Ass. Prof. Riccardo Tommasini's Review

PhD Thesis: Distributed Graph Query Engine Improvements for Big Data Graphs
Author: Tomáš Faltín

The thesis of Tomáš Faltín contains sufficient contribution to be defended.

- Which of the contained results can be referred as a new scientific result?

aDFS and RPDq are definitely novel results in the literature. Scouting queries also look a very promising idea, which needs additional work to go beyond the proof of concept.

- What is the importance of it for the area and what are the possible applications to other (neighbouring) areas.

The presented contributions are relevant in the contexts of distributed graph processing, but also graph analytics and data systems in general.

General Comments

The thesis presents two technical contributions, namely aDFS and RPQd, for distributed graph processing.

The former is a protocol to interleave Bread-First search and Depth-First-Search and ultimately improve the performance of graph queries over distributed graphs. The performance improvement against native graph processing systems is evident. The other selected baselines are also outperformed.

RPQd focuses on the execution of Regular Path Queries and Reachability queries the same context of aDFS. The notable aspect is the memory bound, due to the reduction of intermediate results.

Additionally, the author showcases his work on Scouting Queries to demonstrate their positive impact in improving the planning of distributed graph queries. The author claims this is also a contribution, and I could agree. But the technical depth is substantially different than the other two. Therefore, I see it more as an methodological contribution with an empirical evaluation.

A major problem I observed is the lack of a problem statement and research questions. This passage is extremely important, probably essential, in the context of a PhD thesis.
The thesis starts well by identifying the challenges of distributed graph processing, but then it stops digging into the abstract issues to approach the problem in an very pragmatic way. Nothing is wrong with such an approach, although it becomes hard to understand the limitation of the thesis, define treats to its validity, and the

Quality of the Manuscript

The manuscript is well written and easy to follow in most of its parts (more details below). The clarity can be improved in the core parts of Section 4 and 5, where different executions are presented. Indeed, one of the main issues of the thesis is that it lacks a certain formal rigorousness. Only 4 definitions are provided, and they all are in the background.
The conceptualization behind the authors work is lost in the text and multiple reads are required to extrapolate the main concepts. A bit more structure in the terminological part would be of great benefit to the manuscript.

The lack of structure propagates also in the methodological part of the thesis, in the first chapter. The author starts with the identification of challenges, but then does not develop the reflection into a problem statement, research questions and research hypothesis. Also, the assumptions behind certain choices, e.g., the complexity of the queris in chapter 4 and 5, remain expressed informally in the text. Extrapolate research questions and hypotheses as well as assumption will help clarifying the scientific soundness and potential impact of the work.

The presented references are adequate to sustain the work. The author is aware of the state of the art, the most related works, and the theoretical foundations.

Although the literature review is extensive, the organization of the background knowledge could be improved. The author decided to organize the background as follow:
In Chapter 2, the general background.
In Chapter 3, the PGX.D/Async system used as a basis for the prototyping work.
In Chapter 4 and 5, the sections 4.1.1, 5.1.1, and 5.1.2 include the chapter related work.

The spread of the background across three chapters causes some repetitions and divides the attention about prior work. The preferred way is to collect all the prior knowledge within a single chapter, which acts as a foundational chapter to sustains the rest of the thesis, and then constantly refer to it across the various chapters.

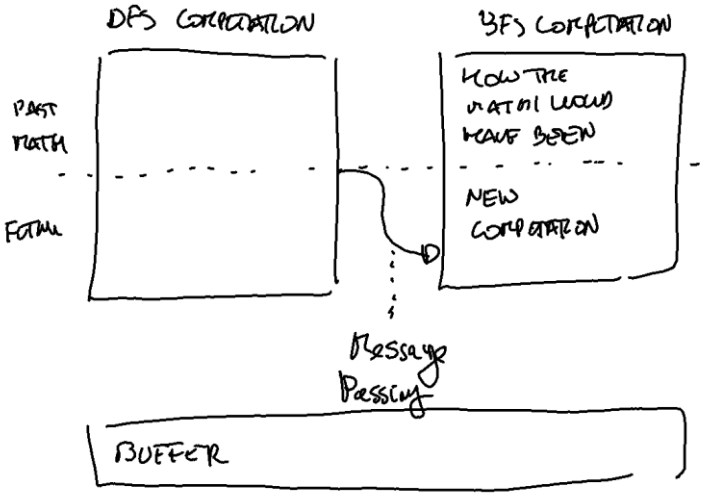
Finally, the conclusion is very short and not articulated. In my opinion this is the caused by the absence of proper research questions and hypothesis, that can lead to a methodological reflection about the work, and potentially spark new ideas.

A table with acronyms would be also good for referencing.

Detailed comments

Chapter 4

The very first period of section 4.1 need rephrasing.
Section 4.2.2, similarly to the related one in Chapter 3, really need an example with data that shows the evolution about the graphs and the DFS/BFS computations. I add an idea on how the computations could be represented. The basic idea is showing how the DFS and BFS would interleave within a figure in multiple stages.



Question: what is the schema for the PostgresSQL execution? Is TripleTable, Wide property table of vertical partitioning?The schema hugely impact the number of joins in the query translation, so how is this taken into account? The SQL query version should be included and discussed.

Question: Morpheus is used in the evalution,however it is no longer maintained. Perhaps SparkSQL would offer a good baseline in comparison to PostgresSQL. It would be possible to add experiments?

Question: how is the overhead in communication of aDFS vs traditional execution? Did you benchmark the message passing?

Chapter 5

Also this section really needs an example to make it easier to follow. Consider adding a running example in the introduction, that could help motivate the work. LDBC would be a good domain. Section 5.2.2. also needs the pseudo code and to visualize the iteraction

5.3.4 needs the queries listings in the discussion.

Fig 5.3 is lower the better, how is the absence of indexing ahs better result? Sorry I am confused.

Page 63: m in O(m^1/2) is not defined

Page 65 last paragraph “non-linear patterns.” Are undefined

Page 68 “Unbounded RPQs.” Are undefined

Page 71 “DFT-oriented” is undefined

Chapter 6

This chapter is much less developed than the other two. The idea is very interesting and potentially impactful, but it is missing a central aspect.

How do we obtain Scouting queries?

In page 77: << We create a scouting query for each of the N query plans.>> but does not explain how.

A number of assumption(to highlight) are listed in page 75

<< Scouting queries are better suited for large graphs and queries, as are typical for distributed graph engines, with any potential overheads amortized by the gains of the improved query plan. Additionally, scouting queries best fit engines with pipelined execution of pattern matching, i.e., engines that eagerly push intermediate results out as final.>>

Page 78
< This means that if the query plan QP1 returns more results than query plan QP2 in the same amount of time, we expect query plan QP1 to be better than plan QP2. Our assumption could fail in theory, as the engine could be lucky while executing a worse query plan.>

Is there a non-deterministic aspect in the scouting query generation?