MMDBench: A Benchmark for Hybrid Query in Multimodal Database

Along Mao[1,2], Chuan Hu[1,2], Chong Li[1], Huajin Wang[1], Junjian Rao[1,2], Kainan Wang[1,2], and Zhihong Shen[1](✉)

[1] Computer Network Information Center, Chinese Academy of Sciences, Beijing, China
{almao,huchuan,lichong,wanghj,jjrao,knwang,bluejoe}@cnic.cn
[2] University of Chinese Academy of Sciences, Beijing, China

Abstract. Multimodal data, integrating various types of data like images, text, audio, and video, has become prevalent in the era of big data. However, there is a gap in benchmarking specifically designed for multimodal data, as existing benchmarks primarily focus on traditional and multimodel databases, lacking a comprehensive framework for evaluating systems handling multimodal data. In this paper, we present a novel benchmark program, named MMDBench, specifically designed to evaluate the performance of multimodal databases that accommodate various data modalities, including structured data, images, and text. The workload of MMDBench is composed of eleven tasks, inspired by real-world scenarios in social networks, where multiple data modalities are involved. Each task simulates a specific scenario that necessitates the integration of at least two distinct data modalities. To demonstrate the effectiveness of MMDBench, we have developed a hybrid database system to execute the workload and have uncovered diverse characteristics of multimodal databases in the execution of hybrid queries.

Keywords: Benchmark · Multimodal Database · Hybrid Query

1 Introduction

In the era of big data, the quantity and variety of data are growing at an unprecedented pace. Among these diverse data types, multimodal data has garnered significant attention. Multimodal data refers to the integration of multiple modes or types of data, such as images, text, audio, and video. This data often contains abundant and complementary information, enabling a more comprehensive understanding of underlying concepts and phenomena. Multimodal data has become increasingly prevalent in various domains, including social media analysis [4], healthcare [3], knowledge graph (Fig. 1) [15], and so on. Moreover,

Supported by National Key R&D Program of China(Grant No. 2022YFF0711600), National Key R&D Program of China(Grant No. 2021YFF0704200) and Informatization Plan of Chinese Academy of Sciences(Grant No. CAS-WX2022GC-02).

S. Hunold et al. (Eds.): Bench 2023, LNCS 14521, pp. 87–103, 2024.
https://doi.org/10.1007/978-981-97-0316-6_6

the emergence of artificial intelligence technologies has provided robust support and impetus for multimodal data analysis, enabling effective exploration and utilization of the latent information within multimodal data.

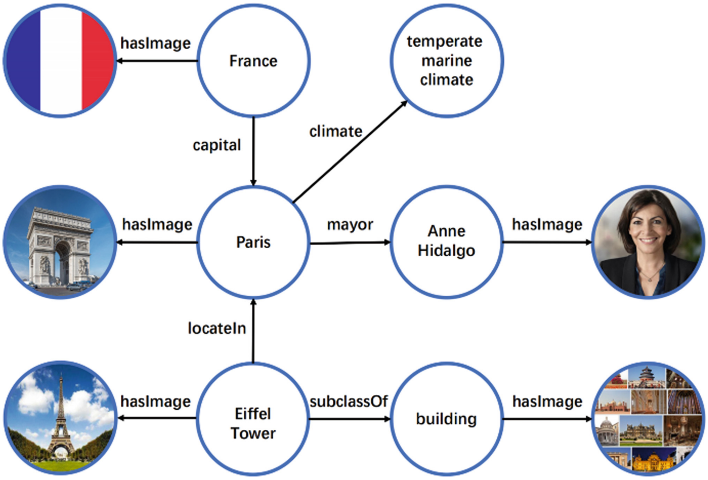

Fig. 1. An Example of Multimodal Data: Multimodal Knowledge Graph [21]

Despite significant advancements in benchmarking techniques for traditional databases and even multimodel databases, there still exists a gap when it comes to benchmarking specifically designed for multimodal data. Existing benchmarks primarily focus on relational databases [12], NoSQL databases [5], or evaluate the performance of multimodel data management systems [7,10,18,19]. In recent years, there has been an emergence of multimodal data management systems [16,20] that can handle both structured and unstructured data. However, there is a lack of comprehensive benchmarking frameworks specifically tailored for evaluating the performance of systems handling multimodal data.

