The Linked Data Benchmark Council (LDBC): Driving competition and collaboration in the graph data management space

Gábor Szárnyas1*, Brad Bebee2, Altan Birler3, Alin Deutsch4,5, George Fletcher6, Henry A. Gabb7, Denise Gosnell2, Alastair Green8, Zhihui Guo9, Keith W. Hare5, Jan Hidders10, Alexandru Iosup11, Atanas Kiryakov12, Tomas Kovatchev12, Xinsheng Li13, Leonid Libkin14, Heng Lin9, Xiaojian Luo15, Arnau Prat-Pérez16, David Püroja1, Shipeng Qi9, Oskar van Rest17, Benjamin A. Steer18, Dávid Szakállas19, Bing Tong20, Jack Waudby21, Mingxi Wu5, Bin Yang13, Wenjuan Yu15, Chen Zhang20, Jason Zhang13, Yan Zhou20, and Peter Boncz1

1 CWI, the Netherlands, 2 Amazon Web Services, 3 Technische Universität München, Germany, 4 UC San Diego, 5 TigerGraph, 6 TU Eindhoven, 7 Intel Corporation, 8 JCC Consulting, 9 Ant Group, 10 Birkbeck, University of London, 11 VU Amsterdam, the Netherlands, 12 Ontotext AD, 13 Ultipa, 14 University of Edinburgh; RelationalAI; ENS, PSL University, 15 Alibaba Damo Academy, 16 work done while at UPC Barcelona and Sparsity, 17 Oracle, USA, 18 Pometry Ltd., 19 individual contributor, 20 CreateLink, 21 Newcastle University, School of Computing

* Corresponding author, gabor.szarnyas@ldbcouncil.org

Abstract. Graph data management is instrumental for several use cases such as recommendation, root cause analysis, financial fraud detection, and enterprise knowledge representation. Efficiently supporting these use cases yields a number of unique requirements, including the need for a concise query language and graph-aware query optimization techniques. The goal of the Linked Data Benchmark Council (LDBC) is to design a set of standard benchmarks that capture representative categories of graph data management problems, making the performance of systems comparable and facilitating competition among vendors. LDBC also conducts research on graph schemas and graph query languages. This paper introduces the LDBC organization and its work over the last decade.
LDBC Graphalytics
The Graphalytics data sets consist of untyped, unattributed graphs, which are either directed or undirected and optionally have edge weights.
## Largest graphs

| graph               | $|V|$   | $|E|$  |
|---------------------|-------|-------|
| datagen-9_3-zf      | 555M  | 1.3B  |
| datagen-sf10k-fb    | 100M  | 18.8B |
| graph500-30         | 450M  | 34.0B |
Algorithms
BFS

Breadth-first search (source: “Bob”)

Assign the level of traversal for each vertex starting from the source (level = 0).
The PageRank variant in Graphalytics redistributes the PageRank values from sinks among all vertices to avoid “leaking” the PageRank out of the network.
This is the only algorithm that uses edge weights. Many implementations use the delta-stepping SSSP algorithm. These are allowed to specify the delta value for each graph.
Weakly connected components

WCC

Ada

Bob

Carl

Dan

Finn

Eve

Gia

A

A

A

D

D

WCC
For each vertex, LCC is $\#\text{triangles} / \#\text{wedges}$.

This algorithm is very similar to triangle count.

\[
LCC(v) = \begin{cases} 
0, & \text{if } |N(v)| \leq 1 \\
\frac{|\{(u,w) | u,w \in N(v) \wedge (u,w) \in E\}|}{|\{(u,w) | u,w \in N(v)\}|}, & \text{otherwise}
\end{cases}
\]
Community detection using LP (iterations: 2)

\[ L_i(v) = \min \left( \arg \max_l \left[ \left| \{ u \in N_{in}(v) \mid L_{i-1}(u) = l \} \right| \right. \right. \]
\[ \left. + \left| \{ u \in N_{out}(v) \mid L_{i-1}(u) = l \} \right| \right] \left. \right) \]

In each iteration, the next label of a vertex is selected as the minimum mode value among the labels of the neighbours.
Graphalytics algorithms

All 6 algorithms:

- have **directed and undirected variants**
- are **deterministic**

Validation uses different matching strategies:

- **Exact match** (BFS, CDLP)
- **Epsilon match** – relative tolerance of 0.01% (LCC, PR, SSSP)
- **Equivalence match** – same equivalence classes (WCC)
Competition site is now open

https://graphalytics.ldbcouncil.org/
LDBC Social Network Benchmark
<table>
<thead>
<tr>
<th>Data set</th>
<th>Queries</th>
<th>Updates</th>
</tr>
</thead>
</table>

