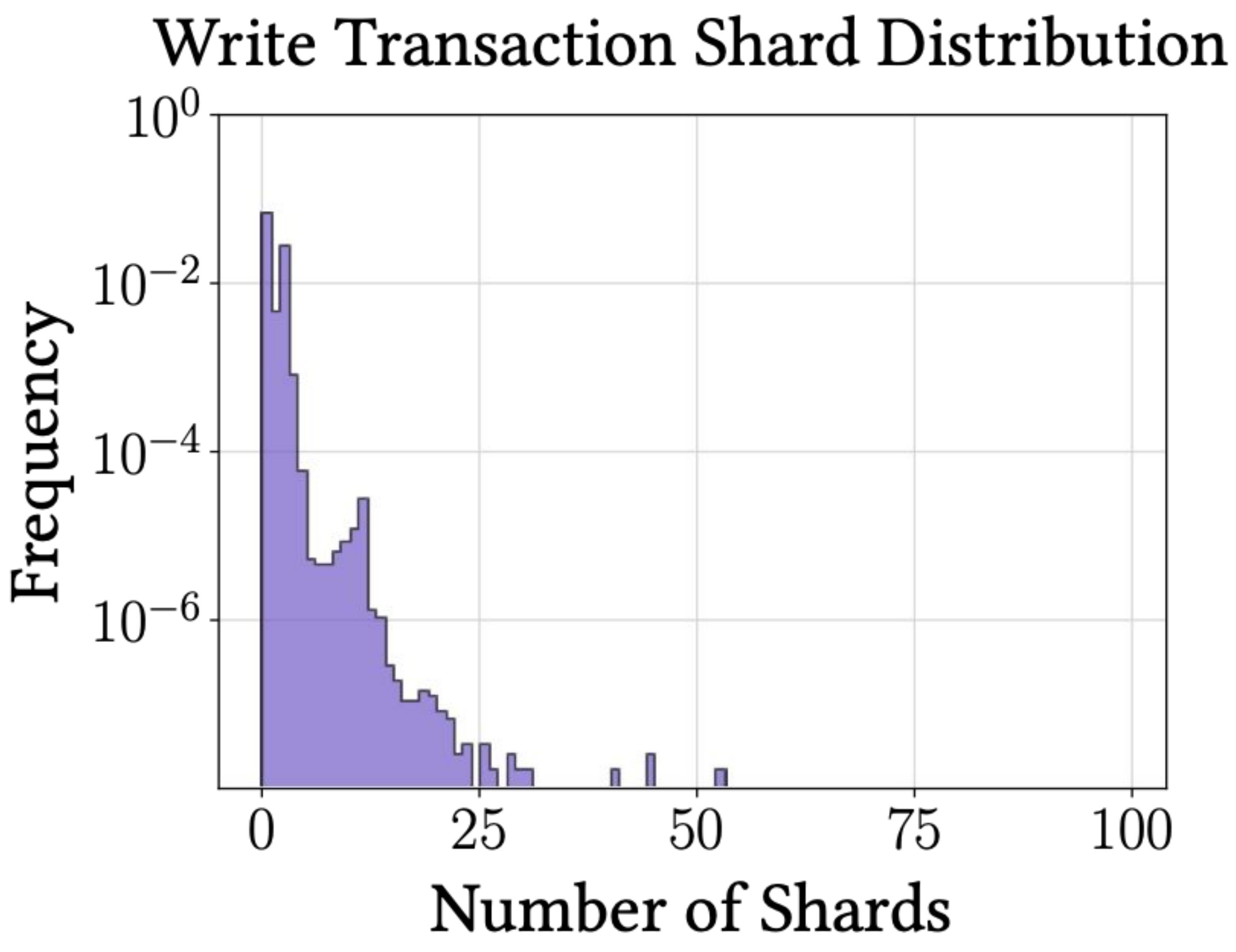


Transaction Size

Some transactions involve **many items**

- Most of these undergo an optimized protocol (described in RAMP-TAO)