In order to evaluate the performance advantages and bottlenecks of such systems in executing hybrid queries, enforce manufacturers to continuously improve the performance of the system, and promote the further development of new database technology, we put forward a benchmark which is called MMDBench. As shown in Fig. 2. It provides a multimodal data generator and a multimodal data analytic workload in social network scenario. The contributions of MMDBench are as follows:

– **Data Generator.** We have developed a generator capable of producing multimodal data in social network scenarios. It uses the property graph model as the foundation to associate unstructured data such as text, and images with the graph data. The generator supports the generation of data in various scales while adhering to the distribution patterns observed in real-world scenarios.

- **Query Workload.** We have designed a workload for hybrid queries that simulate typical operations of querying structured and unstructured data in social networks.
- **Benchmark Framework.** We have designed and implemented a unified framework that provides interfaces for system integration to facilitate the completion of benchmark testing. This framework serves as a standardized platform for evaluating different systems under consistent conditions, ensuring fairness and comparability in performance evaluations.
- **Experiment.** We selected several systems and databases for experimental validation and summarized the characteristics and applicable scenarios of hybrid queries based on the experimental results.

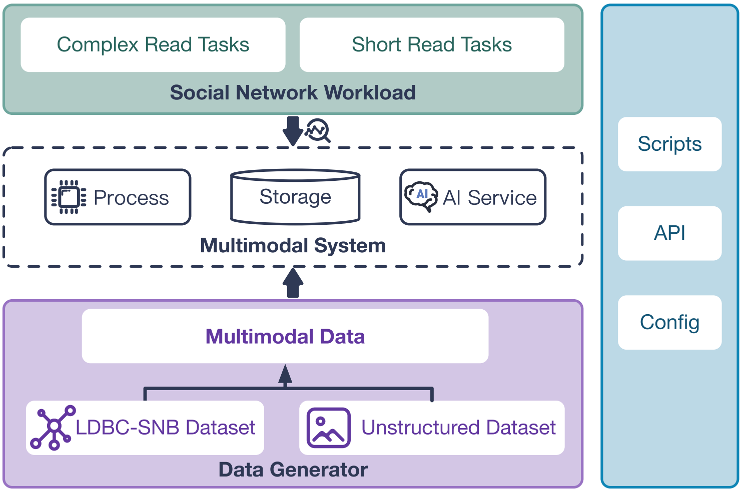

Fig. 2. Overview of MMDBench

This paper is organized as follows. In Sect. 2, we review related work in benchmark for multimodal data, highlighting the limitations of existing benchmarks. In Sect. 3, the modalities of different data are covered. In Sects. 4 and 5, details of the data generator and workload are introduced. The experimental results are shown in Sect. 6. Lastly, the conclusion is covered in Sect. 7.

2 Related Work

In the field of databases, conducting benchmark testing and performance evaluation for different types of data (structured, unstructured, and multi-model) is of great importance. Structured data refers to tabular data commonly found in traditional relational databases, while unstructured data includes data in formats such as text, images, and audio. On the other hand, multi-model data refers to database systems that can simultaneously handle multiple data models. This section provides an overview of the data models supported by these benchmark testing programs.

2.1 Single Model Benchmark Programs

Linkbench [1] is a benchmark tool developed for evaluating graph database systems. It provides a set of simple CRUD (Create, Read, Update, Delete) operations to replicate query patterns in Facebook's graph database TAO [2]. LDBC-SNB [6] is a comprehensive graph database benchmark. It evaluates systems across various social network workloads, including complex queries, updates, and data generation.

NOBENCH [5] is a benchmark testing tool developed for evaluating NoSQL database systems. It provides basic NoSQL queries for JSON documents, including selection, projection, and aggregation operations. By using NOBENCH, the performance and functionality of different NoSQL database systems in handling JSON documents can be evaluated.

2.2 Multi-model Benchmark Programs

Unibench [18,19] is a benchmark testing tool designed for evaluating multi-model database systems. It is designed to simulate various data operations and queries in multi-model data management systems. The goal of Unibench is to provide a repeatable and comparable way to assess the performance and capabilities of different multi-model database systems. Unibench supports multiple data models such as relational, document, and graph models, allowing for the simulation of complex data management and query tasks.

M2Bench [10] is a benchmark testing tool developed for evaluating multi-model database systems. It focuses on simulating multi-model queries and transaction processing in multi-model data management systems. M2Bench provides a set of complex queries and transaction scenarios, including cross-model queries, schema evolution, transaction consistency, and data consistency. By executing these queries and transactions and measuring their performance and resource consumption, the performance and scalability of multi-model database systems can be assessed.

