Real-Time Query Method Based on Distributed Database Demand Information

Qingjia Luo

1 College of Information Engineering, Jiangmen Polytechnic, Jiangmen, Guangdong, China
2 Faculty of Data Science, City University of Macau, Xu Risheng Yin Gong Road, TaiZi, Macao, China

Correspondence should be addressed to Qingjia Luo; 631488015@qq.com

Received 8 November 2021; Revised 17 March 2022; Accepted 2 August 2022; Published 26 August 2022

Academic Editor: M. Pal

Copyright © 2022 Qingjia Luo. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a real-time query method for distributed database requirement information. First, we propose to extract the database information according to the submitted query keyword. Then, we establish a real-time model of distributed database demand information query and a real-time information classification model, using a distributed multivariate hash algorithm model and a query of real-time information. When there is a query keyword, the proposed method directly reads the data above the hash ring based on the keyword to improve query efficiency. Extensive experimental results indicate that our proposed method reduces the number of queries, query pressure, and processing time of information queries in the distributed database. Moreover, the distributed database’s real-time processing rate and amount have increased.

1. Introduction

With the progress of information technology, especially the development of data center [1], machine learning [2], and Internet of Things [3], the amount of data in various fields has shown an explosive growth trend [4]. There are considerable business values in these data [5]. However, with the increasing amount of data, the rich application scenarios also make the type of data more and more complex, and the structure more and more complicated [6]. When people are faced with massive amounts of data, how to quickly and effectively obtain the information they want from the massive amount of data has become a hot research topic in the database community [7]. In this context, data query technology is increasingly valued and widely studied [8].

In the era of big data, the traditional centralized database can no longer meet people’s requirements because they need to update the processing methods from time to time. To this end, the distributed database has emerged to meet the requirements [9]. Compared with the traditional centralized database, the distributed database mainly has the following advantages [10]. On the one hand, the distributed database has a strong expansion ability, and advantages in data storage [11]. On the other hand, it can effectively save costs [12]. The traditional centralized databases incur high storage overhead and low retrieval efficiency. In contrast, the distributed database can expand the capacity and save costs. Moreover, the distributed database is easier and faster to store, analyze, and process big data, so that its users have a significant advantage in data processing [13].

The distributed database plays an essential role in improving the data query and processing function in the big data environment [14], strengthening the research and application of distributed database technology [15], and enhancing the flexibility and extensibility of the operating system [16]. It is also a meaningful way to adapt to the network data explosion in the new era and solve data problems.

In a narrow sense, a distributed database stores the data from a centralized database on multiple storage nodes connected through the network to obtain more significant storage space and higher concurrent access. The distributed database has the advantages of flexibility, convenience, and high communication efficiency. However, the distributed database also suffers from low query efficiency because the query information can be located in different databases. In
this case, how to effectively perform the real-time performance of an information query in a distributed database has become a top priority in the field [17]. The real-time nature of an information query in a distributed database can be solved by establishing an information query pool to reduce the number of readings of a distributed database and improve the efficiency of the query data [18]. It is an effective way to solve the above problems, which has been paid much attention and some progress has been made.

Nowadays, the distributed database has become the mainstream way of data storage, which is necessary to store multisource heterogeneous and massive data. However, in using distributed spatial and temporal database storage, there is the problem of high concurrent reading and writing efficiency [19]. In particular, the reading and writing operations can be performed concurrently by the users in the distributed database. Therefore, designing a good storage architecture and improving the efficiency of real-time data queries are the research topics in distributed storage technology [20].

The standard distributed data information query method is based on Hadoop architecture, which uses the parallel execution mechanism to realize the read and storage of big data [21]. Hadoop is widely used in big data processing applications, thanks to its natural advantages in data extraction, deformation, and loading (ETL) [22]. Hadoop’s distributed architecture, which brings the big data processing engine as close to storage as possible, is relatively appropriate for batch operations like ETL, as the batch results of such operations can go directly to storage [23]. Hadoop’s MapReduce functionality enables breaking single tasks and sending the fragmentation tasks (Map) to multiple nodes before loading (Reduce) into a dataset as a single data warehouse [24].

This paper proposes a real-time query method based on distributed database demand information. Distributed multivariate hash algorithm is introduced to classify and arrange samples of known and unknown types of queries, effectively improve the real-time performance of distributed database information queries, and protect the security of the distributed database.

The main contributions of this paper are as follows:

(i) We propose a real-time method for distributed database information queries by employing the distributed multivariate hash algorithm.