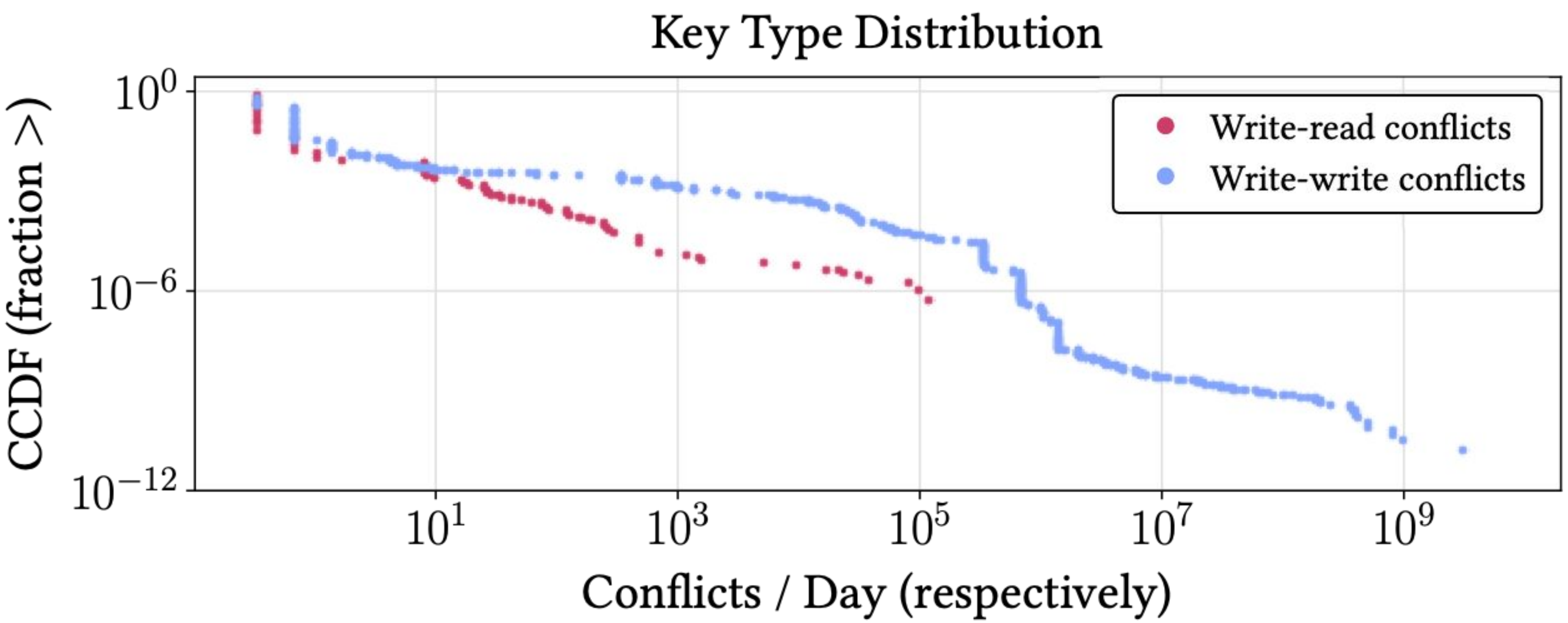
Contention

Transactional conflicts **varies greatly** for different application use cases

- >97.3% of write-write contention due to intentionally racing writes

Application use case:

- Pre-generate edges for live video time slices
- Redundant creates to ensure timely processing



Parametrizing the Workload

Identify **set of parameters** sufficient to reliably reproduce workloads

- 1. Generalizable to other data stores
- 2. Unique to TAO

Parameter	**Description**
Transaction sizes	Discrete distr. for read- & write-only txns
Sharding	Discrete distr. for objects & associations
Op. types	Proportions for single- & mutli-key reqs.
Request sizes	Discrete distr. of data sizes
Association types	Proportions of association types
Preconditions	Proportions of precondition categories
Read tiers	Proportions of reqs. served by each tier

