Property Graphs at Scale: A Roadmap and Vision for the Future (short paper)

Haridimos Kondylakis[1], Vassilis Efthymiou[2,1], Georgia Troullinou[1], Elisjana Ymeralli[1], Dimitris Plexousakis[1]

[1] FORTH-ICS, N. Plastira 100, Heraklion, Crete, Greece, 70013
{kondylak, vefthym, troulin, ymeralli, dp}@ics.forth.gr
[2] Harokopio Univ. of Athens, Omirou 9, Tavros, Athens, Greece, 17778

Abstract. The prevalence and the rapid growth of interconnected data have sparked the rise of graph models and systems focusing on the management of large graphs now available both in research and industry. The property graph model allows the representation of information through multigraphs where nodes and edges can have labels and properties (i.e., key-value pairs). The model is becoming very popular and widespread, however related data management technology still faces many challenges, limiting the wide adoption of the model. In this vision paper, we present directions for future work in the domain focusing on the availability of a single declarative graph language, data integration, and scalable data processing. In our view, these areas represent key challenges for advancing research and practical solutions in the domain.

Keywords: Property Graphs, Graph Query Languages, Schema Discovery, Data Integration, Scalable Data Processing.

1 Introduction

Graphs serve as a versatile and dynamic data model, adept at representing intricate network-structured data across various application domains. These domains include social networks, biological networks, bioinformatics, cheminformatics, medical data, and knowledge management. Graphs, by their inherent nature, act as 'unifying abstractions,' harnessing interconnectedness to depict, explore, predict, and elucidate phenomena in both the real and digital worlds.

In our current landscape, we are witnessing an unprecedented surge in interconnected data, emphasizing the critical role of graph processing in our society. A recent article in CACM [1] highlights that "the future is big graphs". Instead of one single, compelling ("killer") application, we see big graph processing systems underpinning many emerging, but already complex and diverse data management ecosystems in many areas of societal interest [2]. The impact of this growth is evident, with sophisticated graph data management tasks already benefiting from big data processing systems.

To address the increasing prevalence of graphs, academia, start-ups, and major tech companies like Google, Facebook, and Microsoft have introduced diverse systems for managing and processing large-scale graphs. However, the current landscape remains fragmented, lacking a clear direction for the community. According to the same CACM paper, the Resource Description Framework (RDF) and Property Graph (PG) stand out as the most prominent data models for graph data management. RDF, a W3C recommendation, facilitates data sharing and interoperability scenarios, playing a central role in initiatives like Linked Data and FAIR data within the Semantic Web community. On the other hand, PG, on which we focus here, emerged in the context of enterprise data management. To merge these gaps RDF* [3] has been proposed, extending RDF with a convenient way to make statements about other statements, and practical approaches are pushed (e.g., neosemantics by RDF4J), however yet they have not been highly adopted and extensively supported by commercial vendors.

Property graphs are multigraphs where nodes and edges can have labels and properties (i.e., key-value pairs). The model is becoming very popular and widespread: PG solutions now serve 75% of Fortune 500 companies [5] and Gartner predicted that by 2025, graph technologies will be used in 80% of data and analytics innovations [6]. Note that, at the foundational level, all the models underlying graph database systems are subsumed by the PG model. The popularity of PG in the industrial community is justified by the fact that its development was picked by the main international standards body, namely ISO (International Organisation for Standardisation). However, diverse languages and systems for PG processing and analysis populate a fragmented market, thus causing a lack of clear direction for the research and industrial communities.

As such, a holistic view of data management technologies for property graphs is currently missing. In the following, we present our vision focusing on three distinct areas, i.e., a) the availability of a declarative graph language; b) the integration of disparate graph data, and finally c) how to enable PG data processing at scale, highlighting our vision for future work in the domain.

