The LDBC SNB Implementation in TuGraph

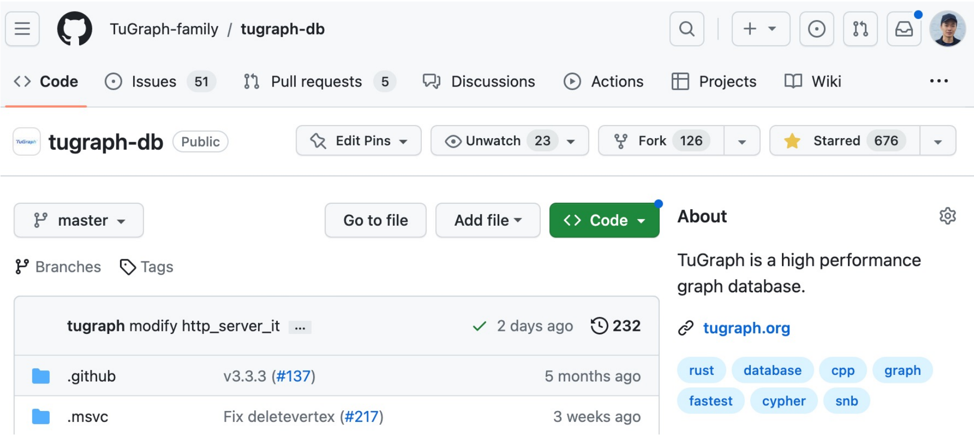
HENG LIN
Ant Group

6/23/2023

TuGraph Overview

TuGraph-DB

- Performance oriented graph database on single machine
- Full ACID support with serializable transactions
- Integrated with query / analytics / learning
- OpenCypher API (IOS GQL 2023.9)
- Stored procedure with C++/Python/Rust API
- Open source from 2022.9



TuGraph-family / tugraph-db

tugraph-db Public

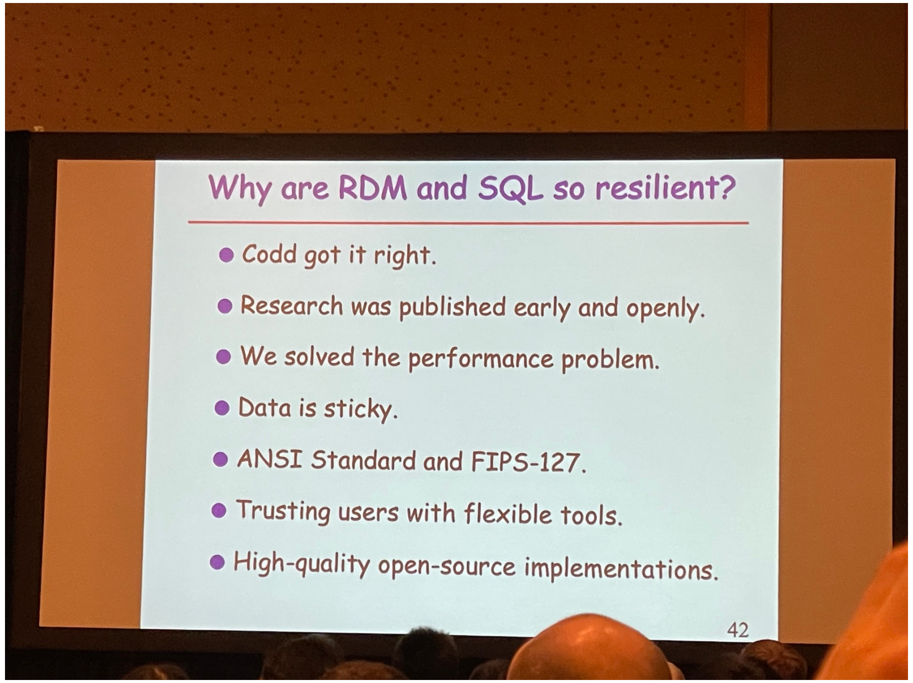
tugraph modify http_server_it ...	✓ 2 days ago	232
.github	v3.3.3 (#137)	5 months ago
.msvc	Fix deletevertex (#217)	3 weeks ago

About

TuGraph is a high performance graph database.

tugraph.org

rust database cpp graph fastest cypher snb



Why are RDM and SQL so resilient?

- Codd got it right.
- Research was published early and openly.
- We solved the performance problem.
- Data is sticky.
- ANSI Standard and FIPS-127.
- Trusting users with flexible tools.
- High-quality open-source implementations.

42

-- Don Chamberlin, 49 Years of Queries, SIGMOD2023

Design

Query Impl.