3 Data Modalities

With the development of artificial intelligence, modern application analytics data is no longer limited to structured data, and the exploitation of unstructured data is becoming increasingly important. Many applications represent their data as a combination of multimodal data. Similarly, MMDBench represents a database in a combination of these modal data. This section will describe the data modality, focusing on the following two aspects:

Table 1. Key operations of MMDBench

Data type	Operation
Structured Graph Data	Join
	Selection
	Aggregation
	Pattern Matching
	Shortest Path
Unstructured Data	Unstructured Property Filtering
	Relationship Inference
	Similarity Matching

Data Representation. The property graph is one of the most suitable methods for describing social networks due to its convenience in implementation, and in MMDBench, graph data is chosen as the structured data representation. The property graph represents structured data using nodes and edges in a graph structure, which is formally expressed as G = (V, E, P), where G, V, E, and P represent the whole data, node collection, edge collection, and property collection, respectively. In this model, nodes represent entities or objects, and edges represent the relationships or connections between those entities. Each node and edge have properties associated with them. The graph is especially useful for representing and querying highly interconnected data, where relationships between entities are as important as the entities themselves. Nevertheless, alternative methods can also be employed to represent structured data.

On the other hand, unstructured data representation requires organizing and capturing semantic information that lacks a predefined data model. AI offers various approaches to achieve this, enabling the transformation of unstructured data into a meaningful and machine-readable format. For example, these data can be converted into vectors by AI models. Generally, the higher the dimension of the vector, the more information it can represent.

Data Manipulation. The key operations supported by MMDBench for multimodal data are summarized in Table 1. The structured graph data supports several typical operations, such as selection, join, aggregation, pattern matching, and advanced operations like finding the shortest path. Additionally, multimodal data can be treated as unstructured properties from which semantic information can be extracted and used as a filter condition for hybrid query. Moreover, these unstructured properties facilitate the exploration of latent relationships between nodes, which is called relationship inference. For instance, when we want to find topics of a post, we not only search for existing relationships but also extract semantic information from multimodal data to determine whether the post has a specific topic or not. Similarity matching in unstructured data is also a crucial operation. Generally, similarity algorithms are applied to vectors of unstructured data, including Cosine Distance, Euclidean Distance, Manhattan Distance, and others.

4 Data Generator

4.1 Constructing Data

MMDBench combines structured data with unstructured data to build multimodal datasets. For structured data, MMDBench utilizes public real-world datasets and some benchmark data generator tools. However, unstructured data is derived from realistic datasets. All the sources of the datasets are summarized in Table 2.

Table 2. Datasets of MMDBench

Data Name	Multimodal Data Type	Data Source
Social Network	Structured Graph	LDBC [6], News Category Dataset [11]
Person Faces	Image	LFW [9], IMDB-WIKI [13]
Comments	Short Text	Tweet Dataset [8]
Posts	Long Text	News Category Dataset

We employ the LDBC data generator to build linked data, which is one of the most popular data generators in the social network benchmark, and import this data into the property graph. The data generator has the capability to provide images and text, but the image file is an artificial filename rather than an existing URL or path. Moreover, the absence of sentiment tags in Messages makes it challenging to perform hybrid queries and validate the accuracy of the query results. To address these issues, we simplify the LDBC schema and replace its dictionary with some common unstructured data found in social platforms to align with our objectives. For instance, we incorporate face image files and sentiment texts, which are derived from publicly available datasets, including LFW, IMDB-WIKI, Tweet, and News Category.

As illustrated in Fig. 3, each Person node in the LDBC dataset is associated with a unique face image from either LFW or IMDB-WIKI. Each comment node contains a text and a corresponding sentiment label from the Tweet dataset. Additionally, each Post node contains a long news abstract text and a topic category from the News Category dataset. Each topic information extracted from the News Category dataset is treated as a node, facilitating relationship inference based on unstructured data. Specifically, when querying whether there is a relationship between two nodes, we not only search for existing relationships in the graph but also implicitly infer potential relationships between nodes by extracting semantic information from unstructured data.

4.2 Scaling Data from Different Modalities

MMDBench database is designed to be scalable with a specified scale factor. To accommodate different modalities of data, various expansion methods are employed. This section will provide a detailed explanation of the scaling-up methods.

Unstructured Data. When extending unstructured data, a process known as data augmentation in the field of Artificial Intelligence is employed. Several methods are used for image data enhancement, including geometric transformations, color space enhancement, kernel filters, mixed images, random erasure, feature space enhancement, generative adversarial networks, neural style transfer, and meta-learning [14]. To produce high-quality pictures, pre-trained models are a rational approach. However, generating large image datasets not only requires excellent hardware but also takes a significant amount of time, which will be addressed in our future work. Nonetheless, as for the public image dataset collected, it boasts a substantial scale, allowing us to employ the method of sampling from large-scale samples to scale up the image dataset.

