LDBC Financial Benchmark Introduction

Heng LIN @ Ant Group

June 16 2022
Catalog

1 / Background

2 / FinBench Design

3 / FinBench Progress
1  /  Background
LDBC Benchmarks

- **SPB**  An RDF-based benchmark for semantic databases
- **Graphalytics**  Graph algorithms benchmark
- **SNB**  - Benchmark based on social network scenarios
  - Limited when applying to financial field.
- **FinBench**  Benchmark based on financial scenarios
Differences From SNB

**Application Scenarios**
- SNB: friend recommendation, ...
- FinBench: risk control, AML, KYC, ...

**Schema Characteristics**
- SNB:
  - Single edge
- FinBench:
  - Multiple edges
  - More properties in edge

**Workload Characteristics**
- Tight latency bounds (e.g. P999 20ms)
- Read-write query
- Temporal window: constrained by start_time
- Different subgraph patterns (e.g. cycles, paths, chains)
- Recursive path filtering: Recursive path filtering and regular path query
- ...

FinBench Design
FinBench Data Schema (Proposed)

**Vertices:**
Person, Company, Account, Loan, Medium

**Edges:**
transfer, withdraw, repay, deposit, signIn, apply, workIn, invest, guarantee, own

Note: The dashed arrows represent multiple edges from source vertex to the destination
FinBench Workload (Proposed)

**Workloads**

**TP Workload**
- Latency: Tight bound
- Query: 1-4 hops

**AP Workload**
- Latency: unlimited
- Query: Iterative graph analytics

**Load Definition**
- Strong transactional requirement
- Peaks and troughs
- ...
FinBench Query Example: Read-Write Query

**Read Query**
- `account: Account`  
  - `account.id: $\{id\}`
  
  **RESULT properties(a)**

**Exact Account Property Query**
[Ref: read / 22]

**Write Query**
- `transfer`  
  - `id <- account.id; timestamp <- currentTime; amount <- amt`

**Add a transfer edge**
[Ref: write / 3]

**AML: Three accounts in a cycle**
[Ref: read / 4]

**Mark an account as blocked**
[Ref: write / 10]
AML Case:

```
TXN Begin
account: Account
  account.id: $[id]
RESULT properties(a)

TXN Abort
```

```
Account
  id <- account.id
  timestamp <- current_time
  amount <- sum
```

```
TXN Begin
  srcAccount: Account
  srcAccount.id <- $[id]
  srcAccount.timestamp <- $[start_time]
  srcAccount.amount <- $[start_amount]

edge1: transfer
  srcAccount.id <- $[id]
  srcAccount.timestamp <- $[end_time]
  srcAccount.amount <- $[end_amount]

TXN Abort
```

```
Account
  id <- account.id
  transfer
```

```
TXN Begin
  dstAccount: Account
  dstAccount.id <- $[id]
  dstAccount.timestamp <- $[start_time]
  dstAccount.amount <- $[start_amount]

edge2: transfer
  dstAccount.id <- $[id]
  dstAccount.timestamp <- $[end_time]
  dstAccount.amount <- $[end_amount]

TXN Commit
```

```
AML Case:

Account
  id <- account.id
  isBlocked <- True
```

```
TXN Commit
```

```
TXN Abort
```

```
TXN Begin
account: Account
  account.id: $[id]
RESULT properties(a)
```

```
Account
  id <- account.id
  timestamp <- current_time
  amount <- sum
```

```
TXN Begin
  srcAccount: Account
  srcAccount.id <- $[id]
  srcAccount.timestamp <- $[start_time]
  srcAccount.amount <- $[start_amount]

edge1: transfer
  srcAccount.id <- $[id]
  srcAccount.timestamp <- $[end_time]
  srcAccount.amount <- $[end_amount]

TXN Abort
```

```
Account
  id <- account.id
  transfer
```

```
TXN Begin
  dstAccount: Account
  dstAccount.id <- $[id]
  dstAccount.timestamp <- $[start_time]
  dstAccount.amount <- $[start_amount]

edge2: transfer
  dstAccount.id <- $[id]
  dstAccount.timestamp <- $[end_time]
  dstAccount.amount <- $[end_amount]

TXN Commit
```

```
AML Case:

Account
  id <- account.id
  isBlocked <- True
```

```
TXN Commit
```

```
TXN Abort
```

```
TXN Begin
account: Account
  account.id: $[id]
RESULT properties(a)
```

```
Account
  id <- account.id
  timestamp <- current_time
  amount <- sum
```

```
TXN Begin
  srcAccount: Account
  srcAccount.id <- $[id]
  srcAccount.timestamp <- $[start_time]
  srcAccount.amount <- $[start_amount]

edge1: transfer
  srcAccount.id <- $[id]
  srcAccount.timestamp <- $[end_time]
  srcAccount.amount <- $[end_amount]

TXN Abort
```

```
Account
  id <- account.id
  transfer
```

```
TXN Begin
  dstAccount: Account
  dstAccount.id <- $[id]
  dstAccount.timestamp <- $[start_time]
  dstAccount.amount <- $[start_amount]

edge2: transfer
  dstAccount.id <- $[id]
  dstAccount.timestamp <- $[end_time]
  dstAccount.amount <- $[end_amount]

TXN Commit
```

```
AML Case:

Account
  id <- account.id
  isBlocked <- True
```
Financial Scenarios