Data set

- Ada
- Bob
- Carl
- Dan
- Finn
- Gia
- Eve

Queries

- M1 Mon
- M2 Tue
- M3 Sun
- M4 Tue
- M5 Fri

Updates

- author
- reply

knows

- Ada knows Bob
- Finn knows M1 Mon
- Finn author M1 Mon
- Eve reply M2 Tue
- M3 Sun
- M4 Tue
- M5 Fri
Q9(“Bob”, “Sat”)

Data set:
- Bob
- Ada
- Finn
- Carl
- Eve
- Dan
- Gia

Queries:
- M1: Mon
- M2: Tue
- M3: Sun
- M4: Tue
- M5: Fri

Updates:
- Pa knows Pb
  - knows *1..2
  - name = “Bob”
- M
  - creation date < “Sat”
null
Q9("Bob", "Sat")

Pa knows *1..2

name = "Bob"

author

creation date < "Sat"
### Data set
- Ada
- Bob
- Carl
- Eve
- Dan
- Finn
- Gia

### Queries
- **Q9**(“Bob”, “Sat”)
- Pa knows *1..2
- name = “Bob”
- creation date < “Sat”

### Updates
- ✔
- ✔
- ✔

### Queries
- M1: Mon
- M2: Tue
- M3: Sun
- M4: Tue
- M5: Fri

### Updates
- Pa knows *1..2
- name = “Bob”
- creation date < “Sat”
Data set:
- Ada
- Bob
- Carl
- Dan
- Gia

Queries:
- Finn
- M1 (Mon)
- M2 (Tue)
- M3 (Sun)
- M4 (Tue)
- M5 (Fri)

Updates:
- Pa knows Pb
  - *1..2
- name = "Finn"
- creation date < "Wed"

Queries:
- Q9("Finn", "Wed")
Data set

- Ada
- Finn
- Bob
- Carl
- Eve
- Dan
- Gia

Queries

- M1 (author = Finn, Mon, reply)
- M2 (author = M1, Tue)
- M3 (author = M2, Sun)
- M4 (author = M3, Tue)
- M5 (author = M4, Fri)

Updates

- Q9 ("Finn", "Wed")
- Pa knows Pb
- *1..2
- name = "Finn"
- creation date < "Wed"
Q9("Finn", "Wed")

Pa knows *1..2
name = "Finn"
creation date < "Wed"
Ada
Bob
Dan
knows
Carl
Eve
Finn
Gia

Factors table

<table>
<thead>
<tr>
<th>name</th>
<th>1-hop friends</th>
<th>2-hop friends</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Carl</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Ada</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Dan</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Finn</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Gia</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Factor table

<table>
<thead>
<tr>
<th>name</th>
<th>1-hop friends</th>
<th>2-hop friends</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Carl</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Ada</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>Dan</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>Finn</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Q9("Bob", "Sat"): 10 nodes
Q9("Finn", "Wed"): 5 nodes
Data set

[Diagram of a network with nodes: Ada, Bob, Carl, Eve, Dan, Gia, Finn. Edges include knows relationships between nodes and authorship links.]

Queries

Updates

+ knows("Eve", "Gia")
When is this operation executable?
Data set

Queries

Updates

Ada

Bob

Carl

Eve

Dan

Gia

Finn

creation date: 09:00
dependent date: --:--

M1

M2

M3

M4

M5

M6

author

reply

creation date: 10:00
dependent date: 09:00

creation date: 14:30
dependent date: 10:00

creation date: 14:31
dependent date: 14:30

knows
Updates

- Person("Eve")

+ knows("Eve", "Gia")

+ Comment("Gia", "M3")

Queries

Data set

knows

Bob

Ada

Finn

M1 Mon

author

M2 Tue

reply

M3 Sun

M4 Tue

M5 Fri

M6 Sun

Eve

Carl

Gia

Dan
Cascading deletes remove lots of entities:
- have a big impact on the data distribution
- affect the executability of operations
- influence parameter selection

For databases, deletes:
- prohibit append-only data structures
- stress the garbage collector
Data set

Queries

Benchmark driver

System under test

Updates
- Schedules operations to be executable
  (hard: needs careful parameter selection and dependency tracking)

- Runs queries and updates concurrently
  (hard: needs partitioned updates)

- Collects benchmark results and performs validation
  (very hard due to concurrent updates: we perform it sequentially)
SNB Workloads
SNB Interactive v1 (2015)

Queries start in 1–2 person nodes
Concurrent inserts (no deletes)
Goal: high throughput (ops/s)