2 Roadmap to the future

2.1 Establishing a declarative query language for PGs

There are already well accepted languages for RDF graphs, such as the SPARQL language [7], whereas graph databases that adopt property graphs, e.g., Amazon Neptune, Neo4j, Oracle, SAP, TigerGraph, enable graph access via non-declarative APIs, such as Gremlin or, in the style of traditional relational databases, declarative languages, such as Cypher [8], PGQL [9], and GSQL [10]. An upcoming graph query language standard from ISO, called GQL [11], aims to unify these declarative languages, in a way SQL did for relational databases. For example, GQL is expected to be composable like Cypher, to fully support regular path queries like PGQL, and to eventually offer an expressivity as close as possible to Gremlin and GSQL, which are Turing-complete. The first version of the GQL Standard is scheduled to appear in early 2024, but it will have a number of important omissions. The two most notable omissions are support for

sophisticated graph schemas and a complete lack of support for graph-to-graph transformations. Indeed, GQL currently allows writing only graph-to-relational queries, which are adequate for many scenarios, though not all. For example, in data analytics tasks that require extensive data exploration, a user expects to see graphs as query answers [12]. To cater to these needs, existing PG solutions provide various ad-hoc tools (basic visualization, library functions for limited graph projection, etc.). However, a proper graph query language must treat graphs as first-class citizens, and support, among others query compositionality – i.e., the result of a query should be a graph itself allowing further queries on it. In fact, treating graphs as first-class citizens is a stated goal of GQL design [13]. However, compositionality has been dropped from the first version of GQL for a simple reason: there is no underlying research telling us how to add such facilities to the language.

Moving beyond the state of the art. Graphs constantly need to be transformed, to be updated with new information, and to be moved between applications. Our state of understanding of such transformations is very preliminary (as is indicated, for example, by multiple issues existing in the updating facilities of the leading graph query language Cypher). Crucially, we completely lack the framework for checking the correctness of such transformations, such as adherence to the typing information or being compatible with the requirements of an application that uses the output of a graph query.

The lack of research underpinnings for a proper graph-to-graph language is currently limiting the development of essential features like views, subqueries, and updates in graph query languages. Indeed, the first specification of the GQL industrial standard, which will appear in the first half of 2024, will still be a graphs-to-relations language borrowing its engine - pattern matching - from its purely relational counterpart SQL/PGQ. As such, a concentrated effort is required in order to understand and ultimately unlock the aforementioned features. In particular, more research is required on sophisticated types of graph modifications that we currently do not know how to perform safely and how to incorporate schemas into querying, which is yet another aspect of relational databases that is well understood and commonly used that requires much new foundational research for graphs. Future research could benefit from experience in formalizing languages for graphs (e.g., ICS-FORTH RDFSuite [14] and RQL [15]).

2.2 Data Integration

Data Integration is a well-established research area with tangible results for relational databases. However, the PG data model significantly differs from the relational one especially due to the fact that graph instances can be defined without a priori schema. While entity alignment has been studied for knowledge graphs in other data models (e.g., RDF) by leveraging ontologies and RDF types, schema-based PG integration is largely unexplored, due to the lack of definitions of schemas and constraints for this data model. In this direction, methods for schema discovery [16], which are necessary for establishing mappings [17] and transformation rules between multiple PGs, are currently underway. Property graph schemas [18] are being defined as part of the LDBC standardization activities and these definitions are expected to be adopted by graph database vendors. Schema inference methods can be used to extract standard schemas

from PGs and use them to specify mappings across different PGs. Similarly, an appropriate mapping language for data exchange and transformation [19] is still not present and should be defined on top of standard graph query languages. The correctness and the validity of the generated mappings are critical for processing tasks and, thus, there is a need for methods to ensure that the available mappings represent the intended transformations appropriately. Finally, the integration of PG with other graph data models (e.g., from/to RDF) is relevant, with appropriate characterization of cases where loss of information might occur.