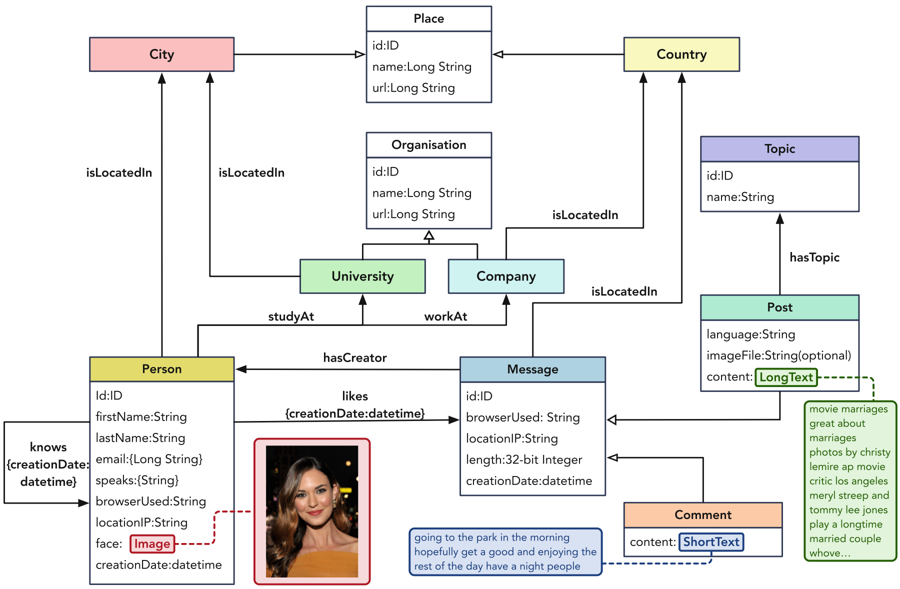

Fig. 3. The Multimodal Social Network Schema

For text data scaling up, as we did not find a text dataset of sufficient scale, we developed our data generator. EDA [17], a simple but powerful data augmentation method, consists of four important operations: synonym replacement, random insertion, random swap, and random deletion. We employ EDA in our data generator to achieve text data scaling up. The scale of expansion is limited by the scale of the original dataset. In the real world, text on social networks is often forwarded and rewritten, resulting in some similar data. This data augmentation method can partially simulate the generation of a substantial volume of data from emergency events, aligning with the characteristics of real-world data.

Structured Property Graph Data. MMDBench utilizes the extension method provided by LDBC's original data generator, enabling the generation of a social network of up to 36 million people, which sufficiently meets the requirements of MMDBench.

5 Workload

We have implemented our workload in the social network scenario, which is one of the most popular scenarios nowadays, covering a vast majority of typical operations. By default, our structured data representation is based on the property graph model. However, users have the flexibility to implement the interfaces provided by our framework to utilize other data models if needed. The tasks are divided into two parts: complex read and short read.

The complex read tasks involve multiple operations for querying multimodal data in a hybrid manner, including unstructured attribute filtering, relationship inference based on multimodal data, and more. On the other hand, the short read queries focus on the ability to process unstructured data using artificial intelligence and several simple structured data operations to fulfill typical query requirements. Each task involves data from at least two modalities, ensuring a comprehensive evaluation of the system's capabilities. A concise summary of the tasks can be found in Table 3.

Table 3. Tasks in MMDBench

	Task	Operation	Description
complex read	T1	Structured and unstructured property filtering	Given a starting person with an ID, the task is to find a friend within a 3-hop network who has specific facial features and first names. The objective is to return information about the friend's workplaces, residential cities, and study places.
	T2	Multiple unstructured property filtering	Given two individuals with their facial photos, the objective is to identify direct friendship relationships between them. If such a relationship exists, the task is to retrieve the ten most recent positive comments made by that friend.
	T3	Hybrid query with join	Search for a friend with a facial photo and geolocation information. When provided with a person's ID and a city's ID, the task is to return a friend of this person who resides in the specified city and resembles the given facial photo.
	T4	Hybrid query with aggregation	Given a person with an ID, the objective is to count the number of comments with a specific sentiment that are liked by the person's friends.
	T5	Hybrid query with Subgraph Matching	Given a person with a facial photo, the task is to query recent negative messages created by their friends or friends of friends.
	T6	Relationship inference	To find the topics of posts made by a given person with the ID, we can use both explicit and implicit relationships. Explicit relationships refer to direct connections and associations, such as topics explicitly assigned to the posts. Implicit relationships, on the other hand, involve analyzing patterns and context to identify related themes.
	T7	Hybrid query with unweighted shortest path	Given a person with an ID and a person with a facial photo, the aim is to find and return the shortest path connecting them.
short read	T8	Face recognition and pattern matching	Given a person's facial photo, the task is to retrieve their first name, last name, birthday, IP address, browser, and city of residence.
	T9	Face recognition and pattern matching	Given a person with a facial photo, the objective is to retrieve information about friends, including their ID, first name, last name, and the date they became friends
	T10	Sentiment analysis	Given a comment identified by its ID, the task is to determine its sentiment
	T11	Sentiment analysis and pattern matching	Given a person with id, the task is to retrieve the sentiment distribution of the last 10 messages they have sent

