

DATA READING METHOD AND APPARATUS

The present application claims priority to Chinese Patent Application No. 202010568737.9, titled “DATA READING METHOD AND APPARATUS” and filed on June 19, 2020, which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

Embodiments of this specification relate to the field of data processing technologies, and in particular, to a data reading method. One or more embodiments of this specification simultaneously relate to a data reading apparatus, a computing device, and a computer-readable storage medium.

BACKGROUND

At present, most of the distributed databases on the market do not support consistent read of a standby database, and the so-called consistent read of the standby database means that results read from the standby database and the primary database at the same time are the same. Generally, due to the delay in the primary and standby (that is, the primary database and the standby database) replication, the data status on the standby database lags behind that of the primary database, which may result in different primary and standby reading results. That is, the data in the standby database is asynchronous with that in the primary database, causing the data read from the standby database to be inconsistent with the data in the primary database.

Therefore, it is necessary to provide a data reading method that can achieve the same primary and standby reading results, to ensure consistent read of primary and standby databases.

SUMMARY

In view of this, embodiments of this specification provide a data reading method. One or more embodiments of this specification simultaneously relate to a data reading apparatus, a computing device, and a computer-readable storage medium, to resolve the technical defects existing in the prior art.

According to a first aspect of the embodiments of this specification, a data reading method is provided, including:

receiving a data reading transaction request;

acquiring a snapshot timestamp based on the data reading transaction request; and

sending, in a case that the snapshot timestamp is less than a maximum commit timestamp for a write transaction of first databases, the data reading transaction request carrying the snapshot timestamp to the first databases for data reading.

Optionally, that the snapshot timestamp is less than a maximum commit timestamp for a write transaction of first databases includes:

the snapshot timestamp being less than a maximum commit timestamp for a write transaction of a second database to the first databases, where the first database is a backup database of the second database.

Optionally, after the acquiring a snapshot timestamp based on the data reading transaction request, the method further includes:

reacquiring, in a case that the snapshot timestamp is greater than the maximum commit timestamp for the write transaction of the first databases, a maximum commit timestamp for the write transaction of the first databases after waiting for a preset duration.

Optionally, after the reacquiring a maximum commit timestamp for the write transaction of the first databases after waiting for a preset duration, the method further includes:

acquiring a log sequence number (LSN) of a second database in a case that the snapshot timestamp is less than the maximum commit timestamp for the write transaction of the first databases;

sending the data reading transaction request carrying the LSN of the second database to the first database, and acquiring an LSN of the first database; and

reading data in the first database based on the data reading transaction request and the LSN of the first database in a case that it is determined that the LSN of the first database matches the LSN of the second database.

Optionally, the acquiring an LSN of a second database includes:

acquiring an LSN of the last complete database backup of the second database; and

the acquiring an LSN of the first database correspondingly includes:

acquiring an LSN of the last complete database backup of the first database.

Optionally, after the receiving a data reading transaction request, the method further includes:

determining at least one corresponding second database based on the data reading transaction request.

Optionally, the acquiring an LSN of a second database includes:

acquiring an LSN of each second database; and

the sending the data reading transaction request carrying the LSN of the second database

to the first database correspondingly includes:

sending the data reading transaction request carrying the LSN of the second database to the first database corresponding to each second database.

Optionally, the data reading method is applicable to a distributed database.

According to a second aspect of the embodiments of this specification, a data reading apparatus is provided, including:

a request receiving module, configured to receive a data reading transaction request;

a timestamp acquisition module, configured to acquire a snapshot timestamp based on the data reading transaction request; and

a data reading module, configured to send, in a case that the snapshot timestamp is less than a maximum commit timestamp for a write transaction of first databases, the data reading transaction request carrying the snapshot timestamp to the first databases for data reading.

Optionally, the data reading module is further configured to that:

the snapshot timestamp is less than a maximum commit timestamp for a write transaction of a second database to the first databases, where the first database is a backup database of the second database.

Optionally, the apparatus further includes:

a retry module, configured to reacquire, in a case that the snapshot timestamp is greater than the maximum commit timestamp for the write transaction of the first databases, a maximum commit timestamp for the write transaction of the first databases after waiting for a preset duration.

Optionally, the apparatus further includes:

a first sequence number acquisition module, configured to acquire an LSN of a second database in a case that the snapshot timestamp is less than the maximum commit timestamp for the write transaction of the first databases;

a second sequence number acquisition module, configured to send the data reading transaction request carrying the LSN of the second database to the first database, and acquire an LSN of the first database; and

a sequence number matching module, configured to read data in the first database based on the data reading transaction request and the LSN of the first database in a case that it is determined that the LSN of the first database matches the LSN of the second database.

Optionally, the first sequence number acquisition module is further configured to:

acquire an LSN of the last complete database backup of the second database; and

the second sequence number acquisition module is correspondingly further configured to:

acquire an LSN of the last complete database backup of the first database.

