



The LDBC Financial Benchmark

19th TUC @ London

Shipeng Qi

(with contributions from members of the FinBench Task Force)

About me

- Former Chief Product & Development Manager @ Startup
- LDBC FinBench Leader, LDBC Board Director
- Technical Expert in TuGraph team @ Ant Group
- TuGraph R&D and Open Source Develop Relationship Advocate



Motivation

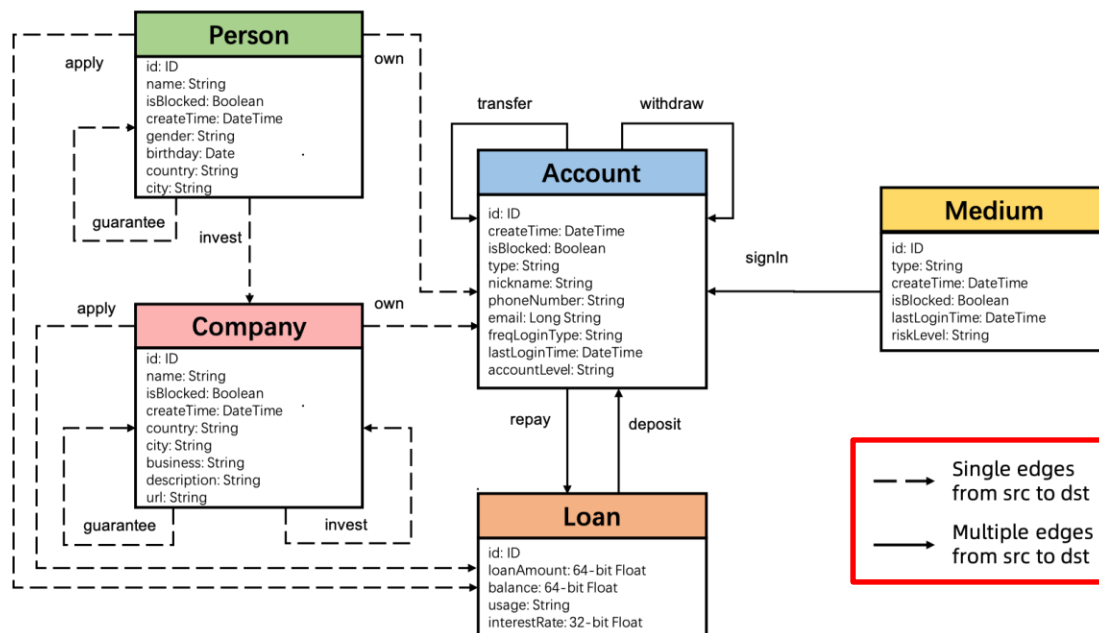
Graph databases play a **pivotal role** in the FinTech industry. However, existing ones **fail to capture the unique characteristics** of financial datasets and workloads, making them **inadequate** for evaluating graph databases in financial scenarios.

LDBC Financial Benchmark (FinBench) is a novel benchmark that adopts a **choke point-driven design methodology**, emphasizing performance bottlenecks, and incorporates distinct features.

New Features in FinBench

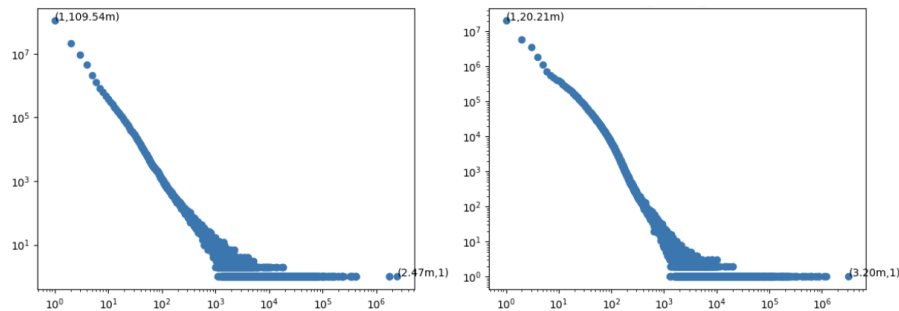
LDBC benchmarks	FinBench Transaction	SNB Interactive v1 ¹	SNB BI	Graphalytics
labelled property graph	⊗	⊗	⊗	○
temporal property graph	⊗	⊗	⊗	○
edge multiplicity	⊗	○	○	○
insert operations	⊗	⊗	⊗	○
delete operations	⊗	○	⊗	○
query footprint	small	small	large	all data
inter-query parallelism	required	required	optional	not applicable
path finding	⊗	○	⊗	⊗
time-window queries	⊗	○	○	○
recursive path filtering	⊗	○	○	○
truncated traversal	⊗	○	○	○
read-write queries	⊗	○	○	○
workload type	OLTP	OLTP	OLAP	graph algorithms
query mix	⊗	⊗	⊗	not applicable
time-biased query mix	⊗	○	○	not applicable