Moving beyond the state of the art. Graphs provide a very flexible data model that makes it appropriate for integrating data from multiple disparate sources. Inspired by the work done for relational databases, future research in the domain should tackle several foundational issues for integrating PGs, starting by adapting to PGs the well-known three-layered architecture of relational data integration systems, i.e., sources to be integrated, the target providing an integrated view over the sources, and the mappings establishing the relationship between the sources and the target. In this direction, future research should carry both a thorough investigation of the complexity of basic foundational services of graph data integration; it should investigate the definition of a mapping language based on existing standards, capitalizing experience accumulated over the years for mapping languages (i.e., X3ML [20]), establishing the formal underpinnings of graph data integration and exchange. Future work should consider both schema-based and schema-less mappings: for the former, methods should be explored for discovering the schema of PGs that will be used for establishing mappings and transformations. For the latter different techniques will have to be devised. Finally, mappings between other data models (i.e. RDF) should be explored, both for the sources and the target, with appropriate characterization of cases where loss of information might occur.

2.3 Scalable Data Processing

Graph data management and processing systems have been adopted by many companies and organizations, but the gap in their adoption for business intelligence use cases and analytical tasks is still substantial. Indeed, practitioners are missing guidelines and best practices that can help them identify non-trivial applications of graph analytics tools and approaches beyond one-shot operations. Moreover, we have only recently witnessed the creation of data management tools that are able to map, within themselves, data imported from multiple sources and models. Thus, their performance, capabilities, and limitations when trying to address hybrid transactional and analytical workflows are still largely unexplored [21, 22]. Further, as schema on top of graph data sources is not available, methods for schema-based data partitioning and query optimization are still missing.

Moving beyond the state of the art. State-of-the-art research is already focusing on schema discovery [16, 23] schema-based data partitioning using big data infrastructures [24, 25] and analytical tasks through materialized views [26] for knowledge graphs. Such approaches can fuel the development of similar solutions for PGs. Future

research in the domain should focus on partitioning for big graph data, exploring summary-based partitioning and hierarchical schemas for improving query answering efficiency. Further, summaries will be exploited as materialized views for further speeding up query answering, whereas there is room for focusing not only on exact but also on approximate answers.

3 Conclusions

The property graph data model has emerged as a versatile and powerful paradigm for data management and its penetration in the industry is constantly increasing. However, in order to enable effective and efficient management of the data, a set of problems should be first tackled starting from establishing a powerful declarative query language, defining and discovering schemas for property graphs, and enabling data integration. Further techniques for data processing at scale, such as data partitioning, should also be devised in order to facilitate efficient querying. Already, the first steps in all these domains exist, however, a holistic solution in the domain is still missing.

Acknowledgments. The work reported in this paper is implemented in the framework of H.F.R.I call “Basic research Financing (Horizontal support of all Sciences)” under the National Recovery and Resilience Plan “Greece 2.0” funded by the European Union – NextGenerationEU (H.F.R.I. Project Number: 16819).

References

1. Sakr, S. et al.: The future is big graphs: a community view on graph processing systems. Commun. ACM 64(9): 62-71 (2021).
2. Hegeman, T., Iosup, A.: Survey of graph analysis applications. arXiv preprint arXiv:1807.00382 (2018).
3. RDF-star and SPARQL-star. Draft Community Group Report. Available Online: https://w3c.github.io/rdf-star/cg-spec/editors_draft.html, visited April 2024.
4. NeoSemantics, Neo4j RDF & Semantics toolkit, Available Online: https://neo4j.com/labs/neosemantics/, visited April 2024.
5. Record investment in Neo4j suggests, maybe it IS all about relationships, https://www.hfsresearch.com/research/record-investment-in-neo4j-suggests-maybe-it-is-all-about-relationships/, last accessed 2024/02/26.
6. Gartner Identifies Top 10 Data and Analytics Technologies Trends, https://www.gartner.com/en/newsroom/press-releases/2021-03-16-gartner-identifies-top-10-data-and-analytics-technologies-trends-for-2021
7. SPARQL 1.1 Query Language, W3C Recommendation, Available Online: https://www.w3.org/TR/sparql11-query/, visited March 2024.
8. Nadime F., et al.: Cypher: An Evolving Query Language for Property Graphs. In Proceedings of the 2018 International Conference on Management of Data (SIGMOD '18). Association for Computing Machinery, New York, NY, USA, 1433–1445 (2018). https://doi.org/10.1145/3183713.3190657.