Optionally, the apparatus further includes:

a database determining module, configured to determine at least one corresponding second database based on the data reading transaction request.

Optionally, the first sequence number acquisition module is further configured to:

acquire an LSN of each second database; and

the second sequence number acquisition module is correspondingly further configured to: send the data reading transaction request carrying the LSN of the second database to the first database corresponding to each second database.

Optionally, the data reading method is applicable to a distributed database.

According to a third aspect of the embodiments of this specification, a computing device is provided, including:

a memory and a processor, where

the memory is configured to store computer-executable instructions, the processor is configured to execute the computer-executable instructions, and the processor, when executing the computer-executable instructions, implements the steps of the data reading method.

According to a fourth aspect of the embodiments of this specification, a computer-readable storage medium is provided, storing computer-executable instructions, the instructions, when executed by a processor, implementing the steps of the data reading method.

Embodiments of this specification provide a data reading method and apparatus, where the data reading method includes: receiving a data reading transaction request; acquiring a snapshot timestamp based on the data reading transaction request; and sending, in a case that the snapshot timestamp is less than a maximum commit timestamp for a write transaction of first databases, the data reading transaction request carrying the snapshot timestamp to the first databases for data reading. In the data reading method, acquisition for a snapshot timestamp is performed once, and consistency between data of distributed standby databases and data of the corresponding primary database read under the same data reading transaction request can be ensured based on the snapshot timestamp, which greatly improves the user experience for data reading and querying in the distributed standby databases.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a schematic diagram of a data reading apparatus according to an embodiment of this specification;

FIG. 2 is a flowchart of a data reading method according to an embodiment of this specification;

FIG. 3 is a flowchart of a processing process of a first data reading method according to an embodiment of this specification;

FIG. 4 is a flowchart of a processing process of a second data reading method according to an embodiment of this specification;

FIG. 5 is a schematic structural diagram of a data reading apparatus according to an embodiment of this specification; and

FIG. 6 is a structural block diagram of a computing device according to an embodiment of this specification.

DETAILED DESCRIPTION

In the following description, many specific details are described for thorough understanding of this specification. However, this specification may alternatively be implemented in other manners than those described herein, and a person skilled in the art may make similar modifications without departing from the content of this specification. Therefore, this specification is not limited to the implementations disclosed below.

The terms used in one or more embodiments of this specification are merely used to describe the specific embodiments but are not intended to limit one or more embodiments of this specification. The “a” and “the” in a singular form used in one or more embodiments of this specification and the appended claims are also intended to include a plural form, unless other meanings are clearly indicated in the context. It should further be understood that the term “and/or” used in one or more embodiments of this specification indicates and includes any or all possible combinations of one or more associated listed items.

It should be understood that although the terms such as first and second may be used to describe various information in one or more embodiments of this specification, but such information should not be limited to these terms. These terms are merely used to distinguish between information of the same type. For example, without departing from the scope of one or more embodiments of this specification, the first may also be referred to as the second. Similarly, the second may also be referred to as the first. According to the context, the word “if” used herein may be interpreted as “during” or “when” or “in response to determining.”

First, the terminology involved in one or more embodiments of this specification is explained.

Transaction: It is an execution unit including a plurality of SQL statements, and having four characteristics of atomicity, consistency, isolation, and durability.

MVCC: It is fully referred to as multi-version concurrency control, and is a common concurrency control mechanism for databases, where each piece of data is stored with a plurality of versions, and for a read request of a transaction, a version visible to it is always read, which does not block a write request.

Consistent read: In MVCC, transaction-visible data can be regarded as a "snapshot" or "view" of a database at a certain time point. The consistency mentioned in this specification refers to the following issue: within the same transaction, a snapshot read from a standby database is necessarily the same as that read from a primary database.

Timestamp: It is usually a 64-bit integer, and is used for comparing and determining a temporal relationship between events.

DN: It is fully referred to as data node, and an entire database is usually distributed on a plurality of DNs in the form of sharding.

CN: It is fully referred to as coordinator node, and is responsible for coordinating and executing transactions on one or more DNs.

Transaction ID: It is referred to as `trx_id` for short, and is the unique ID of each transaction.

Snapshot timestamp: It is also referred to as `snapshot_seq`, which is a snapshot timestamp of a transaction, where visibility of a data version is determined according to the timestamp.

Commit timestamp: It is also referred to as `commit_seq`, which is a commit timestamp of a transaction.

GTS: It is fully referred to as global timestamp service, and is responsible for issuing timestamps. All other nodes in the data reading apparatus in the embodiments of this specification request timestamps from the GTS.

In this specification, a data reading method is provided. One or more embodiments of this specification simultaneously relate to a data reading apparatus, a computing device, and a computer-readable storage medium, which are described in detail one by one in the following embodiments.

FIG. 1 is a schematic diagram of a data reading apparatus according to an embodiment of this specification, including: a client 102, a transaction coordination module 104 (CN), a timestamp distribution module 106 (GTS), a primary database 108, and standby databases 110.