TAOBench

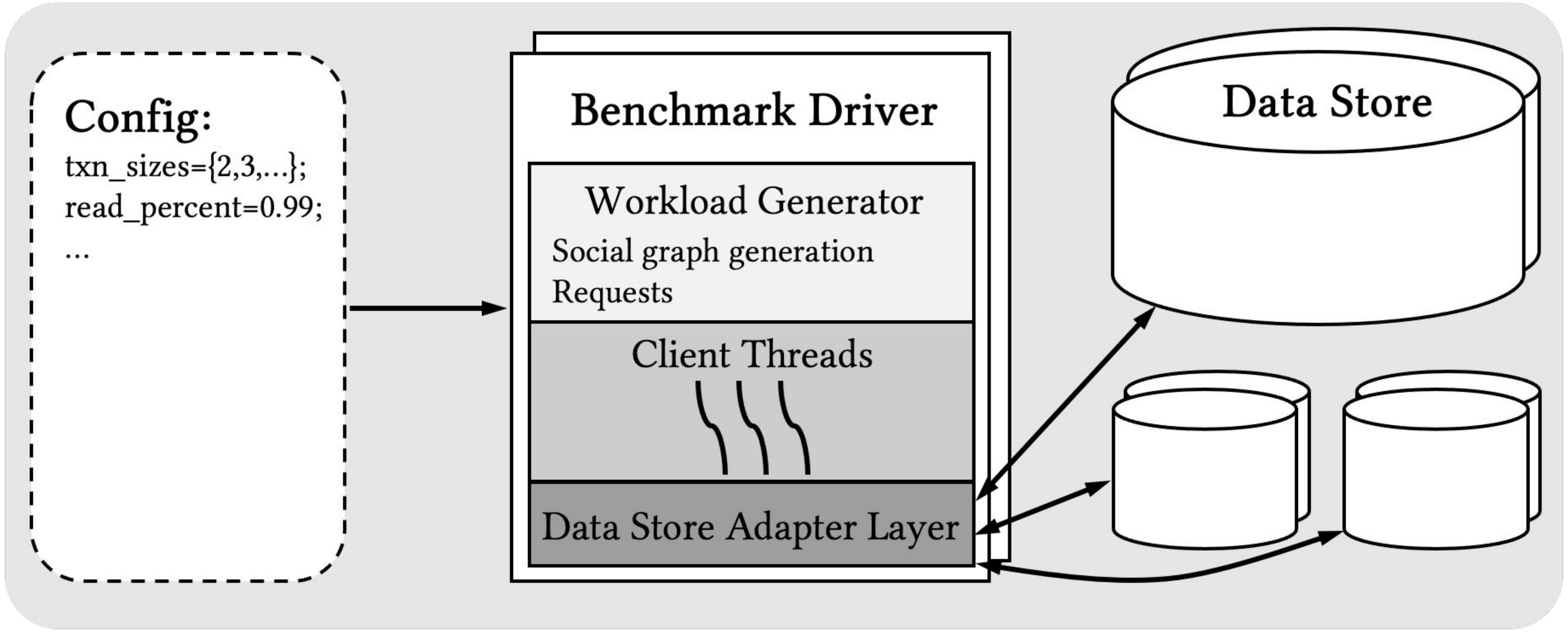
02

A new benchmark for social networks

Benchmark Architecture

Scalable, distributed drivers that are easily extensible to other systems

- Benchmark parameters: duration, target throughput, warm-up
- Workload parameters: configuration file with probability distributions



Benchmark API

Simple API based on TAO's:

- `read(key)`
- `read_txn(keys)`
- `write(key,[preconditions])`
- `write_txn(key,[preconditions])`

Easy to map to a range of databases

- Support for MySQL and PostgreSQL
- Adapters for Cloud Spanner, CockroachDB, PlanetScale, TiDB, and YugabyteDB

