PandaDB: An AI-Native Graph Database for Unified Managing Structured and Unstructured Data*

Zihao Zhao[1,2], Zhihong Shen[1(✉)], Along Mao[1,2], Huajin Wang[1], and Chuan Hu[1,2]

[1] Computer Network Information Center, CAS, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
bluejoe@cnic.cn, airzihao@gmail.com, {alongmao, wanghj, huchuan}@cnic.cn

Abstract. In many applications, data are organized as graphs (e.g., social network and smart city). There could be unstructured data on such a graph, for example, the users' avatars and images included in a post. It is natural to think of these unstructured data as attributes of nodes or relationships. Then the users would tend to query the semantic information of unstructured data on the graph, namely hybrid queries. To meet the demand of hybrid queries, this paper introduces PandaDB, an AI-native graph database, and it has the following characteristics: (1) Unified management of unstructured data and graph data. (2) Online extracting and indexing semantic information of unstructured data. (3) Optimization of hybrid queries. The system and its concept have been verified by multiple applications based on it. Users could deploy PandaDB to support hybrid queries and data mining.

Keywords: Graph Database · AI · Unstructured Data.

1 Introduction

In many applications, data are organized and managed as graphs, for example, social network [1] and the smart city [2]. An entity (e.g., a person or a university) is usually regarded as a node. The relationships (e.g., *studyAt* and *workFor*) between entities are regarded as edges. The attributes of entities and relationships are taken as properties (e.g., a person's name and birthday). In real applications, unstructured data (e.g., images, texts, audio, and videos) and structured data are often used to describe the properties. These unstructured data are properties of the entities. There are relationships between entities. Thus potential relationships exist among the multimedia data. For example, users who post similar content tend to construct a tight community (a.k.a, cluster). Two users who are thought unrelated are likely to know each other if they appear in the same photo.

* This work was supported by the National Key R&D Program of China(Grant No.2021YFF0704200) and Informatization Plan of Chinese Academy of Sciences(Grant No.CAS-WX2022GC-02)

These two features would obviously help to recommend potential friends for the users. Thus, users tend to query the semantic information of multimedia data on the graph, namely hybrid queries. It implies the requirement of a database that could natively support hybrid queries.

Given the state-of-the-art works in both academic and industry communities, we are still facing the following challenges: (1) The graph databases can not manage and understand the semantic information of unstructured data. (2) The query language can not describe the statements of querying unstructured data on the graph [3]. (3) The newly introduced operations of unstructured data make it hard to optimize hybrid queries for databases. To solve these challenges, we propose an AI-native graph database, namely PandaDB[3], with the following characteristics: (1) Unified management of unstructured data and graph data. (2) Offering an extended query language to help users to query the unstructured data in a graph. (3) Online extracting and indexing of the semantic information of unstructured data, and optimizing the hybrid query plans according to the query features and data characteristics.

2 System Overview

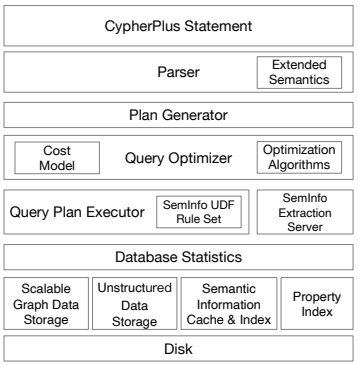

Fig. 1. Architecture of PandaDB

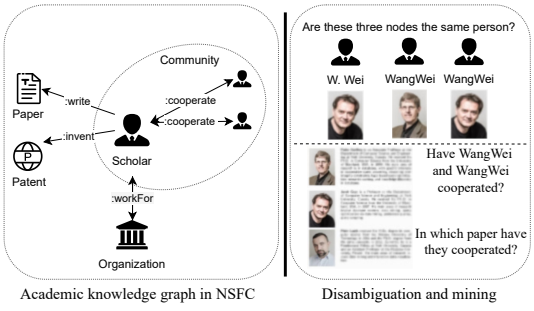

Fig. 2. Academic Graph Disambiguation

Figure 1 illustrates the architecture of the proposed system PandaDB. Compared with traditional databases, it is updated and enhanced on these modules and components: *Parser*, *Query Optimizer*, *Query Plan Executor*, *Unstructured Data Storage* and *Semantic Information Cache/Index*. The AI technology (e.g. AI models to understand unstructured data) is natively supported.

Parser: We proposed a new query language, namely CypherPlus, to facilitate the description of hybrid queries, with introducing new functions: *BLOB Functions*, *Semantic Information Extractor* and *Logical Comparison Symbols*. The parser are modified to support these newly introduced semantics.

Query Optimizer is to optimize the hybrid query plans. The cost model is designed to support estimating the cost of unstructured data operations in a hybrid query plan. The optimizer would observe the data statistics and distribution in real time, estimate each operation's cost by the cost model and

[3] The project is open-sourced at: https://github.com/grapheco/pandadb-v0.1

re-order the operations by the optimization algorithm to get a query plan with minimal estimated cost. There are traditional optimization rules (e.g. predicate push down), the newly introduced operations of unstructured data extend the ordinary rules. For example, the combination of a *SORT* and *LIMIT* operation of unstructured data could be transferred to a kNN search operation of the semantic information in vector format. The latter would be much cheaper than the former. The optimization algorithm will take the new rules into consideration.