- Materialized views
- Intra-query parallelism

Transactions

- Lock free read txn
- Multi-version B+ tree

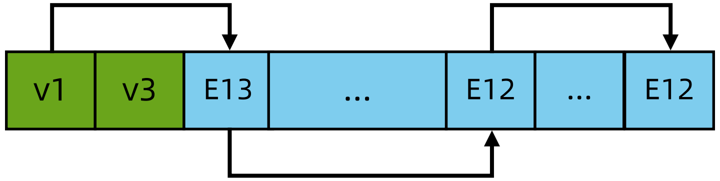
Topology

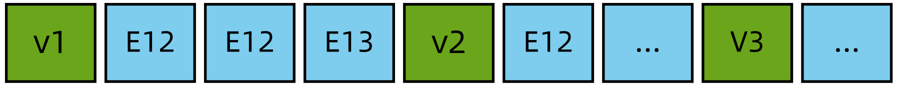
- Adaptive continuous packing
- Prop. value based sorting

Properties

- Compact layout
- Compress with fixed and variable length data

Adaptive Continuous Packing (1)





Chained (Neo4j, etc.)
Using pointer to connect vertices and edges
-> Fast write and a lot of random access

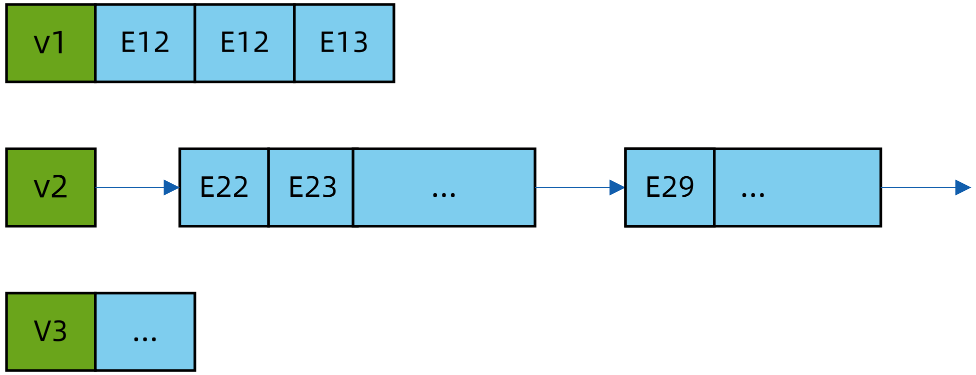
Discrete KV (Nebula, Geabase, etc.)
Each vertex and edge is a KV
-> Relative fast write, but sacrifice scan

Our observations:

1. More read than write (a.k.a 10:1 in LDBC SNB)
2. Access locality of certain vertex’s edges
3. Power-law distribution

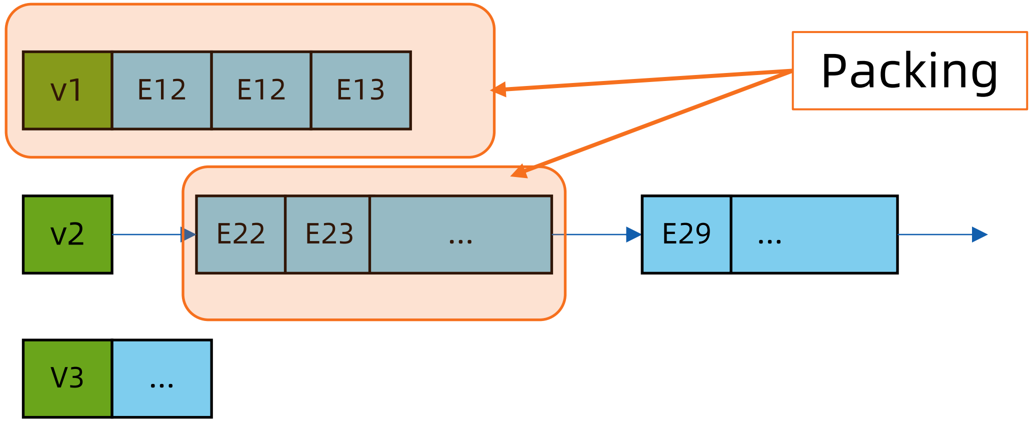
Adaptive Continuous Packing (2)

Observations	Techniques	Result
1. R/W = 10:1	Repack data while writing	A bit slow write but faster read
2. Access locality	Continuous pack edges in a block	Faster scan
3. Power-law	One block for small vertex, which more for hub	Avoid slow down for hub vertex