During specific implementation, the CN receives a read transaction request, for example, a transaction request for querying for some data, sent by the client 102, and the read transaction request in this embodiment is consistent with the content to be expressed in the data reading

transaction request in the following embodiments, where the read transaction includes one or more SQL statements, and the specific data to be read or queried for under the read transaction request can be specified through the SQL statement; and

the CN simultaneously sends a request for acquiring the corresponding snapshot timestamp for the read transaction request to the GTS, and after receiving a snapshot timestamp of the read transaction returned by the GTS, the CN sends the read transaction request carrying the snapshot timestamp to each standby database 110 corresponding to the read transaction request, and acquires the maximum commit timestamps for a write transaction of all the standby databases 110, where the commit timestamp is the maximum commit timestamp for the primary database 108 to execute the write transaction to the corresponding standby databases 110, and the commit timestamp is incremented according to write transactions executed by the primary database 108. Specifically, when the primary database 108 executes a write transaction to the standby databases 110, scheduling is also performed through the CN, and each time a write transaction is executed, the CN also sends a request for acquiring a timestamp to the GTS, acquires the corresponding commit timestamp for the write transaction, and the like.

In a case that it is determined that the corresponding snapshot timestamp acquired based on the read transaction request is less than the maximum commit timestamp for the write transaction of all the standby databases 110, it can be determined that the data of the read transaction request in the primary database 108 has already been all written to the standby databases 110 before the snapshot timestamp, then the data read from the primary database 108 and the standby databases 110 based on the read transaction request of the snapshot timestamp can be kept consistent, and it is unnecessary to introduce other additional network requests or optimize network resources. In addition, by acquiring a snapshot timestamp of a read transaction once only, the data acquired from the standby databases 110 based on the read transaction request of the snapshot timestamp and the data acquired from the primary database 108 based on the read transaction request of the snapshot timestamp can be kept consistent, thereby improving the user experience.

FIG. 2 is a flowchart of a data reading method according to an embodiment of this specification, which includes the following steps:

Step 202: Receive a data reading transaction request.

Specifically, the data reading method provided in this embodiment of this specification is applied to a distributed database, and is specifically applied to a MVCC mechanism, and a specific execution object of the data reading method is the CN in the foregoing embodiments.

The data reading transaction request may be understood as a request for a data reading

transaction to be executed, and the data reading transaction includes one, two, or more SQL statements, such as SELECTMAX(sal), MIN(age), AVG(sal), and SUM(sal) FROM emp. In practical applications, the data reading transaction request is generally sent by a client.

Step 204: Acquire a snapshot timestamp based on the data reading transaction request.

Specifically, after receiving the data reading transaction request sent by the client, the CN sends, based on the data reading transaction request, a request for acquiring a snapshot timestamp for the data reading transaction to the GTS, and the GTS generates a snapshot timestamp for the data reading transaction, and returns the snapshot timestamp to the CN. For ease of understanding, the data reading method is described in detail below by using an example in which the snapshot timestamp is a snapshot_seq.

During specific implementation, when receiving the snapshot timestamp of the data reading transaction returned by the GTS, the CN acquires the maximum commit timestamp for a write transaction of first databases. In practical applications, the first database may be understood as a standby database, that is, a backup database of the primary database; and the maximum commit timestamp for the write transaction of the first databases may be understood as the maximum commit timestamp for the write transaction for writing data to the standby databases by the primary database.

In practical applications, data reading and data writing do not affect each other, and data writing is not affected when data reading is performed. For example, during execution of data querying, the data in the primary database is backed up to the standby database synchronously, and the data backup will not be terminated due to the execution of the data querying transaction. Each time the data in the primary database is written to the standby database, it is equivalent to executing a write transaction once, and when the CN schedules the data in the primary database and writes the data to the standby database, a commit timestamp for the write transaction is generated in the GTS.

In addition, because the data reading method is applied to a distributed database, there are a plurality of sharding primary databases and a plurality of corresponding sharding standby databases. When the CN backs up the data of the primary databases to the standby databases, it backs up the data in the plurality of sharding primary databases to the plurality of corresponding sharding standby databases. In this case, according to the different amount of data to be backed up, the time to complete the write transaction will also be different. Therefore, a plurality of commit timestamps are generated, and in this embodiment of this specification, the acquired maximum commit timestamp for the write transaction of the first databases is the maximum commit timestamp for the write transaction of all the standby databases.

Step 206: Send, in a case that the snapshot timestamp is less than a maximum commit timestamp for a write transaction of first databases, the data reading transaction request carrying the snapshot timestamp to the first databases for data reading.

In a case that the snapshot timestamp is less than the maximum commit timestamp for the write transaction of the standby databases, the data reading transaction request carrying the snapshot timestamp is sent to the first databases for data reading, and data read by the first databases based on the data reading transaction request carrying the snapshot timestamp is received.

Specifically, that the snapshot timestamp is less than a maximum commit timestamp for a write transaction of first databases includes:

the snapshot timestamp being less than a maximum commit timestamp for a write transaction of a second database to the first databases, where the first database is a backup database of the second database.