9. van Rest, O., Hong, S., Kim, J., Meng, X., & Chafi, H.: PGQL: a property graph query language. In Proceedings of the Fourth International Workshop on Graph Data Management Experiences and Systems (2016).
10. Deutsch. A.: Querying Graph Databases with the GSQL Query Language. SBBD. 313 (2018)
11. GQL Standards, https://www.gqlstandards.org/, last accessed 2024/02/26.
12. Lissandrini, M., Mottin, D., Palpanas, T., Velegrakis, Y.: Graph-Query Suggestions for Knowledge Graph Exploration. In Proceedings of The Web Conference 2020 (WWW '20). Association for Computing Machinery, New York, NY, USA, 2549–2555. (2020). https://doi.org/10.1145/3366423.3380005
13. Deutsch, A., et al.: Graph Pattern Matching in GQL and SQL/PGQ. In Proceedings of the 2022 International Conference on Management of Data (SIGMOD '22). Association for Computing Machinery, New York, NY, USA, 2246–2258 (2022). https://doi.org/10.1145/3514221.3526057
14. Alexaki, S., Christophides, V., Karvounarakis, G., Plexousakis, D., Tolle, K.: The ICS-FORTH RDFSuite: Managing Voluminous RDF Description Bases. SemWeb 2001.
15. Karvounarakis, G., Alexaki, S., Christophides, V., Plexousakis, D., Scholl, M.: RQL: a declarative query language for RDF. WWW, 592-603 (2002).
16. Kellou-Menouer, K., Kardoulakis, N., Troullinou, G. et al: A survey on semantic schema discovery. The VLDB Journal 31, 675–710 (2022).
17. Rahm, E., Bellahsene, Z., Bonifati, A. eds.: Schema matching and mapping. Springer (2011).
18. Renzo, A., et al.: PG-Keys: Keys for Property Graphs. In Proceedings of the 2021 International Conference on Management of Data (SIGMOD '21). Association for Computing Machinery, New York, NY, USA, 2423–2436 (2021). https://doi.org/10.1145/3448016.3457561
19. Boneva, I., Bonifati, A., Ciucanu, R.: Graph Data Exchange with Target Constraints. EDBT/ICDT Workshops,171-176 (2015)
20. X3ML Toolkit: https://www.ics.forth.gr/isl/x3ml-toolkit, last accessed 2024/02/26.
21. Mhedhbi, A., Lissandrini, M., Kuiper, L., Waudby, J., Szárnyas, G.: LSQB: a large-scale subgraph query benchmark. In Proceedings of the 4th ACM SIGMOD Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA) (GRADES-NDA '21). Association for Computing Machinery, New York, NY, USA, Article 8, 1–11 (2021).
22. Lissandrini, M., Mottin, D., Hose, K., Pedersen, T.B.: Knowledge Graph Exploration Systems: are we lost? CIDR (2022).
23. Kardoulakis, N., et al.: HInT: Hybrid and Incremental Type Discovery for Large RDF Data Sources. SSDBM, 97-108 (2021).
24. Troullinou, G., Agathangelos, G., Kondylakis, H., Stefanidis, K., Plexousakis, D.: DIAERESIS: RDF Data Partitioning and Query Processing on SPARK, Semantic Web Journal, (2024).
25. Bonifati, A., Dumbrava, S., Kondylakis, H., Troullinou, G., Vassiliou, G: PING: Progressive Querying on RDF Graphs. ISWC (Posters/Demos/Industry) 2023
26. Troullinou, G., Kondylakis, H., Lissandrini, M., Mottin, D.: SOFOS: Demonstrating the Challenges of Materialized View Selection on Knowledge Graphs. SIGMOD Conference (2021).