Dataset Schema



- Vertices are entities in financial systems, while edges are activities involving them
- Asymmetric dynamic temporal graph

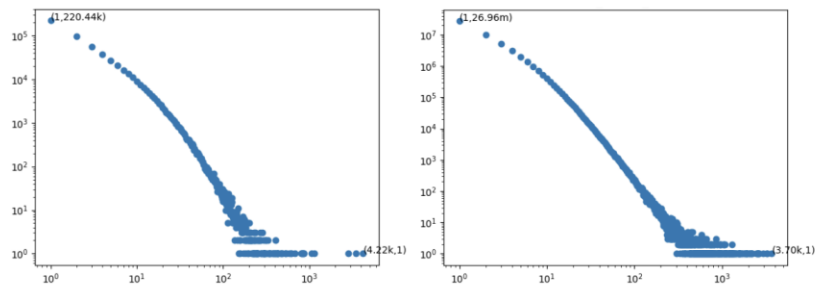
Dataset Distribution



(a) Subgraph 1, in-degree

(b) Subgraph 1, out-degree

Skewness in degree distribution following PowerLaw



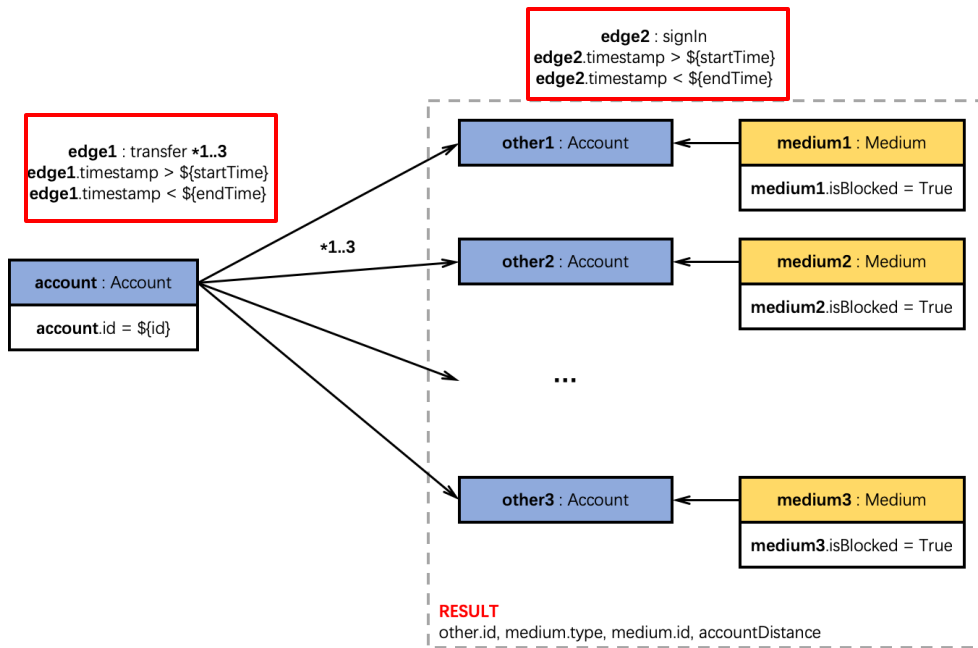
(a) In-edge multiplicity

(b) Out-edge multiplicity

Skewness in multiplicity distribution following PowerLaw

Query: Time Window Filtering

- Fact: queries only look back in a limited time window
- Filtering: filter edges between **startTime** and **endTime** in traversal
- **Checkpoint**: temporal access locality and performance