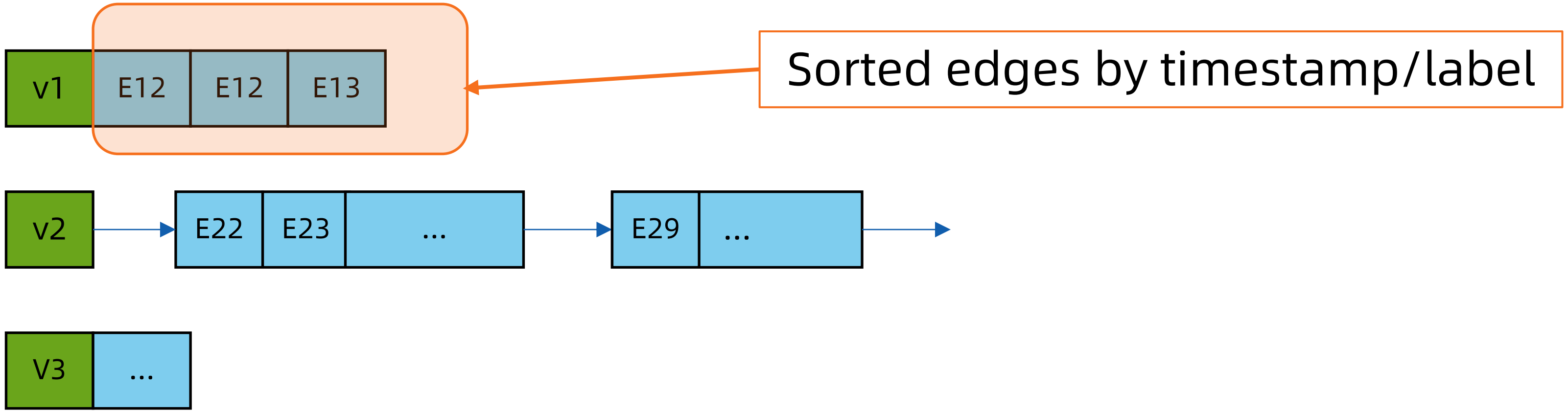
Adaptive Continuous Packing (3)

Observations	Techniques	Result
1. R/W = 10:1	Repack data while writing	A bit slow write but faster read
2. Access locality	Continuous pack edges in a block	Faster scan
3. Power-law	One block for small vertex, which more for hub	Avoid slow down for hub vertex



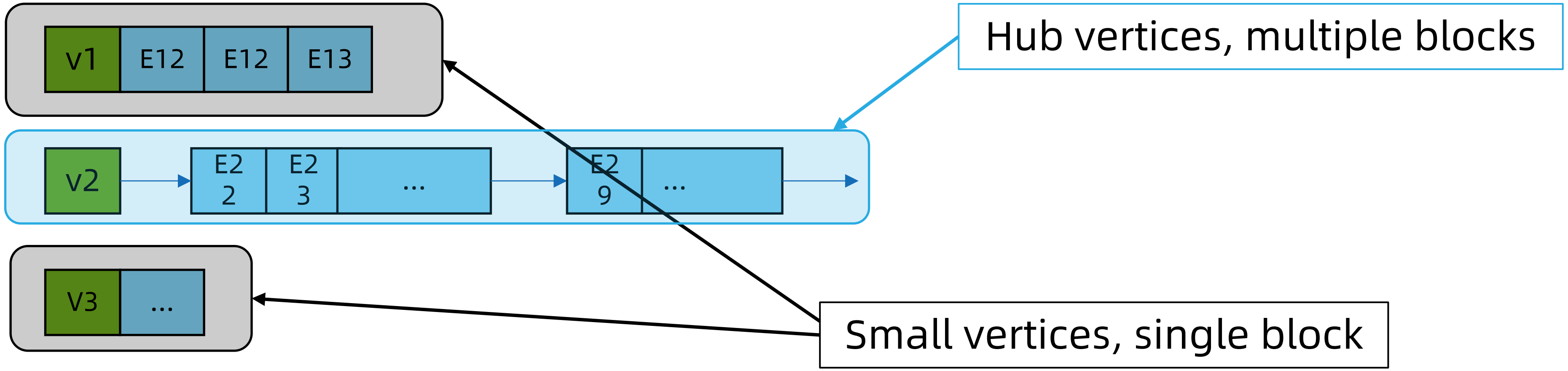
Adaptive Continuous Packing (4)

Observations	Techniques	Result
1. R/W = 10:1	Repack data while writing	A bit slow write but faster read
2. Access locality	Continuous pack edges in a block	Faster scan
3. Power-law	One block for small vertex, which more for hub	Avoid slow down for hub vertex