The second database may be understood as the primary database in the foregoing embodiments.

In this embodiment of this specification, according to the data reading method, the snapshot timestamp is compared with the maximum commit timestamp for the write transaction of the standby databases, and consistency between data acquired from the primary database and the standby databases under the same data reading transaction request can be realized based on the snapshot timestamp, which improves the user experience.

For example, if the snapshot timestamp of the data reading transaction request sent by the client is ****12:05:02, and the maximum commit timestamp for the write transaction of the first databases is ****12:05:03, it can be determined that when the data reading transaction request from the client is received, all the data in the primary database before ****12:05:02 have been backed up to the standby databases, and in this case, data with a timestamp before ****12:05:02 acquired from the standby databases and data with a timestamp before ****12:05:02 acquired from the primary database according to the data reading transaction request are kept consistent. That is, if the completion time of writing data to the standby databases by the primary database is earlier than the time for executing the data reading transaction request by the standby databases, data read from the primary database and the standby databases under the data reading transaction request can be consistent.

The data reading method provided in this embodiment of this specification includes: receiving a data reading transaction request; acquiring a snapshot timestamp based on the data reading transaction request; and sending, in a case that the snapshot timestamp is less than a

maximum commit timestamp for a write transaction of first databases, the data reading transaction request to the first databases for data reading, and receiving data read by the first databases based on the data reading transaction request. In the data reading method, acquisition for a snapshot timestamp is performed once, and consistency between data of distributed standby databases and data of the corresponding primary database read under the same data reading transaction request can be ensured based on the snapshot timestamp, which improves the user experience for data reading and querying in the distributed standby databases.

In another embodiment of this specification, after the acquiring a snapshot timestamp based on the data reading transaction request, the method further includes:

reacquiring, in a case that the snapshot timestamp is greater than the maximum commit timestamp for the write transaction of the first databases, a maximum commit timestamp for the write transaction of the first databases after waiting for a preset duration.

The preset duration may be set according to actual applications, and is not limited herein. For example, it is set to 1ms or 2ms.

Specifically, to avoid poor experience caused to a user by directly terminating the data reading transaction request when the snapshot timestamp is greater than the maximum commit timestamp for the write transaction of the first databases, a waiting time acceptable to a distribution database of the database is set in a case that the snapshot timestamp is greater than the commit timestamp, and after a preset duration is waited for, a maximum commit timestamp for the write transaction of the first databases is reacquired for determining. The consistency between the read data of the primary database and the standby databases is ensured through determining based on snapshot timestamps twice without affecting the processing of the data reading transaction request. In a case that it is determined that the data of the primary database and the standby databases are inconsistent based on the snapshot timestamp for the first time, a second opportunity for determining is strived for the data reading transaction request of the client, and directly ending of the data reading transaction request of the client is avoided, so that the user needs to submit the data reading transaction request again for a plurality of times in this case, to increase the user's data reading time.

During specific implementation, after the maximum commit timestamp for the write transaction of the first databases is reacquired after the preset duration is waited for, whether the snapshot timestamp is smaller than the maximum commit timestamp for the write transaction of the first databases is determined again. If the snapshot timestamp is smaller than the maximum commit timestamp, the above process continues to be performed, and the snapshot timestamp is not smaller than the maximum commit timestamp, to ensure the normal

progress of the data reading transaction request, data acquired from the standby databases by the client under the same data reading transaction request and data acquired from the primary database under the data reading transaction request are ensure to be consistent in an LSN manner. The specific implementation is as follows:

- after the reacquiring a maximum commit timestamp for the write transaction of the first databases after waiting for a preset duration, the method further includes:

- acquiring an LSN of a second database in a case that the snapshot timestamp is less than the maximum commit timestamp for the write transaction of the first databases;

- sending the data reading transaction request carrying the LSN of the second database to the first database, and acquiring an LSN of the first database; and

- reading data in the first database based on the data reading transaction request and the LSN of the first database in a case that it is determined that the LSN of the first database matches the LSN of the second database.

The LSN is fully referred to as log sequence number in English, and is the unique number of each record in a transaction log. Every time write transaction commit is performed by the data in the primary database, a record is generated in the log. A corresponding LSN is generated for each record, and the LSN is simultaneously stored in the file and the corresponding data table.

Specifically, in a case that a data reading transaction request is received, and it is determined that the snapshot timestamp of the data reading transaction request is greater than the maximum commit timestamp for the write transaction of the standby databases, an LSN of the primary database is acquired, and then the data reading transaction request carrying the LSN of the primary database is sent to the standby databases corresponding to the primary database respectively. Subsequently, an LSN of the standby database is obtained, and in a case that the LSN of the primary database matches the LSN of the standby database, data is read from the first database based on the data reading transaction request and the LSN of the first database.