(ii) We synthesize the user query information keywords, database query content keywords, and database query information content into the proposed distributed database information query method, which significantly improves the query comprehensiveness.

(iii) We conduct extensive experiments to evaluate the performance of our proposed methods. The results indicate that our proposed method significantly outperforms the traditional method.

The rest of this paper is organized as follows: Section 2 summarizes the related work. Section 3 presents the proposed method articulating the technical advances. Section 4 shows the experimental results. Finally, Section 5 concludes this paper.

2. Related Work

A distributed database is a database in which data is stored across different physical locations. Query optimization on the distributed database has been studied for decades. In this paper, we summarize the related work in recent years.

Singh proposed a multiobjective parametric query optimization (MPQO)-based approach for distributed database systems [25]. MPQO generalizes parametric optimization by catering to the multiple metrics for query optimization, builds a parameteric space of query plans, and progressively explores the multiobjective space according to user tradeoffs on query metrics. It is the first to employ genetic algorithms for query optimization in distributed database systems.

Kacem and Grissa Touzi proposed a practical approach of querying the distributed database based on the definition of relevant sites to the query while knowing the fragmentation and duplication of distributed data [26]. The proposed approach minimizes the volume of transferred data via network and consequently reduces the query execution cost. The authors have successfully implemented the proposed method on Oracle distributed database management system. This work is practical in real-world implementation and evaluation.

Moritz et al. presented Perfopticon [27], an interactive visualization and profiling tool of the distributed database providing insights into performance bottlenecks, data skew, etc. Perfopticon is evaluated by a wide range of users, such as system developers, scientists, and students. This work is the first visualization and profiling tool for the distributed database.

Chen et al. proposed MemSQL, a distributed SQL database designed to exploit memory-optimized, scale-out architecture to enable real-time transactional and analytical workloads which are fast, highly concurrent, and highly scalable [19]. The architecture of MemSQL is shown with a set of subtle techniques for query optimization. This work is a commercial solution to the distributed database. However, many of the technical details are omitted.

Qasas et al. considered the privacy issue of skyline queries in the distributed database [28]. The authors employ the secure multi-party sorting protocol and homomorphic encryption in the honest-but-curious adversary model for transforming each attribute value of the objects without changing their order on each attribute. The security is analyzed in the sense that sensitive attribute values are not disclosed. It is the first work concerning privacy preservation in querying the distributed database.

Panahi and Navimipour proposed an artificial bee colony algorithm based on genetic operators to join the query optimization problems in distributed database systems [29]. In the paper, the query optimization problem is well formulated, and the adapted genetic algorithm performs the global-local search. It is another work using the genetic
algorithm for the optimization of distributed database queries. However, there is proof that the global optima can be reached. Mohsin et al. examined how quantum-inspired ant colony algorithm, a hybrid strategy of probabilistic algorithms, can be devised to improve the cost of query joins in the distributed database [30]. Quantum computing can diversify and expand, thus covering large query search spaces. The technique enables the selection of the best trails, speeds up convergence, and helps avoid falling into a local optimum. This work is the first to consider quantum computing for query optimization for the distributed database. However, the real-world experiments and theoretical analysis are missing in the paper. Another similar work is done by Mohsin et al. [31].

Onan et al. conducted an empirical analysis of keyword extraction methods for text classification [32]. Then, a novel text sentiment classification model is developed based on the fuzzy-rough nearest neighbor method [33]. Furthermore, a classifier based on the multiobjective differential evolution algorithm is proposed for text sentiment classification [34]. In addition, a hybrid ensemble learning model is designed for sentiment classification, leveraging the multiobjective evolutionary algorithm [35]. Finally, the sentiments on product reviews are analyzed leveraging weighted word embeddings and deep neural networks [36].

To summarize, the existing works either employ genetic algorithms and quantum computing for query optimization, or focus on the distributed database profiling or privacy issues. The traditional probabilistic methods are not adequately considered in the literature.

3. Proposed Method

The purpose of this paper is to provide a real-time query method and system based on the distributed database demand information to effectively improve the efficiency of the distributed database query and improve the data query timeliness.

