Retrospective review of publications related to LDBC benchmark standards: Keynote at NODES 2021 (Neo4j Online Developer Expo and Summit)

Summary

document	• keynote: https://www.youtube.com/watch?v=4ZCs83_iHU8&t=2874s • blogpost: https://medium.com/neo4j/behind-the-scenes-of-creating-the-worlds-biggest-graph-database-cd22f477c843
date	June 2021
workload	SNB Interactive variant
scale factor(s)/data set size	100TB
code availability	https://github.com/neo4j/trillion-graph
data generation	custom
parameter generation	custom
benchmark driver	custom
systems-under-test	Neo4j Enterprise (including the Fabric feature), Neo4j Driver v4.2.3

1. What does it purport to be? What does the publication state as its goals and results, and does it state any qualifications or limitations to those goals and results?

The *publication by Neo4j* consists of a keynote presentation at the NODES 2021 conference, a follow-up blog post, and a GitHub repository with the implementation used for the demo presented in the keynote. The claim made by Neo4j is that the Enterprise version of the software with Neo4j Fabric (a distributed GDBMS) is capable of running some deep, complex graph global queries of the LDBC standard benchmark touching over 1000 shards. Additionally, these queries return in less than 100ms and exhibit constant read performance as the graph grows two orders of magnitude up to one trillion edges and slightly more than 208 billion nodes, which is an instance of the LDBC SNB benchmark metamodel. The blog post of the publication report also *discusses the necessary cloud infrastructure setup and its operating costs*.

2. How is it being used, quoted or described in published material available on the web? This should be conducted to a reasonable degree with a focus on statements made by LDBC members.

There are only a few notable references found on the internet since the appearance of the publication. None of the works from the references below are scientific publications. Below is the list of direct references:

- Neo4j website: https://neo4j.com/nodes-2021/ and https://neo4j.com/press-releases/neo4j-scales-trillion-plus-relationship-graph/
- Twitter: https://twitter.com/emileifrem/status/1405528420402925575
- GraphStuff.fm: https://poddtoppen.se/podcast/1557747094/graphstufffm-the-neo4j-graph-database-developer-podcast/compliance-services-in-gaming-the-path-to-nodes-2021-with-arthur-namias-de-crasto

3. How closely does it adhere to the LDBC benchmark standards, in terms of completeness, fidelity, reproducibility etc.? Which changes were made compared to the official benchmark specification (data sets, queries, query parameters, driver/workload)?

The most current LDBC Benchmark Specification of LDBC SNB is v0.3.3 which was released on January 27, 2021 and it includes an auditing guidelines section (link to this section in the specification shared on ArXiv; page 81). The NODES 2021 keynote by Emil Eifrem (Neo4j) took place on June 17, 2021, which means that version 0.3.3 of the standard is considered as the specification in effect for the LDBC Benchmark when writing this retrospective report.

Furthermore, the report does not include a dedicated section/chapter to explicitly discuss the differences between the performed query runs and the one prescribed by the LDBC SNB specification. The accompanying GitHub repository's *About* section has "A scale demo of Neo4j Fabric spanning up to 1129 machines/shards running a 100TB (LDBC) dataset with 1.2tn nodes and relationships." This suggests that an official LDBC dataset was used, however, the only thing that is shared between the official LDBC dataset and the one used in the demo is that the edge and node labels and properties come from a similar domain.

Below, we go over the steps in the auditing guidelines and discuss the presented experiment with respect to these. In each case, the respective section number from the benchmark specification document is indicated in parentheses.

Preparation (6.1)

- *System details (6.1.1)*
  The report provides a detailed description (in the form of a README and a set of scripts available on GitHub) of the infrastructure setup that complies with the auditing

guidelines. The piece that would additionally be necessary for an audited execution is a detailed breakdown of the price of the solution including license costs.[1]

- *Benchmark environment setup (6.1.2)*
  The demo application runs selected queries continuously with fixed parameters. In an audited benchmark, the official LDBC driver is expected to be used to ensure that the query performance is measured according to the benchmark standards. For example, in the case of the Interactive workload, operation throughput is measured over the course of 2 hours continuously executing queries with different parameters. In this case, however, one selected query is evaluated repeatedly with the same parameters, thus resulting in frequent data cache hits during subsequent query runs.

  Moreover, a very serious issue with the experiment is that it does not use the official LDBC benchmark data generator. The data is generated using a custom generator that uses a trivial algorithm to insert nodes, edges, and labels into the database. By examining the code of this generator, one can conclude that the resulting data does not bear any characteristics that are otherwise provided by the official data generator. In fact, the resulting graph is very simple where connections between nodes are highly localized:
  - `KNOWS` edges are added in a way that a `Person` with a given `ID` knows other persons with adjacent IDs (code)
  - The outdegree of `KNOWS` edges is much lower (max. 10) compared to LDBC dataset (36)
  - Based on the layout of persons, the friend of a friend (foaf) count is capped at `MAX_KNOWN_PERSON * 4 = 40`, while in the LDBC dataset average foaf is around 2300
  - `Comments` for a `Post` come from `Person` nodes with adjacent IDs (code)
  - `Likes` to `Comments` come from `Person` nodes that have adjacent IDs with creators of the respective `Post` (code)
- *Data loading (6.1.3)*
  The details of data loading are provided within the GitHub repository with sufficient detail. However, the generated data is a subset of the LDBC benchmark data, which would be a blocking issue in case of an audit.

Running the benchmark (6.2)
A selected subset of the LDBC query set (LDBC SNB Interactive - Complex Read 4, 6, 7, and 9) was implemented. The implementations are faithful to the specification, however, they do incorporate the information that is referred to as “graph-native sharding” in the keynote, i.e., how the graph is sharded and data is allocated to them (1 person shard, rest of the shards are forum shards). For a successful audited run, the whole set of queries would need to be implemented and the throughput measurements should be reported as described in the auditing guidelines.

---

[1] However, we also note that in the case of the nowadays dominant cloud-based setups such as the one used in this case, disclosing the price of a dedicated instance (or instances) for 3 years will be required in the future audits.

This would mean 2 hours of simulation that also incorporates updates with a desired time compression ratio and a diverse parameter set for the queries.

Furthermore, we find that the execution of the queries is significantly different from the goals of an LDBC benchmark execution in the following two ways:

- *Single query parameter:* Each query execution uses a single parameter assignment. For this reason, we argue that these execution times do not faithfully represent the true querying capabilities of the underlying graph database, as caching mechanisms can store the results between runs and can significantly reduce the time needed to retrieve the results.
- *Queries do not scatter out:* Even though the underlying graph is scaled up to very large sizes, the query run is independent from this expansion of the data. Each query execution will only involve a selected small graph fragment for which the shards are known a priori based on the query parameters.

Recovery and Serializability (6.3, 6.4)
The recovery and serializability aspects were not discussed in the report. These would also be mandatory in case of an audited benchmark run.

4. How accurate is its reporting and analysis?
Results are reported as individual query run times on the selected set of queries. We find the reported constant query times on 10x as well as on 100x the size of the initial graph is misleading in this case, as the query only touches a small slice of the data independent from the size of the graph.

5. How might this publication be categorized if it were produced in the future, after we have adopted policies on how to describe or publish results obtained from audited, non-audited-but-complete, and LDBC derivative-or-inspired tests?

This publication would be categorized as derived.

6. Any reactions, prior to distribution to the board, from the publication's authors

The NODES 2021 demo was never called a benchmark, apart from in error, once, in a blogpost that has since been corrected. Categorizing it as sub-bronze is acceptable, however not categorizing it would probably be more correct as it is not a benchmark.