Retrospective review of publications related to LDBC benchmark standards: 2019 preprint (sponsored by TigerGraph)

Summary

document	https://arxiv.org/pdf/1907.07405.pdf
date	July 2019
workload	SNB Interactive, SNB Business Intelligence
scale factor(s)/data set size	sf1, sf10, sf100, sf1000
code availability	available on GitHub
data generation	SNB Datagen, v0.2.7
parameter generation	custom (uniform random)
benchmark driver	custom
systems-under-test	TigerGraph 2.3.1 Developer Edition, Neo4j 3.5.0 Community Edition

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 comparing Noe4j and TigerGraph performance* is a preprint published on arXiv in July 2019 and a GitHub repository with the implementation of both the Interactive and the Business Intelligence workloads of the LDBC SNB Benchmark. The claim made by this report is that this is the first complete implementation of LDBC SNB for two separate systems executed and ran over scale factors 1-1000. An additional claim is that TigerGraph outperforms Neo4j in case of most queries by one to two orders of magnitude. The 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.

List of scholarly articles referencing the publication:

- Aggregation Support for Modern Graph Analytics in TigerGraph (2020): https://dl.acm.org/doi/abs/10.1145/3318464.3386144

    - The publication is referenced as “a third-party-conducted, comprehensive evaluation of TigerGraph and Neo4j using the Linked Data Benchmark Council’s (LDBC) Social Network Benchmark (SNB)”
- Suitability of graph database technology for the analysis of spatio-temporal data (2020): https://www.mdpi.com/1999-5903/12/5/78
    - The publication is referenced as support for the following statement: “**TigerGraph** is an enterprise level graph analytics platform developed with insights from projects such as Apache TinkerPop and Neo4j and provides advanced features like native parallel graph processing and fast offline batch loading”
- An Empirical Study on Recent Graph Database Systems (2020) https://link.springer.com/chapter/10.1007/978-3-030-55130-8_29
    - The publication is used to motivate and support the choice of LDBC SNB workloads for the performance assessment of graph database systems
- Subgraph matching on property graphs in a distributed setting (2021): https://pure.tue.nl/ws/portalfiles/portal/174887596/Dijkhuizen_L..pdf
    - This is a master’s thesis and it references the publication to introduce the LDBC Benchmark. In our view, this reference could (and possibly should) point to the benchmark specification instead.
- An overview of graph databases and their applications in the biomedical domain (2021): https://academic.oup.com/database/article/doi/10.1093/database/baab026/6277712
    - This article references the publication as a supporting evidence that the “design [of TigerGraph] targets enterprise applications, where the number and heterogeneity of external sources are not a concern, but instead, the size and performance, by optimizing storage format and query execution strategy”

Blog post:

- TigerGraph blog: https://info.tigergraph.com/ldbc-benchmark
    - The publication is referred to as “first complete test of graph database vendors’ performance with intensive analytical and transactional workloads”

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)?

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 of the benchmark environment from a hardware perspective. One thing to call out is the different types of machines used for different scale factors. This can cause some confusion when presenting the results in one single table, like as it is done in Table 3.

- *Benchmark environment setup (6.1.2)*
  Benchmark settings and environment details are provided within the GitHub repository with a sufficient level of detail. The GSQL implementation of benchmark queries were found to match their specifications.
- *Data loading (6.1.3)*
  The details of data loading are provided within the GitHub repository with sufficient detail.

Running the benchmark (6.2)
The complete LDBC SNB query set is implemented and executed over scale factors 1, 10, 100, and 1000 using both TigerGraph and Neo4.
A major shortcoming here is that for the Interactive workload, the official LDBC SNB Driver was not used. Using the driver is a requirement for audited benchmark runs, as the driver ensures that the query workload is the same across different assessments. Furthermore, it executes updates to the data in addition to measuring the database performance of read queries in steady state with diverse parameter bindings.
We would like to note that the current Driver does not yet support BI queries, so the comment above is only applicable to the queries of the Interactive workload. However, the GRADES paper *An early look at the LDBC Social Network Benchmark's Business Intelligence workload* from 2018 does have a methodology section that is recommended to follow:

> "*Methodology.* We executed 100 queries for warmup, then executed 250 queries and measured their response time. Queries were selected randomly, following a uniform distribution and were executed one-by-one, i.e. with no interleave between them. For each scale factor/tool/query, we calculated the geometric mean of execution times"

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?

Query execution run times are reported in a unified table (Table 3) for all scale factors and both tools. These times are obtained by running each query 10 times, and taking the median of the last 9 runs. Dropping the first run time is due to the assumption that the first run will exhibit longer execution due to warmup effects. An issue with this table is that the results for scale factor 100 for complex reads and Business Intelligence queries are exactly the same, which we assume to be a copy-paste error during the compilation of the report.

Besides the results summary table, there is also comparison between the two tools using side-by-side barcharts. These diagrams are somewhat misleading, as they have logarithmic vertical axes. Instead of barcharts, some other visualization kind would have been much better (e.g., a scatterplot).

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, as it is in part faithfully implements the LDBC SNB benchmark, however, there are some benchmark components that are entirely replaced by custom implementations (e.g., driver, query parameters).

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

I recalled that at the time to prepare for this publication, the authors generated the data set from the LDBC driver, they could not obtain the parameters for scale factor 1000 and 100 from the official data generator or they obtain empty result from the generated parameters. This is the main reason they randomly picked some parameters that can produce non-empty result.