In practical applications, the data in the primary database is backed up to the standby database synchronously, so that LSNs of the data are also be backed up to the standby database synchronously. If there is an LSN in the standby database the same as that of data in the primary database, it shows that the data with the LSN in the primary database has been all backed up to the standby database, and the data of the primary database and the standby database are consistent. In this case, data consistent with that in the primary database can be read from the standby database based on the data reading transaction request and the LSN of the standby

database. For example, in this case, there is data with LSNs 1-300 in the primary database, and there is also data with LSNs 1-300 in the standby database, then the data read from the primary database and the standby database under the data reading transaction request is consistent.

During specific implementation, the acquiring an LSN of a second database includes:

acquiring an LSN of the last complete database backup of the second database; and

the acquiring an LSN of the first database correspondingly includes:

acquiring an LSN of the last complete database backup of the first database.

Specifically, acquisition of the LSN of the last complete database backup of the second database and acquisition of the LSN of the last complete database backup of the first database are both for ensuring that the data in the standby database is the data of the last backup of the primary database, to ensure the real-time performance of the data.

However, because the data reading method in this embodiment is applied to a distributed database, one data reading transaction request may correspond to a plurality of primary databases and a plurality of standby databases. Therefore, after a data reading transaction request sent by the client is received, the data reading transaction request is parsed first, and the corresponding primary database is determined for subsequent data processing in the corresponding primary database. The specific implementation is as follows:

after the receiving a data reading transaction request, the method further includes:

determining at least one corresponding second database based on the data reading transaction request.

During specific implementation, the acquiring an LSN of a second database includes:

acquiring an LSN of each second database; and

the sending the data reading transaction request carrying the LSN of the second database to the first database correspondingly includes:

sending the data reading transaction request carrying the LSN of the second database to the first database corresponding to each second database.

Specifically, after at least one corresponding primary database is determined based on the data reading transaction request, an LSN of each primary database is acquired, and then the LSN of each primary database is bound to the data reading transaction request respectively. Subsequently, the data reading transaction request bound to the LSN of each primary database is sent to the standby database corresponding to each primary database, so that the standby database can process the data reading transaction request based on the LSN, to ensure that the data read from the standby database under the data reading transaction request and the data in the primary database are consistent, thereby avoiding data omission.

In practical applications, to ensure the running speed of the database, data reading is often performed in the standby database, and data writing is performed in the primary database. After a data reading transaction request from the client is received, data reading is performed in the standby database. In this case, it is necessary to ensure that the data read from the primary database the data read from the standby database under the data reading transaction request are consistent, and only in this case, the data read from the standby database under the data reading transaction request can be used normally. Therefore, data read from the standby database under the same data reading transaction request and data read from the primary database under the data reading transaction request are ensured to be consistent through determining on the snapshot timestamp or in an LSN manner, thereby enhancing the user experience.

The data reading method provided in this embodiment of this specification is cleverly designed and utilizes the monotonicity of the GTS timestamp to establish a causal relationship on the global shards, and by acquiring a timestamp once only, the primary and standby consistency determining on all shards can be satisfied, which reflects good network optimization and saves network resources.

The following performs correctness verification on use of the following content in the data reading method: in a case that the snapshot timestamp of the data reading transaction is greater than the maximum commit timestamp for the write transaction of the first databases, the foregoing steps are retried after a preset duration is waited for and the data reading can be satisfied:

It is assumed that a write transaction T1, that is, a transaction that has been written to the primary database before, is synchronized to the standby database through a synchronization link, which may alternatively be regarded as data to be read, and there is a read transaction S. Then in practical applications, T1 shall be visible to S, that is, in the standby database, either T1 is visible to S, or S blocks and waits for the visibility which is determined according to a timestamp.

If $T1.commit_seq < S.snapshot_seq$, $S.snapshot_seq < max_commit_seq$, that is, there is a commit event meeting $T2.commit_seq > S.snapshot_seq$, the specific commit timestamp of T1 is less than the snapshot timestamp of S, the snapshot timestamp of S is less than the maximum commit timestamp for the write transaction of the standby databases, and the max_commit_seq is obtained by the write transaction T2,

it can be learned from $T1.commit_seq < S.snapshot_seq$ that, $T1.commit_seq < T2.commit_seq$, so that $T1.GetCommitTS \rightarrow T2.GetCommitTS$ (where $\rightarrow$ means happens-before). Specifically, because the commit timestamp of T1 is less than the snapshot timestamp

of S, and the snapshot timestamp of S is smaller than the commit timestamp of T2, it can be considered that the commit timestamp of T1 is less than the commit timestamp of T2, that is, the commit timestamp acquired from T1 is before the commit timestamp acquired for T2.

The transaction commit process is Prepare -> GetCommitTS -> Commit, so there are:

T1.Prepare --> T1.GetCommitTS --> T1.Commit

T2.Prepare --> T2.GetCommitTS --> T2.Commit;

Specifically, the transaction commit process is: prepare, acquire a commit timestamp, and commit, so this transaction commit process is as follows:

T1.Prepare, acquire the commit timestamp of T1, and T1.Commit;

T2.Prepare, acquire the commit timestamp of T2, and T2.Commit.