With the passage of time:
- Queries only look back in a temporal window

FinBench Design

Queries are constrained by:
- After start_time

(In implementation, systems can choose to deprecate old data.)

 Blocked medium related accounts
(query constrained by start_time and end_time)
[Ref: read / 1]
FinBench Query Example: Subgraph Pattern

Subgraph pattern: Transfer cycle
[Ref: read / 4]

Subgraph pattern: Guarantee cycle
[Ref: read / 18]

Subgraph pattern: Guarantee chains
[Ref: read / 19]

Subgraph pattern: Transfer paths
[Ref: read / 9]
FinBench Query Example: Recursive Path Filtering

**Regular Path Queries**
- Edges in path: deposit, transfer, withdraw -> multi-type edges
- Flexible Expression: RPQs

**Recursively Path Filtering**
Assuming: A -[e_1]- B -[e_2]- ... - X
- Timestamp order: e_1 < e_2 < ... < e_i
- Amount order: e_1 > e_2 > ... > e_i
- Time window: e_{i-1} < e_i < e_{i-1} + \triangle

Transfer trace after loan applied (Path Filtering)
[Ref: read / 11]
3 / FinBench Progress
Work Charter Established

Name of task force: FinBench Task Force
Proposed or current leader: Zhihui Guo, Ant Group, guozhihui.gzh@antgroup.com

Scope of work:

- The FinBench project aims to define a graph database evaluating benchmark and develop a data generation process and a query driver to make the evaluation of the graph database representative, reliable and comparable, especially in financial scenarios.

Members:
FinBench Draft version 0.1.0-RC

Schema

Online / read / 23

<table>
<thead>
<tr>
<th>query</th>
<th>Online / read / 23</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>Accounts with the same transfer sources of exact account</td>
</tr>
<tr>
<td>pattern</td>
<td></td>
</tr>
<tr>
<td>desc</td>
<td>Given an Account, find all the blocked accounts that connect to a third-party account which the given account has transfer-in from. Return all the accounts’ id.</td>
</tr>
<tr>
<td>params</td>
<td>id</td>
</tr>
<tr>
<td></td>
<td>start_time</td>
</tr>
<tr>
<td></td>
<td>end_time</td>
</tr>
<tr>
<td>result</td>
<td>COLLECT(DISTINCT daAccount.id)</td>
</tr>
<tr>
<td>sort</td>
<td>todo</td>
</tr>
<tr>
<td>limit</td>
<td>todo</td>
</tr>
<tr>
<td>CPUs</td>
<td>0.0</td>
</tr>
<tr>
<td>relevance</td>
<td>This query is a typical analysis for gang related accounts in risk control.</td>
</tr>
</tbody>
</table>

* https://ldbcouncil.org/ldbc_finbench_docs/ldbc-finbench-specification.pdf

Read Query *24
Write Query *14
Read-Write Query * 5
FinBench Progress

Timeline

- 2022.02
  - FinBench Proposal
  - Members invitation
  - Policy procedures

- 2022.03 - 2022.04

- 2022.05
  - Official Announce
  - Kickoff

- 2022.06

- 2022.12
  - Version 0.1.0-rc
  - Official Announce
  - Kickoff
Timeline

**Version v0.1.0-rc** DSN&DEV: from Jun to Dec

- **v0.1.0 approved by LDBC BoD based on v0.1.0-rc**

- **Spec DSN and Suite DEV works in parallel**

- **Online meeting for every 2-3 week at design stage**

**Note:**
- DSG short for design
- DEV short for development
Welcome to Join us!

Useful Links:


THE END

THANK YOU!