3.1. Distributed Multivariate Hash Algorithm. The distributed hash algorithm is a classification algorithm based on the Bayesian theorem, which is formulated as follows:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}.$$  \hspace{1cm} (1)

In the equation, $P(A)$ is called the prior probability, $P(B|A)$ is called the conditional probability, and $P(A|B)$ is called the posterior probability. Furthermore, $P(B)$ can be seen as the probability of $B$ occurrence given all the conditions. The full probability formula is as follows:

$$P(B) = \sum_{i=1}^{N} P(B|A_i) \cdot P(A_i).$$  \hspace{1cm} (2)

The naive Bayesian classifier is transformed according to the Bayesian theorem, if there is a dataset $D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})\}$. There are $N$ data in total, in which each $x^{(i)}$ is a vector with $n$ features while each $y^{(i)}$ is the category corresponding the $x^{(i)}$, assuming a total of $K$ categories. The problem is that how to judge the category of a given new $x$.

Using Bayesian theorem, we have

$$P(C_k|x) = \frac{P(x|C_k) \cdot P(C_k)}{P(x)}.$$  \hspace{1cm} (3)

Here, $x$ is the input vector with $n$ features, ranging from $x_1$ to $x_n$. The $n$ features are assumed to be independent with each other. Thus, we have

$$P(x|C_k) = P(x_1, \ldots, x_n | C_k) = \prod_{j=1}^{n} P(x_i | C_k).$$  \hspace{1cm} (4)

Substituting (4) into (3) and expanding (3) in $P(x)$ by the full probability formula yields

$$P(C_k|x) = \frac{P(C_k) \cdot \prod_{i=1}^{n} P(x_i | C_k)}{\sum_{j=1}^{K} P(C_j) \prod_{i=1}^{n} P(x_i | C_j)}.$$  \hspace{1cm} (5)

(5) is the model of the distributed hash algorithm. Distributed hash algorithms have outstanding advantages in improving the efficiency of data queries.

3.2. Algorithm Procedure. A real-time query method based on distributed database requirement information, the identification method includes: according to the submitted query keyword, extract the database information; based on the information queried in a distributed database, establish a real-time model of distributed database demand information query; query the real-time model based on the information, build a real-time information classification model; query the real-time model and the real-time information classification model according to the information, build a distributed multivariate hash algorithm model; according to the distributed multivariate hash algorithm model and the real-time query information, when the query keywords enter the system, read the data above the hash ring directly based on the keyword, to improve the efficiency of reading the distributed database.

The input of this algorithm is the query of the information, while the output is the hash value of the query value. The algorithm runs in three steps as follows:

(i) Step 1. We establish a real-time model of the distributed database requirement information query based on the information query in the distributed database. According to the keywords input by the user and the information queried in the distributed database, store the keywords and the query information keywords and other contents, prepare the data for the following information query, form an information query pool, and establish a real-time model of the distributed database demand information query.
(ii) Step 2. The distributed database demand information query real-time model includes storing user input keywords, keyword information query from the distributed database, and database query information content. The class diagram thumbnails for this model are shown in Figure 1. Users through the distributed database demand information query real-time model interface class of related information storage including user query information keyword, database query content keywords, database query information content, user query information keywords, database query content keywords, database query information content on information keywords, content keyword, the content of the three entity classes.

(iii) Step 3. According to the real-time information classification model, we devise the method as follows. According to the information queried from the distributed database, obtain the keywords of the information, judge whether the type of information can be queried twice, and obtain the first judgment result. If the first judgment result is a secondary query, the information keyword may be placed into the information query pool. If the first judgment is not a secondary query, the information keyword is removed from the information query pool; if the information keyword reappears, it can be allowed to query the distributed database. The specific check processing flow chart is shown in Figure 2.

3.3. System Design. Based on the algorithm, a real-time query system of distributed database demand information in the power system, which has five modules, is shown in Figure 3. The functionalities of the modules are as follows:

(i) Extraction of the database information. The purpose of this module is to extract the distributed database information.

(ii) Real-time query of distributed database demand information. This module is used to store keywords such as the information based on the information query in the distributed database to form a real-time model of distributed database demand information query.

(iii) Real-time information classification. This module is used to query real-time information to classify real-time information and form a real-time information classification model.

(iv) Distributed multivariate hash algorithm. This module is used to establish distributed multi-time hash algorithm model based on information query and real-time information classification model.

(v) Establishment of an information query channel. This module is to read the data above the distributed multiple hash algorithm model and the real-time query information from the query key directly when the query keyword enters the system, thus improving the efficiency of reading the distributed database.