Benchmark Workloads

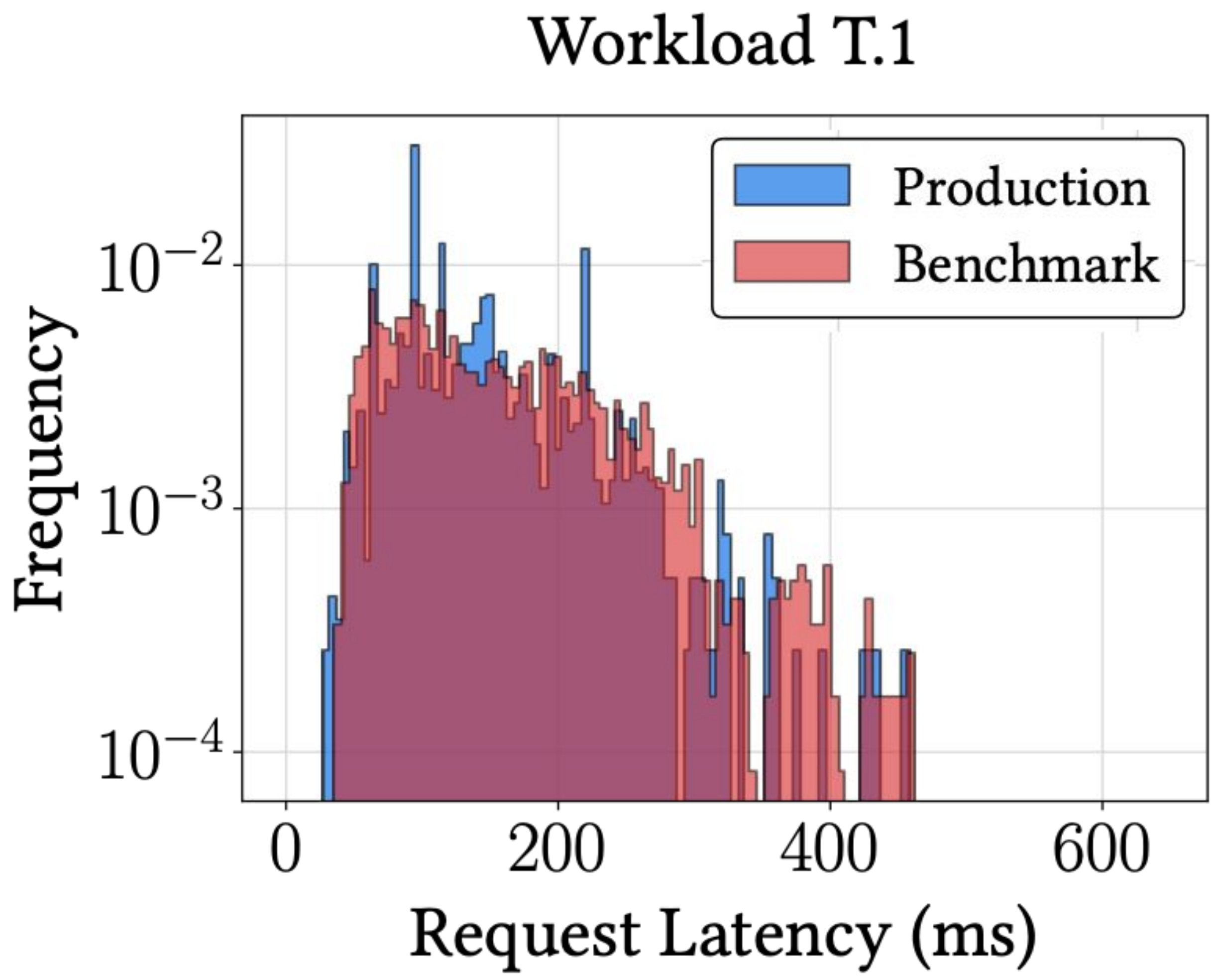
Open source **3 workloads** based on production data