Blocked medium related accounts
[Ref: Transaction Complex Read 1]

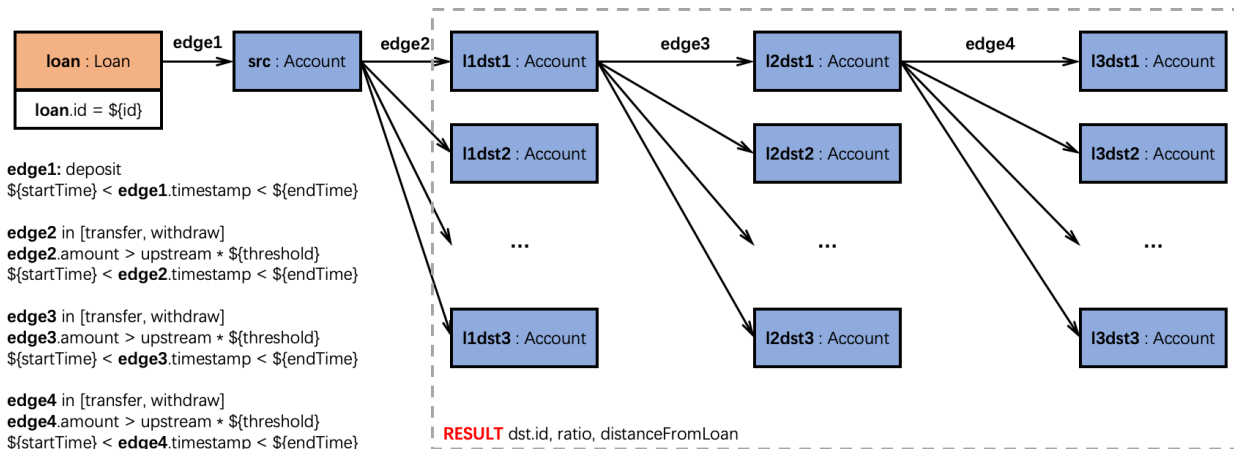
Query: Recursive Path Filtering

Assuming: $A -[e1] \rightarrow B -[e2] \rightarrow \dots \rightarrow X$

- Timestamp order: $e1 < \dots < ei$
- Amount order: $e1 > \dots > ei$

Chokepoints:

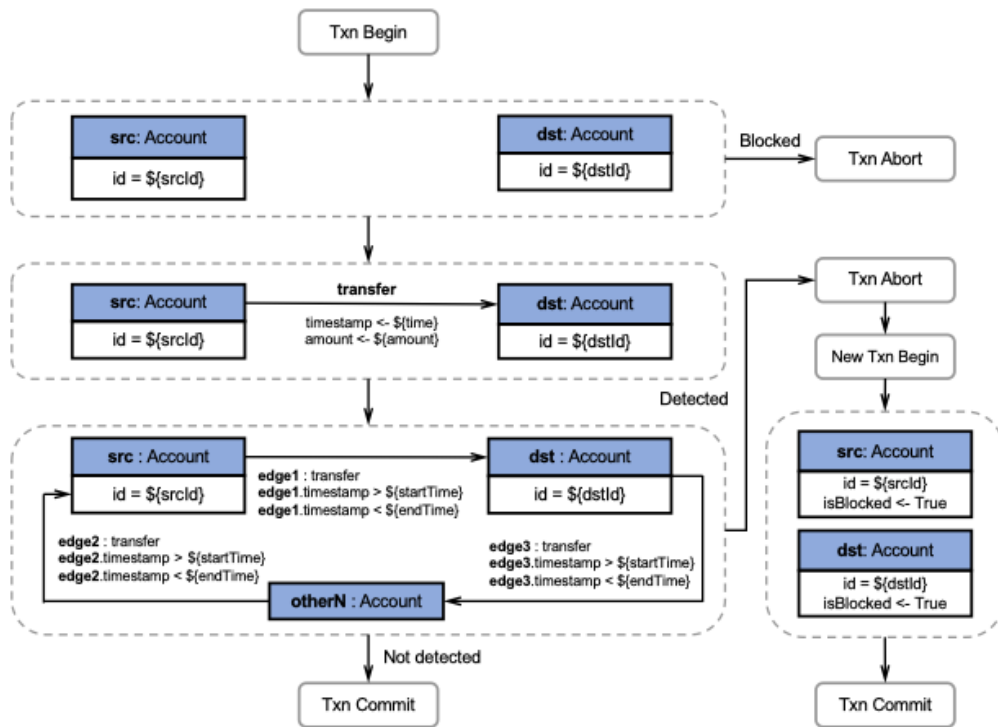
- filter push-down in RPQs for monotonic orders
- language expression