Query Plan Executor: The extraction and logical computation of semantic information are treated as UDFs (short for user defined functions) in PandaDB, the *SemInfo UDF Rule Set* defines how to execute these functions. PandaDB extracts the semantic information by AI models. A kind of semantic information corresponds to an AI model (one-to-one map). The AI models are maintained by the *SemInfo Extraction Server* and offer extraction service for the executor. Users are allowed to define the mapping between semantic information name and AI model in the *SemInfo UDF Rule Set*. Similarly, users could define the rules of semantic information logical computation. PandaDB offers a default implementation of *SemInfo UDF Rule Set*.

Unstructured Data Storage: PandaDB treats unstructured data as BLOB (Binary Large Object) and takes BLOB as the first-class citizen in the system. Traditional graph databases (e.g. Neo4j) , store unstructured data as *ByteArray* without the meta-data (e.g. the ID, MIME-TYPE and length). While in PandaDB, the query engine could obtain all the information of unstructured data, for example, the meta-data, unstructured contents, statistics and distribution information, and further, optimize the queries with reference to these information.

Semantic Information Cache and Index: PandaDB caches and indexes the semantic information to accelerate the query response. PandaDB offers two modes, namely, eager mode and lazy mode. Both of them would try to retrieve data from the cache when dealing with a query. The difference lies when they extract the semantic information. The eager mode would pre-extract all the semantic information and cache them. While the lazy mode does not extract the semantic information before it is queried for the first time.

3 Demonstration Scenarios

Academic graph disambiguation and mining. NSFC (National Natural Science Foundation of China) stores and manages data about scholars, published papers, academic affiliations and scientific research funds details. Figure 2 shows the data overview in NSFC. There are about 1.5TB of data, with 2 million scholars. Three example queries are shown in Figure 2. All of them involve unstructured semantic information. About sixty different types of queries similar to these are carried on the system. PandaDB is deployed to support the NSFC-KBMS[4], namely the Knowledge Base Management System of NSFC. We use OCR technology to extract the author and scientific research organization information from the PDF files of the papers, then construct the corresponding

association relationships between authors and their corresponding universities. This affiliation is used to build the connection between two graph nodes, as shown in Figure 3.

Entertainment System. We extend the biggest movie comment and review dataset in China[4]. It contains more than 100 million movies, super stars, comments and users. We built a graph containing actors, movies, participation relationships and unstructured data (e.g. actors' photos and snapshots). PandaDB is evaluated to help users to find the super star in this graph. When the user submits a photo, PandaDB can find the superstar share the similar photo as the facial information of the input photo, then find the film in which the actor has played from the graph. This system is deployed and used in the production environment, and one demo video is in the link[5]. Figure 4 is the screenshot of the demo video, it shows how the GUI acts with PandaDB. The front-end application shows the content of the graph, which includes the images of actors. The user uploads an image and wants to query who is the man in the image he uploaded. The system should first extract the facial feature in the uploaded image, then compare with all the images in the database, and finally return the result. For this query, the front-end application needs only send a query to PandaDB, the statement is:

```
MATCH(n) WHERE n.image IS NOT NULL AND n.image <: IMAGE return n;
```

Fig. 3. Big Data Knowledge Service on NSFC

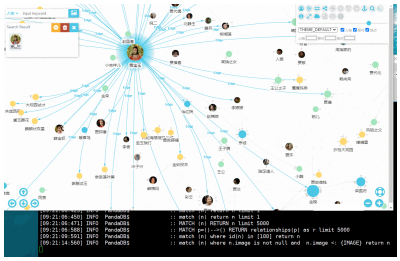

Fig. 4. Entertainment System

4 Conclusion

In this work we proposed an AI-native graph database system to meet the demands of hybrid queries. The plan optimizer and executor are designed to support hybrid queries optimization and execution, respectively. The system would then be able to understand the semantic information of unstructured data. Unstructured data could be efficiently managed by the storage layer. Finally, an new query language, namely CypherPlus is proposed to facilitate the hybrid query description.

[4] http://moviedata.csuldw.com/
[5] https://github.com/grapheco/pandadb-v0.1/blob/master/demo.gif

References

1. Erling, Orri, et al. "The LDBC social network benchmark: Interactive workload." Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data. 2015.
2. Usman, Muhammad, et al. "A survey on big multimedia data processing and management in smart cities." ACM Computing Surveys (CSUR) 52.3 (2019): 1-29.
3. Francis, Nadime, et al. "Cypher: An evolving query language for property graphs." Proceedings of the 2018 International Conference on Management of Data. 2018.
4. Shen Zhihong, Yao Chang, Hou Yanfei, Wu Linhuan, Li Yuepeng. Big Linked Data Management: Challenges, Solutions and Practices[J]. Data Analysis and Knowledge Discovery, 2018, 2(1): 9-20