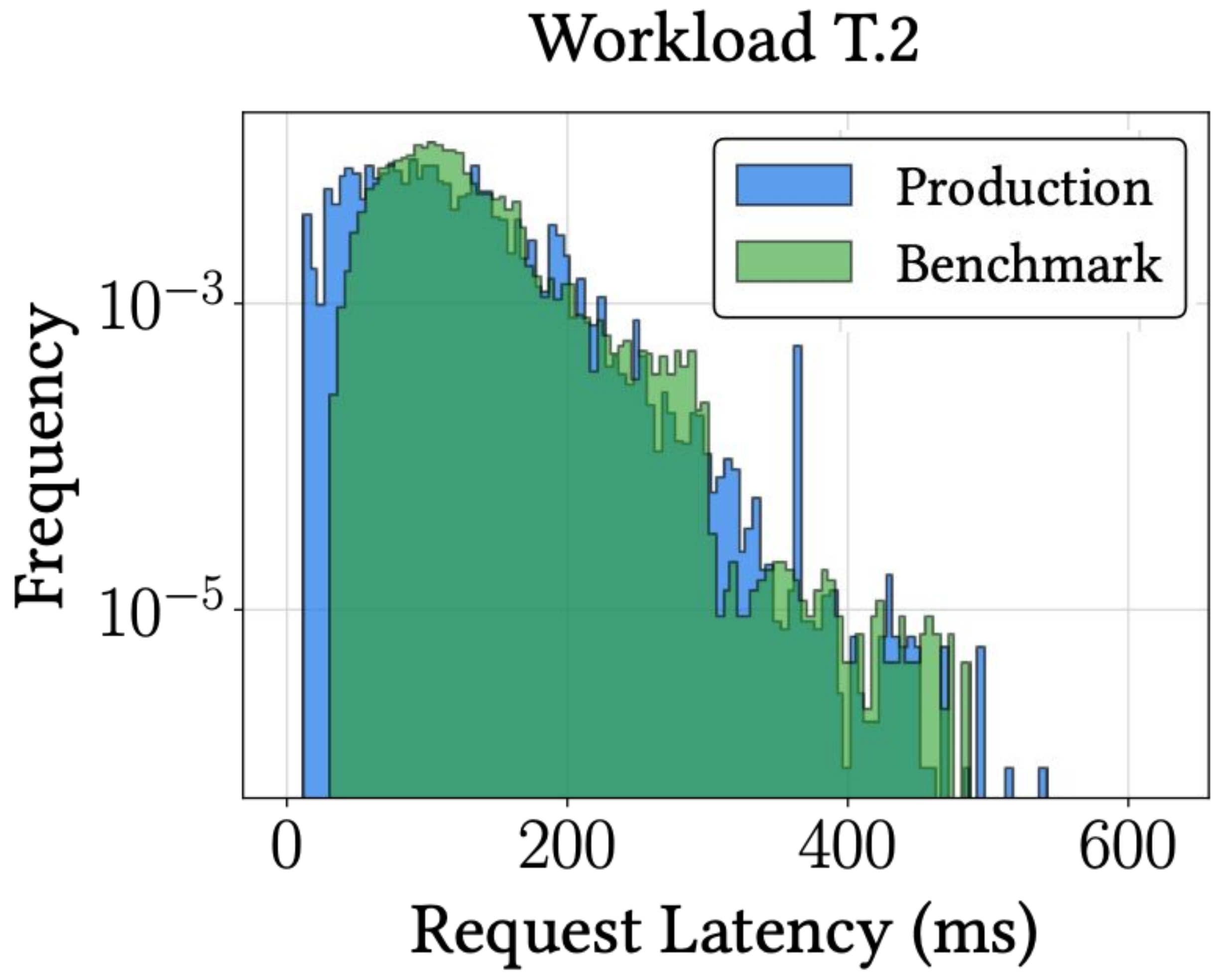
Workload	Description
T — Transaction	Current transactional workload
A — Application	Speculative transactional workload
O — Overall	Comprehensive TAO workload

Validating Benchmark

Compare **latency distribution and contention profiles**

- Statistically identical latency distributions
- Contention errors also match





Comparing Databases

03

How can TAOBench be used on other systems?

Distributed Databases

Cloud Spanner

Google's geo-distributed SQL database

- Custom SQL
- Paxos for replication
- TrueTime for strict serializability

CockroachDB

Commercial, open-source database

- Compatible with PostgreSQL
- Raft for replication
- MVCC for serializability

planetscale

Sharded MySQL database (Vitess)

- MySQL semisync replication
- Read-committed isolation across shards

TiDB

HTAP, open-source database (PingCAP)

- Compatible with MySQL
- Raft for replication
- Optimistic / pessimistic locking for SI

yugabyteDB

Cloud-native, open-source database

- Compatible with PostgreSQL
- Raft for replication
- MVCC for SI and serializability

Evaluation

For cluster configurations, core parity if possible, cost parity otherwise:

- Allocate 48 cores for hosted, cloud clusters in a single region
- 6-node cluster for Spanner

Received **extensive tuning assistance** from all companies except Spanner

Results

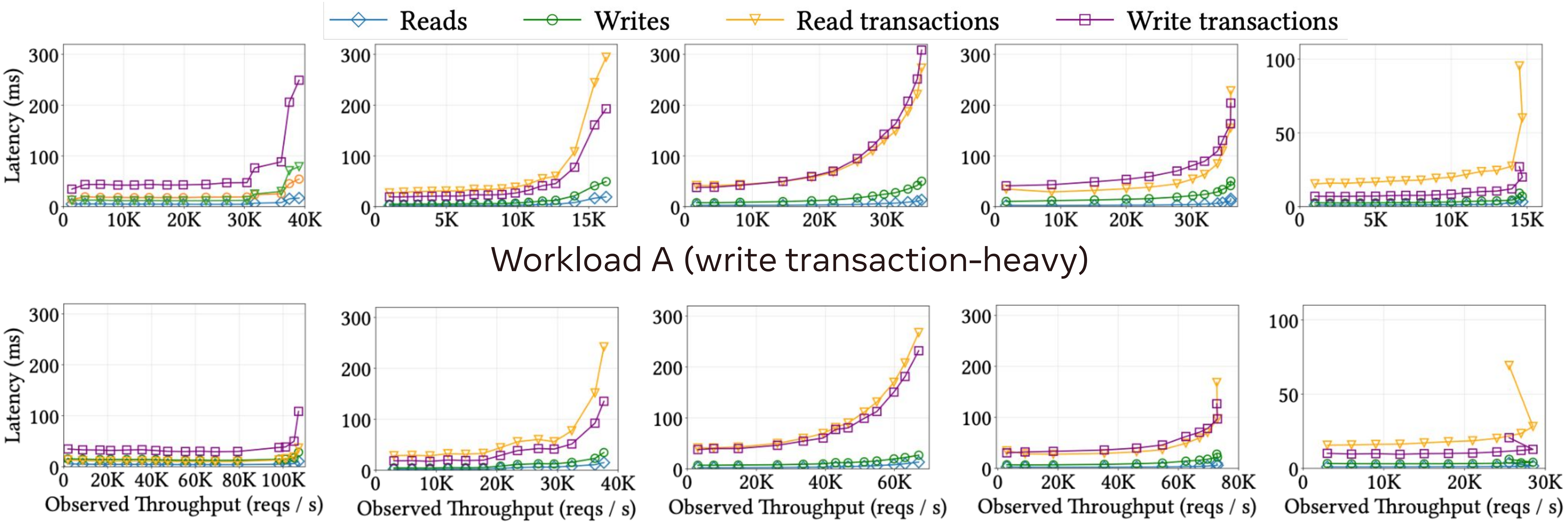
Cloud Spanner

CockroachDB

planetscale

TiDB