According to the above, it can be learned that T1.Prepare and acquire the commit timestamp of T1 are before T2.Prepare and acquire the commit timestamp of T2, but there is a delay in the distributed system, so that the sequence that the commit timestamp of T1 and the commit timestamp of T2 reach the standby database may be out of order, and therefore, the sequence of the two events of T1.Commit and T2.Commit is still uncertain.

That is, the Apply log of the standby database will inevitably ensure that the transaction commit sequence of T1 and T2 is kept unchanged, so that when S is created, T1.Prepare have been necessarily completed, and then there are two cases in the timing of T1.Commit:

if T1.Commit is earlier than T2.Commit, T1 will be directly visible to S; and

if T1.Commit is later than T2.Commit, S will block and waits for T1.Commit to reach the standby database, visibility is then determined according to the timestamp.

Specifically, when the commit of T1 is earlier than that of T2, and in a case that T2 is the last write transaction, T1 is visible to S. In a case that the commit of T1 is later than that of T2, but the commit timestamp of T1 is acquired before the commit timestamp of T2 is acquired, and T2.commit_seq is the maximum commit timestamp for the write transaction of the standby databases, but the commit of T1 is later than the commit of T2, in this case, S blocks and waits for T1.Commit to reach the standby database, and it is then determined that T1 is visible to S according to the maximum commit timestamp for the write transaction of the standby databases, to ensure the integrity of the read data of the standby database.

In the data reading method provided in this embodiment of this specification, a current max_commit_seq is maintained on a standby database node, if a snapshot timestamp of a read transaction S meets snapshot_seq < max_commit_seq, it can be considered that the reading is "safe", and the write transaction in the standby database is visible to the read transaction S. If this premise is not met, block and wait may be performed for a short period of time (for example,

1ms) and then retrying is performed. In a case that the maximum commit timestamp for the write transaction of the standby databases is greater than the snapshot timestamp of the read transaction S, it can still be determined that the write transaction in the standby database is visible to the read transaction S, and the request of the read transaction to read data can be quickly and safely satisfied by blocking and waiting.

The data reading method is further described below by using the application of the data reading method provided in this specification in a distributed database as an example with reference to FIG. 3. FIG. 3 is a flowchart of a processing process of a first data reading method according to an embodiment of this specification, which specifically includes the following steps:

During specific implementation, the data reading method provided in this embodiment of this specification includes a user, a CN, a GTS, and distributed standby databases DN1\DN2\DN3.

Step 302: The user sends a data reading transaction request to the CN.

Step 304: After receiving the data reading transaction request, the CN requests the GTS to generate a snapshot timestamp for a data reading transaction and acquires a maximum commit timestamp for a write transaction of the distributed standby databases DN1\DN2\DN3.

Step 306: After generating the snapshot timestamp for the data reading transaction and acquiring the maximum commit timestamp for the write transaction of the distributed standby databases DN1\DN2\DN3, the GTS returns the snapshot timestamp and the maximum commit timestamp to the CN.

Step 308: In a case that it is determined that the snapshot timestamp is smaller than the maximum commit timestamp, the CN sends the data reading transaction request carrying the snapshot timestamp to the corresponding distributed standby databases DN1\DN2\DN3 respectively.

Step 310: The distributed standby databases DN1\DN2\DN3 read data based on the data reading transaction request, and returns the read data to the CN.

Step 312: The CN returns received data to the user.

In the data reading method provided in this embodiment of this specification, acquisition for a timestamp is performed once, and consistency between read data of distributed standby databases and data of the corresponding primary database can be ensured based on the timestamp, which greatly improves the user experience for data reading and querying in the distributed standby databases.

The data reading method is further described below by using the application of the data

reading method provided in this specification in a distributed database as an example with reference to FIG. 4. FIG. 4 is a flowchart of a processing process of a second data reading method according to an embodiment of this specification, which specifically includes the following steps:

During specific implementation, the data reading method provided in this embodiment of this specification includes a user, a CN, a GTS, distributed standby databases DN1\DN2\DN3, and distributed primary databases MDB1\MDB2\MDB3.

Step 402: The user sends a data reading transaction request to the CN.

Step 404: After receiving the data reading transaction request, the CN requests the GTS to generate a snapshot timestamp for a data reading transaction and acquires a maximum commit timestamp for a write transaction of the distributed standby databases DN1\DN2\DN3.

Step 406: After generating the snapshot timestamp for the data reading transaction and acquiring the maximum commit timestamp for the write transaction of the distributed standby databases DN1\DN2\DN3, the GTS returns the snapshot timestamp and the maximum commit timestamp to the CN.