The distributed multivariate hash algorithm module is implemented as follows. According to the information query real-time model and the information query classification model, judge whether the queried data can be queried twice and obtain the first judgment result. If the information query type of the first judgment result is: the second query can be performed, and arrange the data from the distributed database based on the key, and when encountering the data query by the second key, arrange the data clockwise and record the hash value of the key as 1, forming the information query pool (see Figure 4 for details). If the information query type of the first judgment result is: no secondary query can be made, the information is deleted from the information query pool, and if the information keyword appears again, it can be allowed to query the distributed database.

The establishment information query channel, characterized in the establishment information query channel module, is described as follows. According to the distributed multivariate hash algorithm module, the data is compared from the information query pool and judged to obtain the first result. If the first result is to find the data in the information query pool, it can communicate with the real-time information query model. If the first result is unable to find the keyword in the information query pool, it can read the distributed database again.

This paper studies the real-time query method and system of the distributed database demand information. The identification method includes extracting database information according to the submitted query keyword, establishing a distributed database demand information query real-time model according to the information query real-time model, constructing the real-time information classification model according to the information query and the real-time query information, and directly entering the system to improve the efficiency of reading the distributed database. The method or system of this paper can improve the real-time of distributed database data query, improve the efficiency of data query, reduce the number of queries of distributed database, reduce the pressure of database query, shorten the processing time of information query in the distributed database, and increase the real-time processing rate and processing volume of distributed database.
We improve the existing distributed database architecture, and the processing mechanism of information query pool to abnormal information query pool is added in this paper, which can effectively prevent the data security problems caused by information queries. Through experimental analysis, the identification and prevention method for the distributed database information query attack can effectively improve the data query efficiency and bring good results on the efficient operation of the distributed database.

4. Experimental Results

An experiment requires the need to demonstrate the proposed real-time query method for demand analysis in distributed data. Build a real-time query experimental simulation platform based on demand analysis in distributed data under the ns-2 environment. The simulation sets the common data query type and applies the subject method. Under the same network scenario, the same ten sets of different keywords were recorded, and the two types used the throughput, delay, and packet loss rate for comparison.

4.1. Distributed Real-Time Query Experiment of Database Requirement Information

Distributed real-time query experiment of database requirement information was performed using the distributed hash algorithm and stationary load algorithm. Different query keywords are input to compare the throughput (T)%, delay (D)%, and packet loss rate (P)% of two different algorithms, and measure the comprehensive effectiveness of real-time query of distributed database demand information using the results of two different algorithms. Comparison results are presented in Table 1.

It can be observed from Table 1 that the comprehensive effectiveness of the information query using the distributed hash algorithm is much better than the information query by using the stationary load algorithm. The reason is that when using distributed hash information query, the observation sequence of each protocol layer in the mobile network can show the network behavior of nodes comprehensively, thus thoroughly ensuring the comprehensive effectiveness of the distributed hash algorithm for information query.

4.2. Real-Time Query Experiment of Distributed Database Requirement Information

The real-time query experiment of distributed database requirement information was conducted using the distributed hash algorithm and trust distance-based algorithm. The detection efficiency of two different algorithms is compared in different mobile scenarios, and the comparison results are shown in Figure 5.

As shown from Figure 5, the efficiency of information query using the distributed hash algorithm is much higher than using trust distance algorithm, which effectively ensures the efficiency of information query by the distributed hash algorithm for information query.
experiments demonstrate that the real-time query method based on distributed database demand information has higher query efficiency.

5. Conclusion

In this paper, we propose a distributed hash algorithm that arranges the data over the hash ring and assigns the hash values, thus improving the efficiency of the data query. This topic proposes a real-time query method based on distributed database demand information and introduces a distributed multivariate algorithm to classify and arrange samples of known and unknown types of queries, complete real-time information query, improve the real-time performance of distributed database information query, and protect the security of the distributed database. In the future, we will extend the applied algorithms to real-world scenarios and conduct extensive performance evaluations.

Data Availability

Some or all data, models, or code generated or used during the study are available from the corresponding author by request.

Conflicts of Interest

The authors have no conflicts of interest to declare.

Acknowledgments

The research was supported by Jiangmen Fourth Batch of Supporting Science and Technology Development Fund Project in 2019 (Jiangke No. 256, 2019), University Youth Innovation Talents Project of Guangdong Department of Education (2019GKQNCGX111); Guangdong Science and Technology Innovation Strategy Fund (“Climbing Project”) Project in 2020; Guangdong Science and Technology Innovation Strategy Fund (“Climbing Project” in 2021) Project.

References