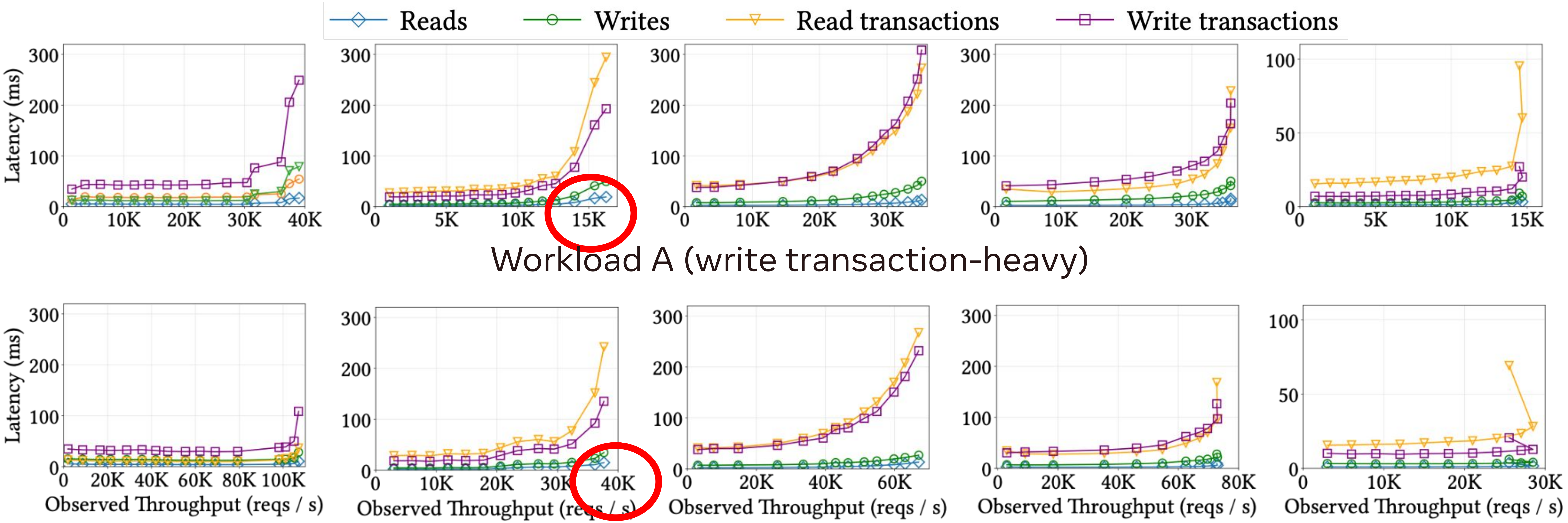
yugabyteDB



Workload O (read-heavy)

Results

Cloud Spanner CockroachDB planetscale TiDB yugabyteDB

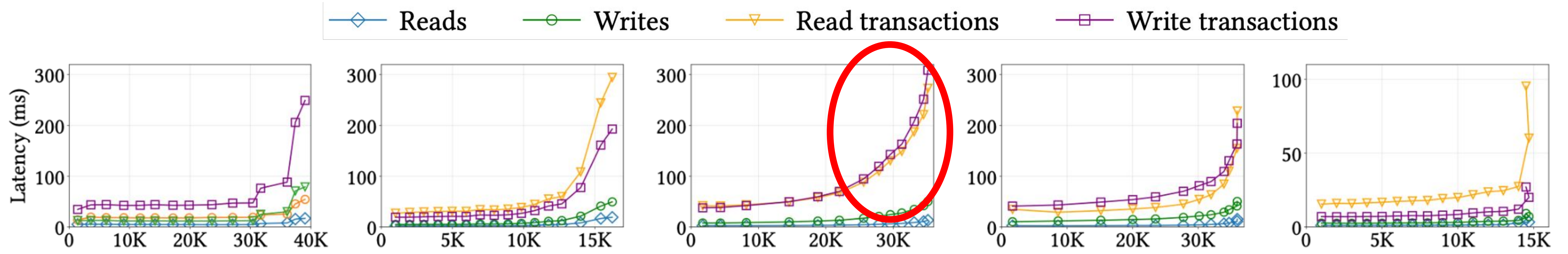


Workload O (read-heavy)

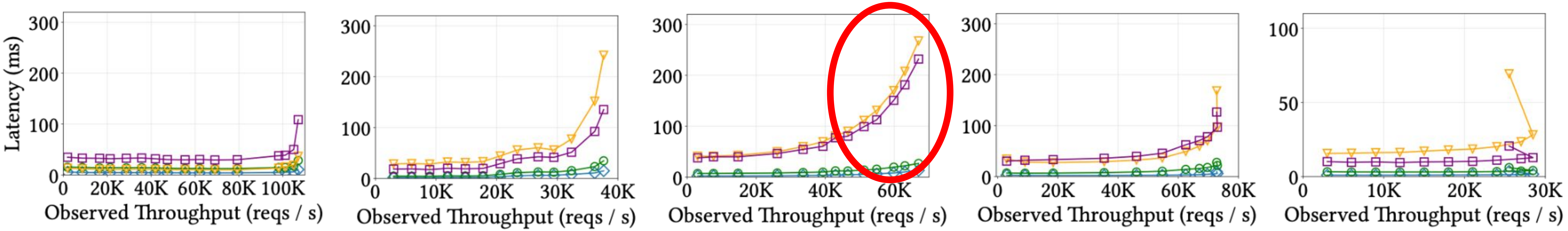
Higher performance on Workload O due to more reads