5.1 Framework of Benchmark Program

The most ideal situation would be to use a standardized query language to express tasks. However, currently, there is no unified and widely accepted multi-modal data query language. To address this issue and improve the generality of benchmark programs, we have developed a framework to assist various databases in integrating with MMDBench. Specifically, we break down all the query tasks into individual atomic operations, and users can customize the implementation of these atomic operations and data models to use MMDBench. The framework consists of models and atomic operators:

- **Model**: Node, Relationship, and PathTriple represent components of the property model.
- **Read**: nodeAt(), nodes(), and relationships() are used for reading data.

We also offer to delete and update interfaces in MMDBench. Additionally, an AI service is provided for databases that do not have integrated AI capabilities to access MMDBench. Users can utilize our default AI operators, which may demonstrate moderate performance. If users aim for higher scores, they need to embed more powerful AI operators. We provide different AI capabilities for different types of data:

- **Text**: The ability of sentiment analysis and topic extraction is provided.
- **Image**: The ability of image information extraction is supported.

5.2 Multimodal Data Schema in Social Network

The multimodal social network schema of MMDBench is illustrated in Fig. 3. The structured data model comprises social network entities, including persons, topics, geographical locations, and organizations. Unstructured data is embedded within these nodes as unstructured properties, with each person having a facial image, each comment containing a short text, and each post containing a long news text. The social network graph is scalable, and while the unstructured data can also be expanded, its scale is limited by the cardinality of the public dataset. For example, the *Person* node contains 11,000 records, the *Comment* node contains 2,581,736 records, and the *Post* node contains 1,237,554 records when the scale factor is one (SF1).

5.3 Hybrid Query in Social Network

Hybrid query refers to the need to process multiple modalities of data simultaneously within a single system [16]. To demonstrate the technical challenges, we employ task one and task six as illustrative examples. Task one involves querying information about a person's friends, and the query process is depicted in Fig. 4. Traditionally, when querying friend nodes, methods rely on filtering based on the structured attributes of individuals. However, hybrid query harnesses the power of AI to extract information from unstructured data, enabling filtering of nodes

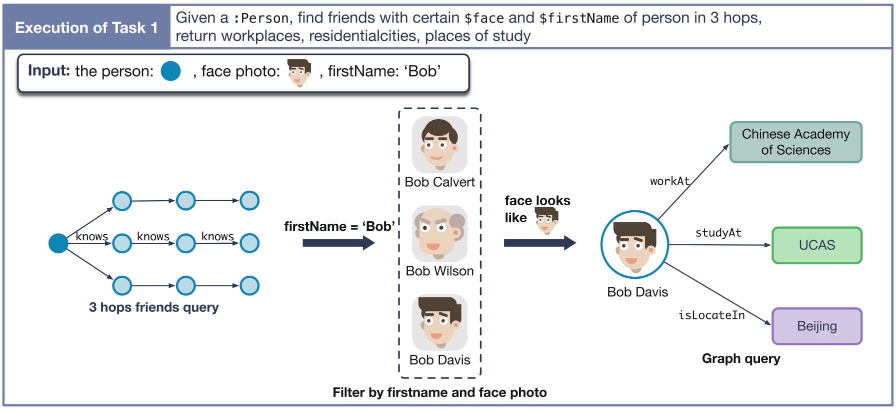

Fig. 4. Process of Hybrid Query in Task 1

based on their unstructured properties. Task 1 significantly tests the database's ability to correctly prioritize filtering conditions since processing unstructured data incurs much higher costs compared to structured data. By initially filtering $firstName$, the query process will be accelerated due to the extreme reduction of AI's search space. More complex tasks will process more multimodal data in one task, not only face photos. Task six, depicted in Fig. 5, demonstrates how to deduce relationships between nodes using unstructured data. Firstly, semantic information is extracted from news by AI operators to help uncover concealed topic types. Although the topic types inferred by AI operators might not precisely match the topic types in the schema, users can establish mapping relationships between them. Subsequently, the second sub-query conducts a direct search for $hasTopic$ relationships that may exist within the graph. Finally, the results from both queries are combined through a union operation. This task will test the ability to find all possible results using an AI-enhanced approach.

