Graph Database in Practice 2020 (GDBP 2020)

GDBP 2020 attendees*

No Institute Given

Abstract. Graph database is one of the emerging technology gathering attention among various database management systems (DBMS). However, its practical utilisation is just started and still in discovery. Graph Database in Practice 2020 (GDBP 2020) aims to facilitate the knowledge sharing on the actual use cases and methods to utilize graph databases in the real-world applications, and also discussions on future development of tools and their eco-systems to accelerate the use of graph databases. For potential users, this is the best opportunity to learn about the technology itself as well as its use cases. Also, for researchers and developers, this is a unique opportunity to share your ideas and actual problems in industry, and to engage in international collaborative projects.

Keywords: Graph Database · Property Graph

1 Introduction

As a backend of A.I. systems, database management systems (DBMS) are dramatically evolving. Among various DBMS, graph database is one of the emerging technology gathering attention in various industries. Its capabilities to manage relationships of entities and to execute graph algorithms have potential to increase the information coverage of AI input. However, its practical utilisation is just started and still in discovery.

Graph Database in Practice 2020 (GDBP 2020) aims to facilitate the knowledge sharing on the actual use cases and methods to utilize graph databases in the real-world applications, and also discussions on future development of tools and their eco-systems to accelerate the use of graph databases. For potential users, this is the best opportunity to learn about the technology itself as well as its use cases. Also, for researchers and developers, this is a unique opportunity to share your ideas and actual problems in industry, and to engage in international collaborative projects.

We created three groups for use cases, including customer analysis, BoM management, and network biology. We also have three groups for development, including algorithm, format, and API.

2 Use Cases

2.1 Customer Analysis

Background While data integration is a mutual challenge for various industries, different industries may have different purposes for utilizing the data. For example, in the financial industry, the motivation can be finding beneficial accounts or fraud accounts. In the material industry, finding relationships between distributors and retailers is important for effective launching of new products. In retail, they like to increase their customer loyalty by analyzing the relation between product and membership.

Customer Analysis has become a key pillar in multiple businesses and is being used for a wide range of use cases, from campaign segmentation to product recommendation. It generally makes use of discrete data such as customer's information (age, location, gender, etc...). But sometimes, those discrete data are not enough to correctly classify customers, and a better segmentation can be done by using connected data such as activity logs which represent user's behavior.

Discussion Summary We propose that there are three main benefits of using graph databases, "graph-based data integration", "data visualization", and "analysis using graph algorithms".

From the data integration perspective, 360-degree analysis needs a single golden profile which sees the customer as a single, complete, multifaceted person, not a series of unrelated brand contacts. That means not just pulling data from multiple sources, but also combining, sanitizing and consolidating that data. It's difficult to meet the requirements by RDB instead of Property Graph Database.

In data visualization, Visualizing information is crucial in the strategy discussion of management level because it can paint the same picture to all team members.

Regarding analysis using graph algorithms, you can identify the valuable group out of a pile of customer data. It's very hard to do the same task with other data structures.

Future Tasks Customer analysis makes use of more and more data, and we need to be careful about laws regarding the usage of it. Laws are not the same everywhere and they also evolve fast as the analysis field grows. Thus, compliance is an everyday effort and graph technology, due to its flexible nature (non-fixed schema), will be faster to adapt to those changes, compared to traditional RDB.

2.2 BoM Management

Background Manufacturing industries, such as automotive companies, have multiple databases of component specification (so called, bill of materials) and its production information. They often need to collect information combining those databases, for example, to simply count the number of necessary parts according to purchase orders, or to investigate all parts affected by a design change of a particular part. For personalising products as well as satisfying traceability requirements, a scalable and efficient data management system is essential.

Since graph traversal operations are optimal for such operations, graph plays an important role in logistics, configuration, and supply chain applications. Take logistics for example, movement of goods, transportation network, points of interest can be modeled as a graph. Based on the interoperable nature of graph data model, queries and analytics can be utilized to track up-to-date status, compute shortest path, and estimate arrival time, much more efficiently than non-graph based solutions.

Discussion Summary Trees are used to represent products. The basic idea is the node represents a piece of the product and edges means the parent "comprise" its children part. The root is the whole product and the leaves are atomic parts that don't need to be split anymore. Where the limit depends on the domain, usually are components that are raw material or bought outside.

Nowadays, manufacturers are required "Mass Customization". Please imagine the case when you buy your PC, you usually customize Companies often sell many similar products and so the resultings trees would be really similar. We want to use a graph database to reduce the repetition as much as possible and being able to generate the trees when needed from the specs.

Future Tasks As for the management of each product, that natively creates a hierarchical graph structure without circulative nature. It can use as a fundamental representation of the component manufacturing process. With this representation, we can enjoy the following areas: manufacturing cost reduction, and parts conflicts detection (it contributes the quality of BoM).

Selecting a structure of the graph should remain in the problem to solve. There is no swiss army knife to the given problem. Picking up the best graph structure may still be in a challenge to tackle. We have brainstormed a suitable graph structure in this domain problem and reached a different type of graph structure (e.g. decision tree structure, folded graph structure) for a specific demand.