Results

Cloud Spanner | CockroachDB | planetscale | TiDB | yugabyteDB



Workload A (write transaction-heavy)

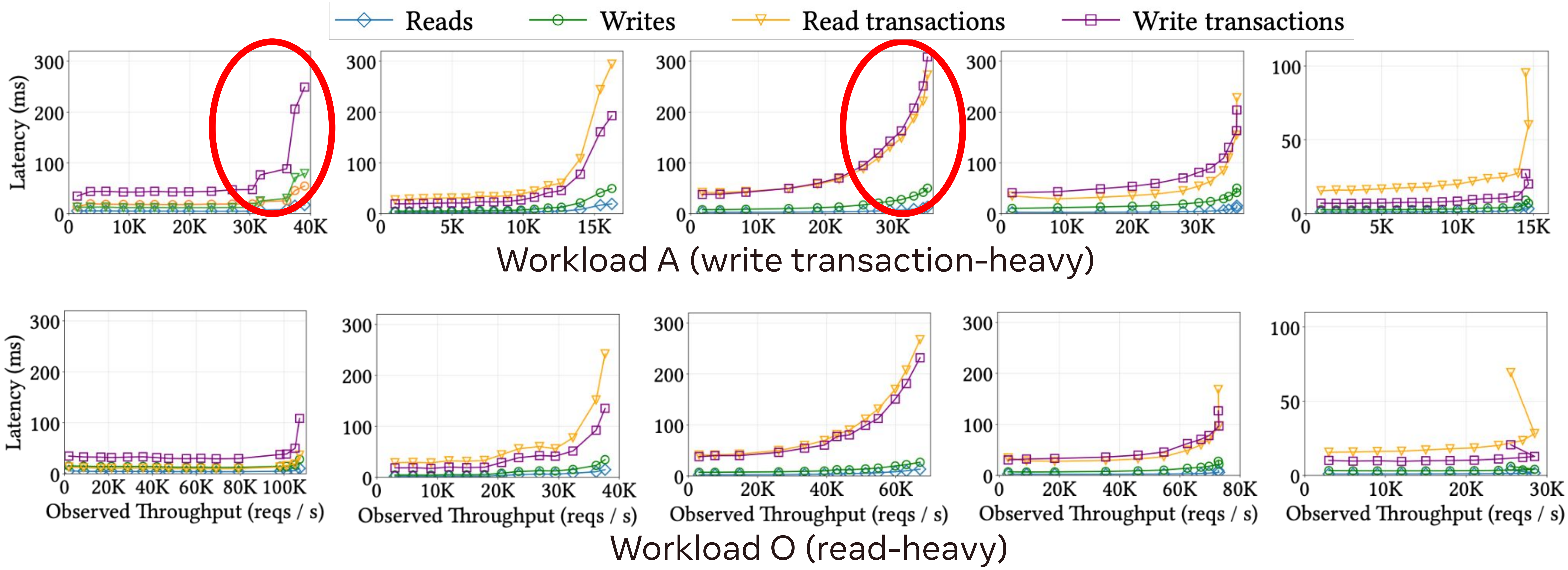


Workload O (read-heavy)

Elucidate performance differences on the same system

Results

Cloud Spanner | CockroachDB | planetscale | TiDB | yugabyteDB



Performance degradation varies across the systems

System Impact

YugabyteDB:

- Performance on TAOBench was unexpectedly slow
- Engineers found bottleneck using our benchmark
  - Postgres monitoring extension using exclusive locks
- Identified optimization for scans
  - OOM errors on TAOBench lead to discovery that filters for scans not pushed down to Postgres

Conclusion

A new benchmark for social networks: **TAOBench**

1. Derived from production traces
2. Captures any transactional requirements
3. Expresses colocation constraints
4. Models request distributions without prescriptive query types
5. Captures multi-tenant behavior over shared data

How can TAOBench be useful to LDBC?

accheng@berkeley.edu