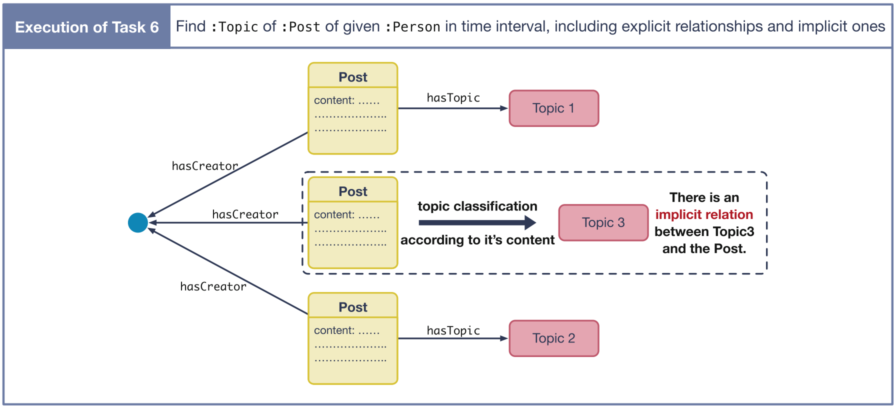

Fig. 5. Process of Hybrid Query in Task 6

6 Evaluation

To evaluate the effectiveness of MMDBench, a polyglot persistence system is developed to implement all tasks. The execution time of tasks is one of the important metrics for evaluating query performance. In the evaluation, we mainly focus on the end-to-end query time. The multimodal dataset is scaled up with the data generator to evaluate the scalability of the database systems.

6.1 Polyglot Persistence System for Evaluation

A Polyglot Persistence System refers to systems that employ multiple systems to achieve storage and query of multimodal data. In our benchmark, the polyglot persistence system provides storage ability for three types of data: structured data, images, and text. Graph data is stored in neo4j, unstructured data is stored in the file system, and AI capabilities are facilitated through the AI Web service.

To enable simultaneous access to data from multiple systems, a coordinating client is created on top of the subsystems. The client is responsible for collecting intermediate results from these subsystems and processing them to obtain the next intermediate result or the final result.

6.2 Data Generation

Experiment Setting. Our experiments are conducted on a high-performance computing cluster with 104 Intel(R) Xeon(R) Gold 6230R CPUs running at 2.10 GHz. The system has 256 GB of RAM, 4 TB of available disk space, and operated on CentOS Linux 7 (Core). The network bandwidth is 1000 Mb/s. The first five columns of Table 4 show the number of objects included in the dataset at different scales, and the last two columns show the time required for dataset generation and import.

Table 4. Characteristics of Datasets.

SF	Number					Import Time(ms)	Generator Time(ms)
	Person	Post	Comment	Likes	Has_Topic		
1	10,295	1,121,226	1,739,438	1,870,268	672,735	18,329	197,052
3	25,066	2,873,419	5,343,582	6,244,522	1,724,051	37,155	264,788
5	31,505	3,665,392	7,041,356	8,468,619	2,199,235	39,920	331,963

The data generation time consists of three stages: the time taken for generating structured data, unstructured data, and data integration. Furthermore, the dataset import time also includes the time required for index creation.

6.3 Baseline Evaluation

Figure 6 shows the execution time of all tasks on Polyglot Persistence System. Each bar represents the execution time of a task and is divided into two different colors to distinguish the time consumption of different modal data.

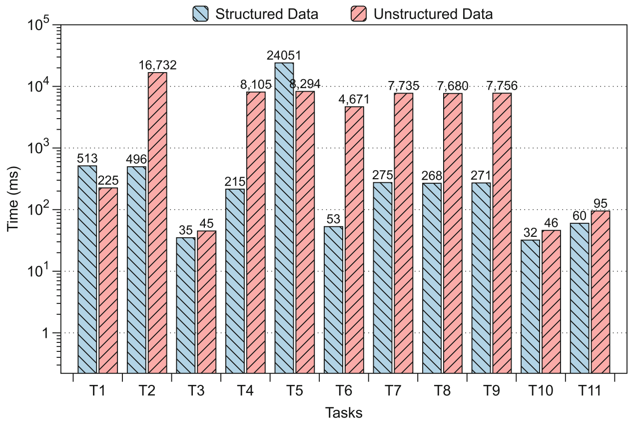

Fig. 6. Processing Time for Structured and Unstructured Data in Tasks

It is evident that the performance of structured data queries is significantly higher than that of unstructured data queries in most tasks because structured data is easier to index and filter while processing unstructured data may demand more computational resources and time. Task 5 is an exception, as it requires the execution of a highly complex subgraph matching operation.