Results on the 100GB data set

SF 100 throughput

- name = $name
- creation date < $day

<table>
<thead>
<tr>
<th>Year</th>
<th>Concurrent Inserts (no deletes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020</td>
<td>4k</td>
</tr>
<tr>
<td>2021</td>
<td>8k</td>
</tr>
<tr>
<td>2022</td>
<td>16k</td>
</tr>
<tr>
<td>2023</td>
<td></td>
</tr>
<tr>
<td>2024</td>
<td></td>
</tr>
</tbody>
</table>
SNB Business Intelligence (2023)

Audited results

Queries touch on large portions of the data
Both bulk and concurrent updates allowed
Goal: high throughput & low query runtimes

10TB:
- Power@SF: 89,444
- Throughput@SF: 30,990

More results expected in 2023
Features backported from BI:

- delete operations
- larger scale factors up to SF30,000
- cheapest path query

New parameter generation features:

- temporal bucketing for each day
- path curation

Queries start in 1–2 person nodes

Concurrent inserts and deletes

Goal: high throughput (ops/s)
The LDBC Social Network Benchmark Interactive Workload v2: A Transactional Graph Query Benchmark with Deep Delete Operations

David Pùroja\(^1\), Jack Waudby\(^2\), Peter Boncz\(^1\), and Gábor Szárnyas\(^1\)

\(^1\) CWI, the Netherlands, \(^2\) Newcastle University, School of Computing
david.puroja@ldbcouncil.org, j.waudby2@newcastle.ac.uk, boncz@cwi.nl,
gabor.szarnyas@ldbcouncil.org

Abstract. The LDBC Social Network Benchmark’s Interactive workload captures an OLTP scenario operating on a correlated social network graph. It consists of complex graph queries executed concurrently with a stream of updates operation. Since its initial release in 2015, the Interactive workload has become the de facto industry standard for benchmarking transactional graph data management systems. As graph systems have matured and the community’s understanding of graph processing features has evolved, we initiated the renewal of this benchmark. This paper describes the Interactive v2 workload with several new features: delete operations, a cheapest path-finding query, support for larger data sets, and a novel temporal parameter curation algorithm that ensures stable runtimes for path queries.
Path curation
Shortest distance from “Ada” to “Eve”
Shortest distance from “Ada” to “Eve”
Shortest distance from “Ada” to “Eve”
Shortest distance from “Ada” to “Eve”
Shortest distance from “Ada” to “Eve”

The shortest path distance changes multiple times during the day.
Path curation with temporal bucketing

For each day, we construct:

**G1** – deletes but no inserts, setting an *upper* bound

**G2** – inserts but no deletes, setting a *lower* bound

*lower* $\leq$ actual length $\leq$ *upper*

Pairs of nodes yielding 3-hop paths in G1 and G2:

- 1 to 5
- 1 to 6
- 2 to 5
- 2 to 6
Is path curation sufficient?

Not yet:

- We also have to consider the degree distribution of the source–target nodes.
Is path curation sufficient?

Not yet:

- We also have to consider the degree distribution of the source–target nodes.

Actually:

- For “perfect” parameter curation, we would need to run the entire workload with many parameter candidates and only keep ones which showed a similar behaviour.
Is path curation sufficient?

Not yet:

- We also have to consider the degree distribution of the source–target nodes.

Actually:

- For “perfect” parameter curation, we would need to run the entire workload with many parameter candidates and only keep ones which showed a similar behaviour.

The real question:

- Is it worth spending more effort on optimizing the parameter curation?
I’m leaving academia

- Moving to DuckDB Labs (CWI spin-off in Amsterdam)
- Staying involved with LDBC at ~1 day / month
Future benchmark ideas

- Financial Benchmark
- Graphalytics
- SNB Interactive
- SNB Business Intelligence
- Semantic Publishing Benchmark
- Financial Benchmark
- Future benchmark ideas

**Domain:**
- Media/publishing industry
- Financial fraud detection

**Target:**
- RDF/SPARQL
- Distributed systems
- RDF/SPARQL
- Distributed systems
- RDF/SPARQL
- Distributed systems
- RDF/SPARQL
- Distributed systems

**Inferencing & continuous updates**

**Strict latency bound (20 ms)**

**Algorithms**
- BFS
- CDLP
- PR
- SSSP
- LCC
- WCC

**Data sets**
- LDBC SNB
- Graph500
- Twitter
- Friendster
- Patents
- wiki-Talk

**Q9**($name$, $day$)

name = $name$
n creation date < $day$

**Q11**($country$)

name = $country$

**Graphalytics**

**Target:**
- Distributed systems
- Distributed systems
- Distributed systems

**Domain:**
- Media/publishing industry
- Financial fraud detection

**Strict latency bound (20 ms)**

**GNNs**

**Graph mining**

**Graph streaming**