Step 408: In a case that it is determined that the snapshot timestamp is greater than the maximum commit timestamp for the write transaction of the distributed standby databases DN1\DN2\DN3, the CN reacquires a maximum commit timestamp for the write transaction of the distributed standby databases after waiting for a preset duration, and in a case that the snapshot timestamp is still greater than the maximum commit timestamp for the write transaction of the distributed standby databases DN1\DN2\DN3, sends a request for acquiring an LSN to the distributed primary databases MDB1\MDB2\MDB3 respectively.

Step 410: After receiving the request for acquiring an LSN, the distributed primary databases MDB1\MDB2\MDB3 respectively return the corresponding LSN to the CN.

Step 412: The CN sends the data reading transaction request carrying the corresponding LSN to the corresponding distributed standby databases DN1\DN2\DN3 respectively.

Step 414: After receiving the data reading transaction request carrying the LSN, and waiting for an LSN of the local log Apply reaches the LSN carried in the currently received data reading transaction request, the distributed standby databases DN1\DN2\DN3 perform data reading based on the data reading transaction request, and return the read data to the CN.

Step 416: The CN returns received data to the user.

In the data reading method provided in this embodiment of this specification, a current maximum commit timestamp (max_commit_seq) is maintained on a standby database DN node, if the snapshot timestamp meets snapshot_seq>max_commit_seq on the standby database DN

node, it can be considered that the reading is "safe", and data reading can be performed directly; and if this premise is not met, retrying can be performed after a short period of time (for example, 1ms) is waited for, that is, a maximum commit timestamp (max_commit_seq) for the current write transaction maintained by the standby database DN node is reacquired, and if the premise is still not met, a current LSN is acquired from the primary database first, and after the LSN of the standby database is waited for to reach the current LSN, data reading is performed in the standby database to ensure that the read standby database data is consistent with that from the primary database. When reading standby database data in the three manners, the data read from the standby database and the data read from the primary database based on the same data reading transaction are ensured to be kept consistent, which improves the user experience for data reading, and reading of the standby database can be generally completed within one round-trip(RT), thereby improving the data querying performance.

Corresponding to the foregoing method embodiments, this specification further provides an embodiment of a data reading apparatus. FIG. 5 is a schematic structural diagram of a data reading apparatus according to an embodiment of this specification. As shown in FIG. 5, the apparatus includes:

- a request receiving module 502, configured to receive a data reading transaction request;
- a timestamp acquisition module 504, configured to acquire a snapshot timestamp based on the data reading transaction request; and
- a data reading module 506, configured to send, in a case that the snapshot timestamp is less than a maximum commit timestamp for a write transaction of first databases, the data reading transaction request carrying the snapshot timestamp to the first databases for data reading.

Optionally, the data reading module 506 is further configured to that:

the snapshot timestamp is less than a maximum commit timestamp for a write transaction of a second database to the first databases, where the first database is a backup database of the second database.

Optionally, the apparatus further includes:

- a retry module, configured to reacquire, in a case that the snapshot timestamp is greater than the maximum commit timestamp for the write transaction of the first databases, a maximum commit timestamp for the write transaction of the first databases after waiting for a preset duration.

Optionally, the apparatus further includes:

- a first sequence number acquisition module, configured to acquire an LSN of a second

database in a case that the snapshot timestamp is less than the maximum commit timestamp for the write transaction of the first databases;

a second sequence number acquisition module, configured to send the data reading transaction request carrying the LSN of the second database to the first database, and acquire an LSN of the first database; and

a sequence number matching module, configured to read data in the first database based on the data reading transaction request and the LSN of the first database in a case that it is determined that the LSN of the first database matches the LSN of the second database.

Optionally, the first sequence number acquisition module is further configured to:

acquire an LSN of the last complete database backup of the second database; and

the second sequence number acquisition module is correspondingly further configured to: acquire an LSN of the last complete database backup of the first database.

Optionally, the apparatus further includes:

a database determining module, configured to determine at least one corresponding second database based on the data reading transaction request.

Optionally, the first sequence number acquisition module is further configured to:

acquire an LSN of each second database; and

the second sequence number acquisition module is correspondingly further configured to: send the data reading transaction request carrying the LSN of the second database to the first database corresponding to each second database.

Optionally, the data reading method is applicable to a distributed database.

An embodiment of this specification provides a data reading apparatus, which acquires a data reading transaction request; acquires a snapshot timestamp based on the data reading transaction request; and sends, in a case that the snapshot timestamp is less than a maximum commit timestamp for a write transaction of first databases, the data reading transaction request to the first databases for data reading, and receives data read by the first databases based on the data reading transaction request. By using the data reading apparatus, acquisition for a timestamp is performed once, and consistency between data of distributed standby databases and data of the corresponding primary database read under the same data reading transaction request can be ensured based on the timestamp, which greatly improves the user experience for data reading and querying in the distributed standby databases.

The foregoing describes an exemplary solution of the data reading apparatus according to this embodiment. It should be noted that, the technical solution of the data reading apparatus and the technical solution of the data reading method belong to the same conception, and for

detailed content of the technical solution of the data reading apparatus that is not described in detail, reference may be made to the description of the technical solution of the foregoing data reading method.