As expected, utilizing the filtering criteria of structured data effectively reduces the search space of unstructured data, significantly reducing the query time. This has been evidenced by the results of Task 1 and Task 9. Task 1 involves querying 1 to 3 friend relationships, while Task 9 involves a much smaller number of friends. However, Task 1 smartly applies the filtering based on the structured attribute “firstName”, which eliminates a substantial portion of the data. This relieves the burden on AI information extraction and greatly accelerates the entire query process.

6.4 Latency of Polyglot Persistence

In an ideal multimodal database, all storage engines and services are localized. Within the hybrid storage system discussed in this paper, latency primarily arises from interactions with AI services. If unstructured data is stored in an external object storage system, accessing this data also introduces significant network transmission latency, and frequent calls to external services incur additional overhead. Bulk submission of requests and deployment of AI services on the nodes where the data is stored were used to eliminate latency as much as possible(The scale of data transferred is out of our control). Figure 7 illustrates

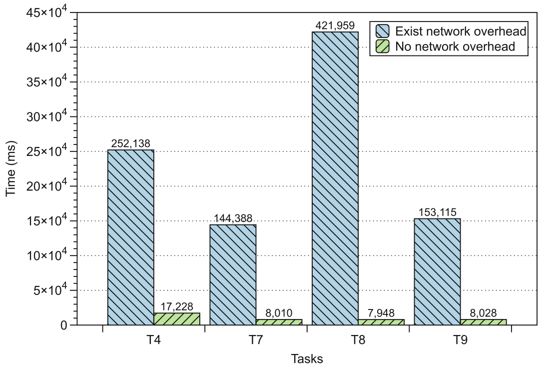

Fig. 7. Latency in Task 4, 7, 8, and 9

the execution times for tasks 4, 7, 8, and 9 after optimizing latency. It becomes evident that when transmitting substantial data volumes, the overhead from network transmission and external service calls far surpasses computational costs. Optimizing this aspect of latency can substantially enhance query acceleration. The Table 5 presents the performance improvements for all tasks after eliminating latency. In tasks 4,7,8,9, latency accounts for more than 90% of the execution time, and there is a lot of room for optimization of unstructured property filtering operation.

6.5 Scaling Data Evaluation

Figure 8 illustrates the performance of tasks on different dataset sizes. It is evident that as the dataset size increases, all task execution time exhibits a linear growth trend. Contrasting tasks 1 and 9, the advantage of prioritizing the execution of structured data filtering conditions becomes more pronounced as the dataset size increases. The elapsed time of task 2 and task 5 increases faster than the other tasks because the two tasks need to process more unstructured data as the size factor increases. The processing time for unstructured data accounts for the majority of the total runtime. Tasks 1 and 6 involve a small amount of unstructured data; thus, in comparison with other tasks, the overall runtime does not experience significant changes as the size factor increases.

6.6 Summary of Evaluation

Through the experiments above, several notable observations made in the evaluation are summarized below.

Table 5. Improvement after Eliminating Latency

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
59%	53%	12%	93%	79%	30%	94%	98%	95%	7%	10%

$Improvement\ Rate = (original\ time - improved\ time)/original\ time$

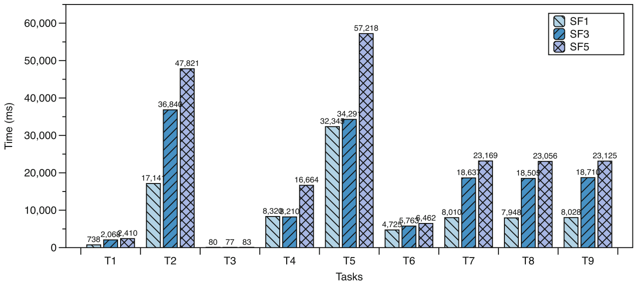

Fig. 8. Elapse time at Different Scales.

- In the Hybrid query of structured and unstructured data, executing filtering conditions on structured attributes first can effectively help accelerate the query process.
- A hybrid storage system is not an actual data management system, so communication between multiple systems and services can be optimized. Especially when dealing with large data volumes and frequent communication, the performance improvements after optimization are pretty significant.
- The query time of unstructured data is much higher than that of structured data. As the scale of data continues to increase, the more tasks touch unstructured data, the more obvious their elapse time increases.