2.3 Graph for Biologists and Network Biology

Background A biological pathway is a network of genes interacting for a specific function in a cell. Many pathways have been established with great details in model organisms, but not much for non-model organisms. The pathway can be easily modeled using a graph database with a weight from experimental data for example. Biologists usually compare a graph from two or more treated groups to a control group to identify genes or proteins that play an important role in diseases or differences in biology. Biologists rely on a public pathway network to learn about the interaction between genes and proteins; therefore, the incompleteness of the pathway is a major issue in biological network analysis. There are many sources of pathways networks available on the Internet and some have APIs and visualization tools.

Discussion Summary Is it possible to do "common" network biology analysis on top of graph databases or not? Biologists have only their research questions in mind, most of which could be tackled by existing computational tools/algorithms. However, they have no time and background to explore the tools suitable for their problems. Thus, it calls for computer scientists and software engineers to introduce computational tools to them. Nevertheless, the gap in communication exists and it is quite challenging to fill due to the difference in their background.

What kind of application is suitable for graph modeling, and what kind application is not? Oftentimes, data analysis is not a solution to the problem in biological research but a starting point. Biologists use data analysis to develop hypotheses to be tested by an experiment. Therefore, a tool that is easy-to-use and helps biologists explore or learn about their data quickly is very useful. GENEMANIA is a good example of layered visualization of network in an interactive manner allowing users to select a specific layer and to analyze what is happening in the dataset.

Besides biological research, a graph can be used to model relationships of various entities and relationships in medical research. One use case is a network representation of medical terms or diseases and bacteria that co-occur in a research paper. It could be thought of as a bipartite graph, comprised of bacteria and terms. This sort of graph is not meant to be used for graph analysis but to provide access to linked information for a quick decision making or hypothesis generation.

Future Tasks If we can specify the needs of a graph in biology, it will be a good input for developers to build tools. Apparently, we need more computer scientists, software engineers to create a tool for biologists. And we need biologists who know how to use tools to create tutorials for practical use. Good examples of community efforts are a cookbook such as BioPython and SciRuby.

The application differs among research fields and most biologists don't want to learn a query languages. Therefore, we need to build a tool for biologists to make use of a graph database with little effort. In addition, good learning materials such as tutorial sessions will help lower the barrier for biologists to adopt a graph database for their research problem.

Also, we need more developers to build user-friendly and high quality tools. To attract more developers, we need to provide a resource for developers who are only interested in specific computer problems to explore to choose a project to contribute. For example, biopython is a project that collects a bunch of tools and tutorials (cookbook for graph databases at https://biopython.org/) by communities.

Source code from academia therefore must be available on Github to be open for contribution and to prolong the life of the project after the students have graduated. Also, promoting code sharing and good programming practice should be good to do. We should create a github organization to collect related tools and increase interaction between Bio Science and IT like biocontainers(https://biocontainers.pro/). https://github.com/gdbp/

It would also be beneficial in the future if the tools developed for biological research can be translated or used outside academic domains.

We further propose to have a “hackathon” for doing such community activities and accelerating to provide Problem: What’s the profit for committers / communities? Using github open-source projects promotes communities to invite new developers, rather than using private pages. Example: BioHackathon series held in Japan. NASA hackathon

3 Tool Development

3.1 Algorithms on Database

Background There are many algorithms on databases, so the topic “Algorithm on Database” is too broad to discuss. So we need to narrow the range of the discussion. We decided to discuss these two topics. The first one is the intuitions and feasibilities of results of graph algorithms on labeled property graph. The other one is pros and cons of using domain specific languages.

Discussion Summary Intuitions and feasibilities about algorithms on labeled property graphs: On a simple graph, the betweenness centrality outputs nodes (or edges) order by how impactful its removal from the algorithm would be in the all-to-all shortest paths. Considering node and edge labels, it is not easy to understand an intuition about such a result. Graph databases provide many algorithms [1] on graph structure with multi-labeled nodes. However, it’s hard for users to decide whether the algorithm works or not.

Pros and Cons of using domain specific languages: Pros and Cons of using domain specific languages on graph databases or analyzers instead of a general purpose programming language like Java or C++.

Future Tasks More discussions are necessary to identify the issues on algorithms on Graph Databases.

3.2 Property Graph Exchange Format

Background Increasing amounts of scientific and social data are described and analyzed in the form of graphs. In the context of graph analysis, the property graph model is becoming popular; various graph database engines, including Neo4j, Oracle Labs PGX, and Amazon Neptune, adopt this model. These graph database engines support powerful algorithms for traversing or analyzing graphs. In contrast to the standardized RDF, however, property graphs lack a standardized data model. We considered the general requirements for representing property graphs and designed two serialization formats as flat text and JSON. These formats can be converted into specific formats for each of the databases mentioned above. The serialization formats independent of certain database implementations will increase the interoperability of graph databases and will make it easier for users to import accumulated graph data. The format is inspired by RDF. Concatenating the PG file will results in integration of the dataset.

