SFU SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

Simon Fraser University
8888 University Drive
Burnaby BC
Canada V5A 1S6

778 782 3111
778 782 6930
sfu.ca

November 31, 2023

Dear Colleagues,

Thank you for inviting me to serve as a referee of the doctoral thesis “Distributed Graph Query Engine Improvements for Big Data Graphs”. As requested, this report: (a) summarizes the contributions that can be refereed as new scientific results and their importance in the area; and, (b) comments on the thesis and assesses the author’s ability for creative scientific work. The report ends with a few questions that the author can comment on during their defense.

The submitted thesis presents improvements for graph pattern matching queries in the production-grade distributed graph processing engine PGX.D. There is a growing need to analyze graph data which has resulted in several large-scale graph processing systems in database and systems research. Graph processing engines like PGX.D (and its extension, PGX.D/Async) are complex data analytics systems that efficiently evaluate graph queries on large graph datasets by leveraging available computation resources (i.e., often distributed and multi-threaded execution). The contributions in the submitted thesis improve graph pattern matching support in PGX.D across three directions. First, a hybrid BFS-DFS pattern matching approach in PGX.D/Async that leverage parallelization and locality benefits from BFS exploration as well as the bounded memory guarantees from DFS exploration. Second, the RPQd algorithm that employs BFS-DFS exploration for reachability regular path queries and leverages PGX.D/Async’s flow control and memory management. Lastly, improvement to query planning in PGX.D using scouting queries that dynamically collect runtime statistics to identify faster query plans in a lightweight manner. The three contributions are published in peer-reviewed systems conferences and workshops.

The submitted thesis makes contributions in graph databases and graph systems research, which is an important and established research area, and is gaining practical adoption in industry. The proposed contributions focus on memory footprint control, which is a crucial aspect in big data systems since it is often ignored by existing state-of-the-art solutions, often rendering them impractical on realistic large-scale workloads. Moreover, the contributions are implemented and evaluated in an existing complex software system, which nicely bridges the scientific research with practical development in industry. Incorporating new techniques in PGX.D/Async is a commendable effort.

While the scientific contributions may be less apparent in terms of novelty compared to state-of-the-art research, developing novel solutions within the scope of an existing software system is challenging. The thesis provides a detailed background about what already existed in PGX.D/Async, which helped clarify my understanding about the subtle novel components

that are new scientific results. Since the proposed contributions are at the system-level, clearly highlighting the technical novelty (as opposed to use case novelty) can help promote the scientific contributions. More importantly, technical novelty for contributions related to reachability RPQs (Chapter 5) compared to pattern matching techniques in Chapter 4 remained unclear since the RPQd algorithm appears to leverage similar exploration strategies, and it builds on top of PGX.D/Async instead of contributions presented in Chapter 4.

A related point of confusion is the presentation of PGX.D/Async in Chapter 3. Since Chapter 3 presents PGX.D/Async which is a prior work and not part of scientific contributions in this thesis, it is unclear why most of the originally published PGX.D/Async text is repeated as a complete chapter. This confusion could be eliminated by: (a) describing only the required components of PGX.D/Async that are relevant to the contributions in this thesis and eliminating other portions that are already published in the peer-reviewed publication (e.g., performance evaluation of PGX.D/Async); and, (b) doing so in Chapter 2 (Background) instead of dedicating a separate chapter for PGX.D/Async.

Adding new features to an existing complex software system is not an easy task, especially since it requires balancing scientific research possibilities with the existing scope of software capabilities. The author has successfully presented useful improvements to PGX.D/Async, which is a commendable effort and it showcases practicality of their proposed solutions. Additionally, the author has demonstrated their ability to paint a wide picture of related works in the area of graph pattern systems; it was good to see the breadth of the topics covered in Chapter 2. The author has shown potential for creative work, especially in the challenging setting where scientific creativity is strictly scoped with practical limitations.

Finally, although my presence during the defense is not required, I was curious if the author could perhaps answer the following two questions during their defense:

1. All contributions in this thesis are for PGX.D, primarily working around its stage-hop execution model. How can the proposed techniques be generalized for other execution models for pattern matching (e.g., filter-process model from Arabesque, and pattern-aware model from Peregrine)?
2. Scouting queries is a neat idea. Graph processing works like PnP [ASPLOS ‘19] propose a similar strategy, although for a different use case. PnP evaluates point-to-point path queries by starting explorations in both forward and backward directions while collecting useful metrics (analogous to scouting plans), and then selecting the faster direction. How does the scouting queries technique relate to such direction prediction? Would these techniques be considered orthogonal such that both can be enabled together when executing a query?

   ``` PnP: Pruning and Prediction for Point-To-Point Iterative Graph Analytics. ASPLOS ’19.```
   ```

I hope the above assessment helped in evaluating the submitted thesis. I have no doubt the author will be able to successfully defend the thesis.

If I can be of any further assistance, please do not hesitate to contact me.

Sincerely,

Keval Vora
Associate Professor
School of Computing Science
Simon Fraser University
TASC 1 9419, 8888 University Drive
Burnaby, BC V5A 1S6
Canada
keval@cs.sfu.ca

Prof. Keval Vora is an Associate Professor in the School of Computing Science at Simon Fraser University. He received his Ph.D. from the Department of Computer Science and Engineering at the University of California, Riverside where he was advised by Prof. Rajiv Gupta. He was also a visiting researcher at the University of California, Irvine where he worked with Prof. Harry Xu. His research addresses challenges in building scalable modern data analytics systems, with a focus on graph data processing and management. His work lies at the intersection of runtime systems (often touching various parts of the technology stack) and algorithmic semantics (to build smarter solutions). He specializes in developing efficient techniques with provable guarantees for large-scale graph systems.