Adaptive Continuous Packing (5)

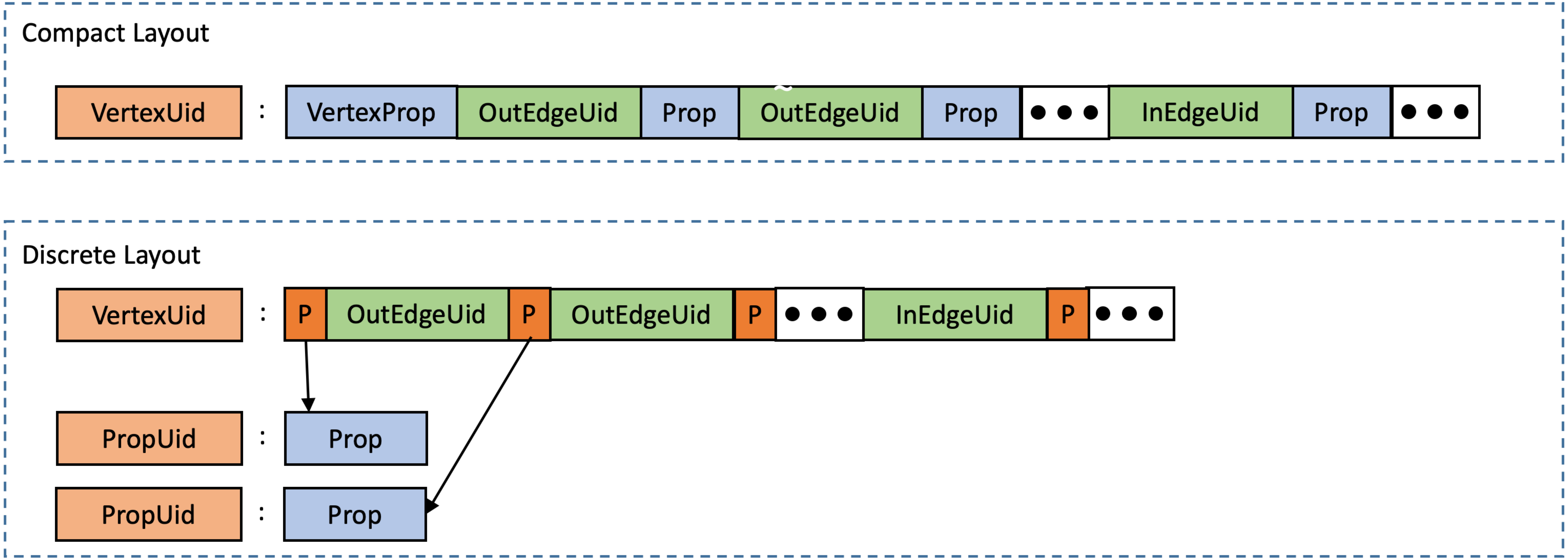
Observations	Techniques	Result
1. R/W = 10:1	Repack data while writing	A bit slow write but faster read
2. Access locality	Continuous pack edges in a block	Faster scan
3. Power-law	One block for small vertex, which more for hub	Avoid slow down for hub vertex