Transfer trace after loan applied
[Ref: Transaction Complex Read 8]

Query: Read-Write Query

- Complex read query: it represents a risk control strategy
- RW query: Transaction-wrapped complex reads (risk control strategy)
- If the risk control strategy is not hit, transaction commits with write query. Otherwise, transaction aborts
- **Chokepoint:** write operation contention and conflicts



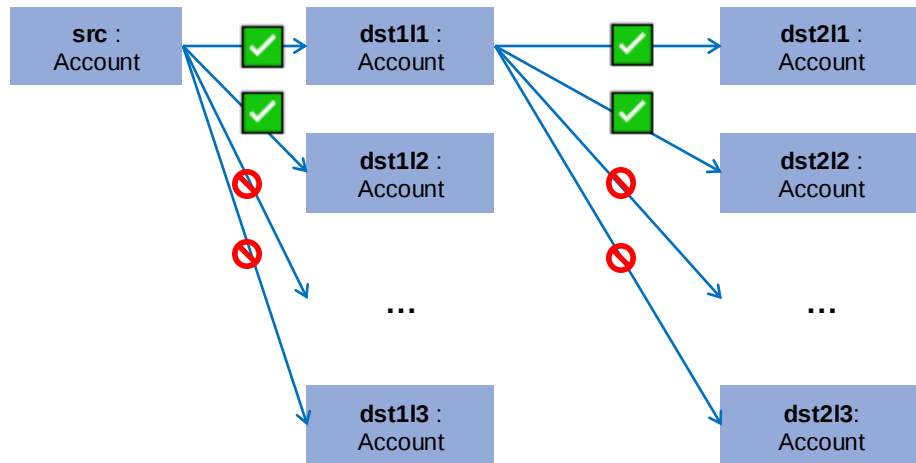
Transfer under transfer cycle detection strategy
[Ref: Transaction Read Write 1]

Query: Truncation

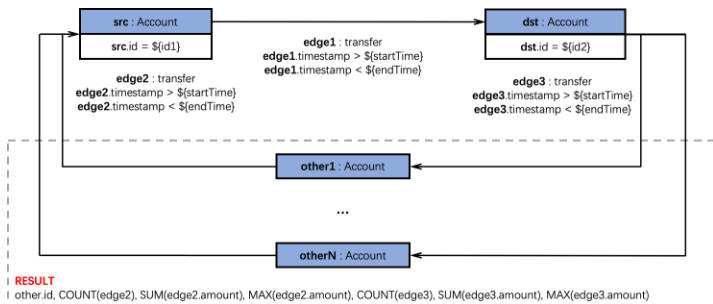
- In practice, system optimization cannot keep up with the increase of the workload complexity
- Truncate less-important edges to avoid complexity explosion, which is actually sampling.
- TruncationLimit and TruncationOrder is defined to ensure consistency of results.
- For example, keep only the top 100 edges in order of timestamp descending

Chokepoints:

- truncation push-down
- language expression

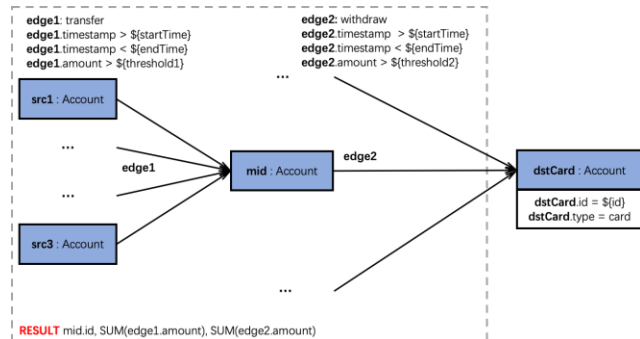


Query: Patterns in temporal graph



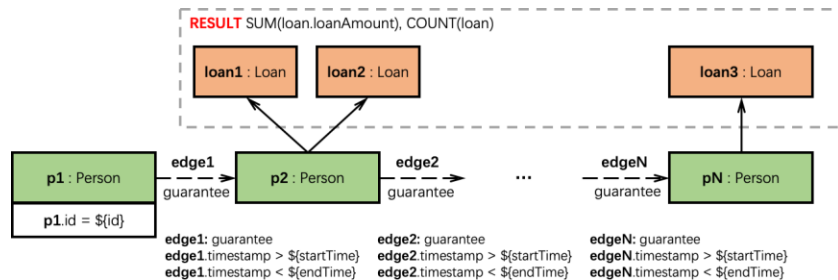
Cycle

[Ref: Transaction Complex Read 4]



Tree

[Ref: Transaction Complex Read 6]



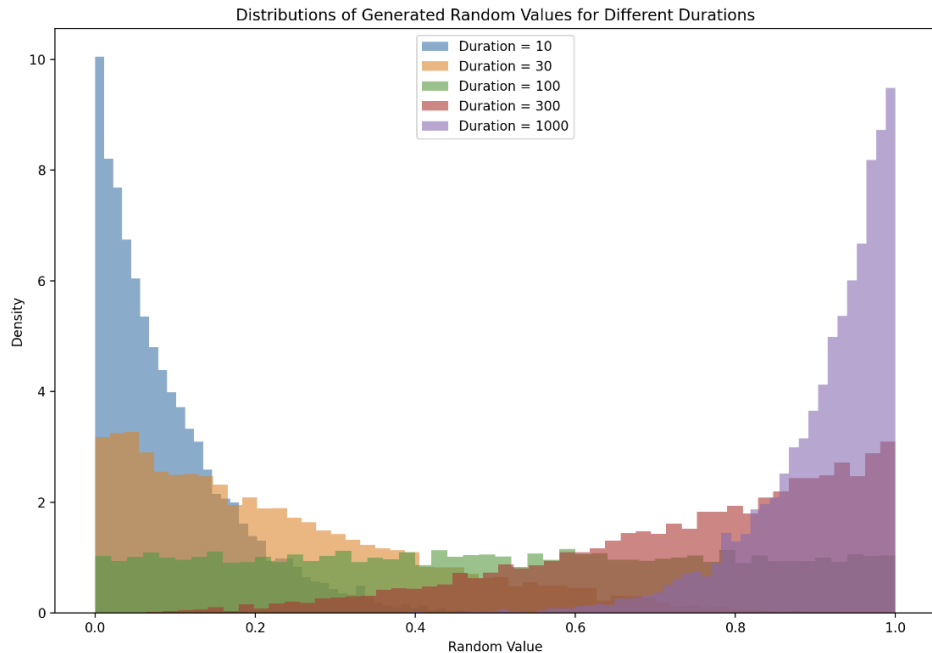
chain

[Ref: Transaction Complex Read 11]

Workload: Time-biased query mix

$$Prob(x) = 1 - U^{k \cdot t}$$

Longer temporal window in complex read queries results in more simple read queries following



The probability density distribution of the time-biased function

Future work

- Paper submitted. Working on the new version v0.2 release
- Ongoing Implementation: TigerGraph, gpStore(gStore team), Pyrrho
- Official audits planned in 2025
- New features planned: auto-benchmarking, htap metrics (freshness), new workload for AP



Thanks!



Acknowledgement

Developers: Shipeng Qi, Bing Tong, Jiatao Hu, Bin Yang, Changyuan Wang

Collaboration: Tao Lv@CSTC, Prof. Lei Zou@PKU, Prof. Malcome Crowe,
Prof. Qiang Yin@SJTU

Welcome collaboration on benchmark and
research on chokepoints

Contact me at **shipeng.qi AT ldbcouncil.org**



WeChat QR Code



*The graph & RDF
benchmark reference*