Discussion Summary We tried to incorporate differences in data modeling in property graph databases. There still remains issues to be addressed. Do we need both directed and undirected nodes? Practically, relationships like gene co-expression should be represented as undirected edges. Even if the PG format has an advantage in integrating different datasat just by concatenating the PG files, more use cases are necessary to def If there is a standardized format, it will promote open data sharing by publishing in a common format. Inspired by RDF, using URI as identifiers will help integrate data from various data sources. It will also enhance development of tools or eco-systems for graph databases. What is the bottleneck to develop and spread the general format? Unlike the case of RDF, where various implementations follow the standardized format, in the case of property graph, many implementations emerged before standardizing the model and formats.It is probably difficult to standardize the format.

Future Tasks There still remains issues in data modeling. For example, supporting undirected edges. It will be a hard task to develop more converters into each of the database implementations. Ideally, many implementations should support a common format. We need to share information on similar activities world wide.

3.3 Graph Database API

Background There are many benefits from using graph databases, but there are still some problems. Relational databases have SQL as a common language, but graph databases currently do not have a common language. Once you learn SQL, querying against relational databases is easy, but in the case of graph databases knowledge of database-specific query languages and graph formats is necessary. Whilst the standardization effort of graph query languages is going on, the API (including the essential inputs, parameters, data access, expected outputs...) for graph-based operations are not widely discussed.

There is a prototype implementation of the APIs supporting multiple graph databases, “X2” (pronounced “Cross To”). X2 aims to collect common queries for graph visualization and provide them as a REST API. X2 is also expected to mitigate the burden to support different query languages, communication protocols, and output. On top of X2, we explored the future direction of the graph database API.

Discussion Summary GraphQL and the corresponding Neo4j library provides a good abstraction. You just need to define the mapping based on your DB schema and it will generate all CRUD operations, and also enabling you to traverse your graph through relationships without any business logic. For complex queries, you can directly write Cypher queries. GraphQL provides more flexibility and the cost to adapt every change is cheaper compared to a REST implementation.

Future Tasks Investigate on how to execute operations on the whole database using GraphQL (PageRank, community detection, ...), which add or update properties and relationships. Need to write a parser library for every database (PGX, TigerGraph, ...), using Neo4j's implementation as a starting point. Define "frequent" queries on graph visualization projects, to use as a metric to set a goal like "aim to support 90Automatically generate the GraphQL template file from the DB schema.

4 Conclusion

We hold a one-day workshop in Graph Database in Practice 2020 with about 30 attendees, which include researchers and engineers from both industry and academia. The participants are current or future users or developers who work on the utilization of graph databases.

In this workshop, we have selected 3 possible use cases of graph database and 3 tool development tasks which accelerate the use of graph database. While the users in the domains have concrete expectations on the applications of the technology, the knowledge of system development on graph databases is not shared well yet. We have also clarified that we need not only graph database itself, but also the methodology of application development on it.

We have also agreed that the intense workshop is useful for knowledge sharing while we need longer time for prototyping the ideas, both in use cases and in tool development. Since the matured graph databases are available, the further community effort should accelerate the actual utilization.

Acknowledgements

We appreciate the all participants of GDBP 2020 Workshop at Bangkok.

References

1. Angles, R., Gutierrez, C.: An introduction to Graph Data Management. arXiv preprint arXiv:1801.00036 (2017)
2. The Neo4j Graph Platform. `https://neo4j.com/`
3. Oracle Labs Parallel Graph AnalytiX (PGX). `https://www.oracle.com/technetwork/oracle-labs/parallel-graph-analytix/overview/index.html`
4. Amazon Neptune. `https://aws.amazon.com/neptune/`
5. 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, p. 7. ACM (2016)
6. RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation 25 February 2014. `http://www.w3.org/TR/rdf11-concepts/`

7. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., ... Bizer, C. (2015). DBpedia–a large-scale, multilingual knowledge base extracted from Wikipedia. Semantic Web, 6(2), 167-195.
8. Vrandečić, D., Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. Communications of the ACM, 57(10), 78-85.
9. SPARQL 1.1 Query Language, W3C Recommendation 21 March 2013. `http://www.w3.org/TR/sparql11-query/`
10. Angles, R., Arenas, M., Barceló, P., Hogan, A., Reutter, J., Vrgoc, D.: Foundations of Modern Query Languages for Graph Databases. ACM Computing Surveys (CSUR), 50(5), 68. (2017)
11. Angles, R., Arenas, M., Barceló, P., Boncz, P., Fletcher, G., Gutierrez, C., *et al.*: G-CORE: A core for future graph query languages. In Proceedings of the 2018 International Conference on Management of Data, pp. 1421–1432. (2018)
12. W3C Workshop on Web Standardization for Graph Data. `https://www.w3.org/Data/events/data-ws-2019/`
13. Chiba, H., Yamanaka, R., Matsumoto, S. (2019). Property Graph Exchange Format. arXiv preprint arXiv:1907.03936.
14. Franz, M., Rodriguez, H., Lopes, C., Zuberi, K., Montojo, J., Bader, G. D., Morris, Q. (2018). GeneMANIA update 2018. Nucleic acids research, 46(W1), W60-W64.