Hub vertices, multiple blocks

Small vertices, single block

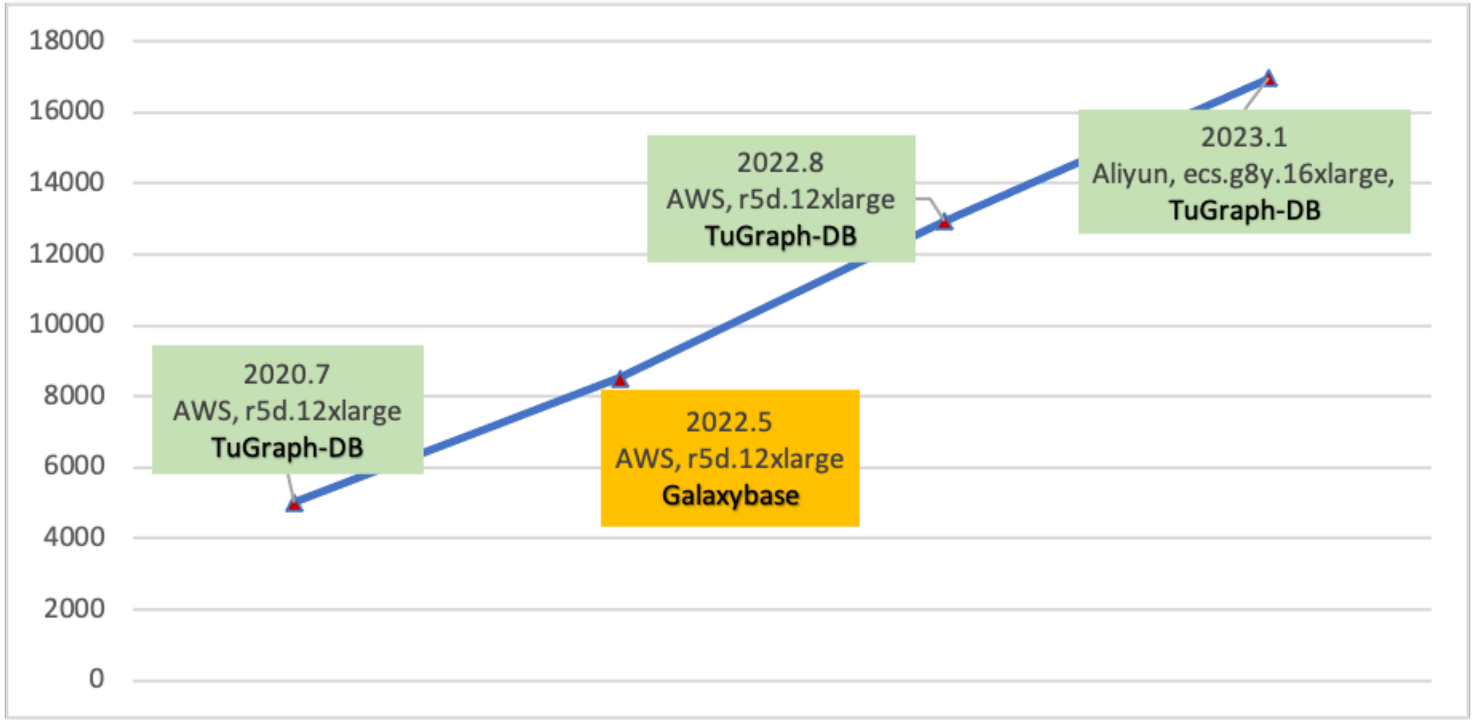
Compact Layout



- During traversal, one or more properties of vertices or edges are accessed.
- --> Choose Compact Layout

LDBC SNB Interactive Audit

SNB SF100 QPS， Higher is better

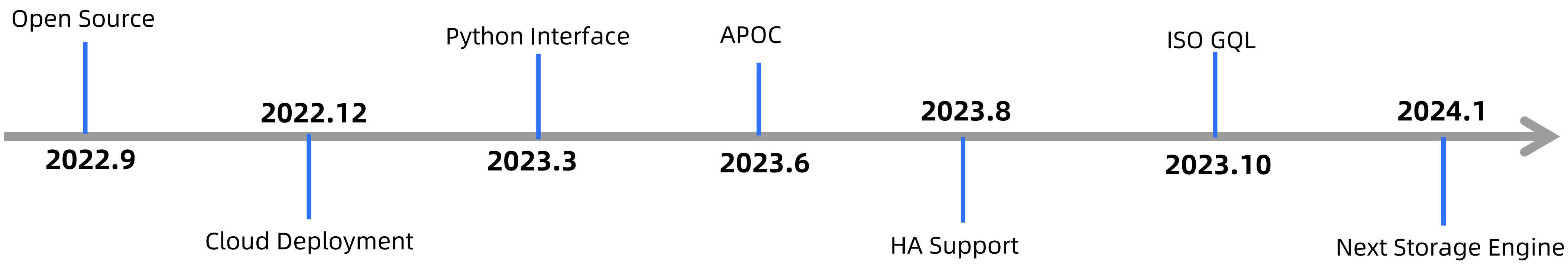


LDBC SNB Interactive

- Formulated by LDBC
- Simulate social network senario
- Workload including 29 R/W queries
- Transaction/Correctness/Throughout

TuGraph ranks No.1 in the overall throughput.

Roadmap



2022.9 Open Source

2022.12 Cloud Deployment

2023.3 Python Interface

2023.6 APOC

2023.8 HA Support

2023.10 ISO GQL

2024.1 Next Storage Engine

TuGraph-DB Repo: https://github.com/TuGraph-family/tugraph-db

TuGraph-Analytics Repo: https://github.com/TuGraph-family/tugraph-analytics

Free Trail: https://tugraph-db.readthedocs.io/en/latest/5.developer-manual/1.installation/1.cloud-deployment.html

Thank You