7 Conclusion

The study presents a benchmarking program for multimodal databases in executing hybrid queries, aimed at assessing system performance when handling diverse data modalities, including structured data, and unstructured data like images and text. We propose a generator capable of producing multimodal data with different scales. To further simulate real-world demands, a multimodal social network workload is introduced to MMDBench, and some experiments are designed to demonstrate the effectiveness of the workload. We have also developed a framework that splits query into atomic operations to facilitate the integration of various types of databases into the benchmarking program. In the future, we plan to utilize AIGC to enable the generation of larger-scale datasets. Additionally, we intend to conduct experiments using real databases to obtain more precise performance evaluation reports.

References

1. Armstrong, T.G., Ponnekanti, V., Borthakur, D., Callaghan, M.: Linkbench: a database benchmark based on the Facebook social graph. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, pp. 1185–1196 (2013)
2. Bronson, N., et al.: {TAO}:{Facebook's} distributed data store for the social graph. In: 2013 USENIX Annual Technical Conference (USENIX ATC 2013), pp. 49–60 (2013)
3. Cai, Q., Wang, H., Li, Z., Liu, X.: A survey on multimodal data-driven smart healthcare systems: approaches and applications. IEEE Access **7**, 133583–133599 (2019)
4. Chandrasekaran, G., Nguyen, T.N., Hemanth D, J.: Multimodal sentimental analysis for social media applications: a comprehensive review. Wiley Interdisc. Rev.: Data Min. Knowl. Disc. **11**(5), e1415 (2021)
5. Chasseur, C., Li, Y., Patel, J.M.: Enabling JSON document stores in relational systems. In: WebDB, vol. 13, pp. 14–15 (2013)
6. Erling, O., et al.: The LDBC social network benchmark: interactive workload. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 619–630 (2015)
7. Ghazal, A., et al.: Bigbench v2: the new and improved bigbench. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), pp. 1225–1236. IEEE (2017)
8. Go, A., Bhayani, R., Huang, L.: Twitter sentiment classification using distant supervision. CS224N Proj. Rep. Stanford **1**(12), 2009 (2009)
9. Huang, G.B., Mattar, M., Berg, T., Learned-Miller, E.: Labeled faces in the wild: a database forstudying face recognition in unconstrained environments. In: Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition (2008)
10. Kim, B., Koo, K., Enkhbat, U., Kim, S., Kim, J., Moon, B.: M2bench: a database benchmark for multi-model analytic workloads. Proc. VLDB Endowment **16**(4), 747–759 (2022)
11. Misra, R.: News category dataset. arXiv preprint arXiv:2209.11429 (2022)
12. Nambiar, R.O., Poess, M.: The making of TPC-DS. In: VLDB, vol. 6, pp. 1049–1058 (2006)
13. Rothe, R., Timofte, R., Gool, L.V.: Deep expectation of real and apparent age from a single image without facial landmarks. Int. J. Comput. Vision **126**(2–4), 144–157 (2018)
14. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. J. Big Data **6**(1), 1–48 (2019)
15. Wang, Z., Li, L., Li, Q., Zeng, D.: Multimodal data enhanced representation learning for knowledge graphs. In: 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2019)
16. Wei, C., et al.: AnalyticDB-V: a hybrid analytical engine towards query fusion for structured and unstructured data. Proc. VLDB Endowment **13**(12), 3152–3165 (2020)
17. Wei, J., Zou, K.: Eda: easy data augmentation techniques for boosting performance on text classification tasks. arXiv preprint arXiv:1901.11196 (2019)
18. Zhang, C., Lu, J.: Holistic evaluation in multi-model databases benchmarking. Distrib. Parallel Databases **39**, 1–33 (2021)

19. Zhang, C., Lu, J., Xu, P., Chen, Y.: UniBench: a benchmark for multi-model database management systems. In: Nambiar, R., Poess, M. (eds.) TPCTC 2018. LNCS, vol. 11135, pp. 7–23. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11404-6_2
20. Zhao, Z., Shen, Z., Mao, A., Wang, H., Hu, C.: PandaDB: an AI-native graph database for unified managing structured and unstructured data. In: Wang, X., et al. (eds.) Database Systems for Advanced Applications, DASFAA 2023. LNCS, vol. 13946, pp. 669–673. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30678-5_53
21. Zhu, X., et al.: Multi-modal knowledge graph construction and application: a survey. IEEE Trans. Knowl. Data Eng. (2022)