FIG. 6 is a structural block diagram of a computing device 600 according to an embodiment of this specification. Components of the computing device 600 include, but are not limited to, a memory 610 and a processor 620. The processor 620 and the memory 610 are connected through a bus 630, and a database 650 is configured to store data.

The computing device 600 further includes an access device 640, and the access device 640 enables the computing device 600 to communicate through one or more networks 660. Examples of these networks include a public switched telephone network (PSTN), a local area network (LAN), a wide area network (WAN), a personal area network (PAN), or a combination of communication networks such as the internet. The access device 640 may include one or more of wired or wireless network interfaces (for example, a network interface card (NIC)) of any types, such as an IEEE802.11 wireless local area network (WLAN) wireless interface, a worldwide interoperability for microwave access (Wi-MAX) interface, an Ethernet interface, a universal serial bus (USB) interface, a cellular network interface, a Bluetooth interface, a near field communication (NFC) interface, and the like.

In an embodiment of this specification, the foregoing components and other components not shown in FIG. 6 of the computing device 600 may be connected, for example, through the bus. It should be understood that the structural block diagram of the computing device shown in FIG. 6 is merely exemplary, and is not intended to limit the scope of this specification. A person skilled in the art may add or replace other components according to a requirement.

The computing device 600 may be a stationary or mobile computing device of any type, including a mobile computer, a mobile computing device (for example, a tablet computer, a person digital assistant, a laptop computer, a notebook, or a netbook), a mobile phone (for example, a smartphone), a wearable computing device (for example, a smartwatch or smart glasses), a mobile device of another type, or a stationary computing device such as a desktop computer or a personal computer (PC). The computing device 600 may alternatively be a mobile or stationary server.

The processor 620 is configured to execute the following computer-executable instructions, and the processor, when executing the computer-executable instructions, implements the steps of the data reading method.

The foregoing describes an exemplary solution of the computing device according to this embodiment. It should be noted that, the technical solution of the computing device and the

technical solution of the data reading method belong to the same conception, and for detailed content of the technical solution of the computing device that is not described in detail, reference may be made to the description of the technical solution of the foregoing data reading method.

An embodiment of this specification further provides a computer-readable storage medium, storing computer instructions, the instructions, when executed by a processor, implementing the steps of the data reading method.

The foregoing describes an exemplary solution of the computer-readable storage medium according to this embodiment. It should be noted that, the technical solution of the storage medium and the technical solution of the data reading method belong to the same conception, and for detailed content of the technical solution of the storage medium that is not described in detail, reference may be made to the description of the technical solution of the foregoing data reading method.

Specific embodiments of this specification are described above. Other embodiments fall within the scope of the appended claims. In some cases, the actions or steps recorded in the claims may be performed in sequences different from those in the embodiments and an expected result may still be achieved. In addition, the processes depicted in the accompanying drawings are not necessarily performed in the shown specific order or successively to achieve an expected result. In some implementations, multitasking and parallel processing may be feasible or beneficial.

The computer instructions include computer program code. The computer program code may be in a source code form, an object code form, executable file or some intermediate forms, or the like. The computer-readable medium may include: any entity or apparatus that is capable of carrying the computer program code, a recording medium, a USB flash drive, a removable hard disk, a magnetic disk, an optical disc, a computer memory, a read-only memory (ROM), a random access memory (RAM), an electric carrier signal, a telecommunication signal and a software distribution medium, and the like. It should be noted that, the content contained in the computer-readable medium may be appropriately increased or decreased according to the requirements of legislation and patent practice in jurisdictions. For example, in some jurisdictions, according to legislation and patent practice, the computer-readable medium does not include an electric carrier signal and a telecommunication signal.

It should be noted that, for ease of description, the foregoing method embodiments are represented as a series of action combinations, but a person skilled in the art should know that the embodiments of this specification are not limited to the described order of the actions

because some steps may be performed in another order or performed simultaneously according to the embodiments of this specification. In addition, a person skilled in the art should also appreciate that all the embodiments described in this specification are exemplary embodiments, and the related actions and modules are not necessarily mandatory to the embodiments of this specification.

In the foregoing embodiments, the description of each embodiment has respective focuses. For a part that is not described in detail in an embodiment, reference may be made to related descriptions in other embodiments.

The exemplary embodiments of this specification disclosed above are only used to help explain this specification. The optional embodiments do not describe all the details, and are not intended to limit this application to the specific implementations described. Obviously, many modifications and changes may be made according to the content of the embodiments of this specification. These embodiments are selected and described in this specification for better explaining the principles and practical applications of the embodiments of this specification, so that a person skilled in the art can better understand and use this specification. This specification is only subject to the claims and full scope and equivalents thereof.

CLAIMS

What is claimed is:

[Claim 1] A data reading method, comprising:
receiving a data reading transaction request;
acquiring a snapshot timestamp based on the data reading transaction request; and
sending, in a case that the snapshot timestamp is less than a maximum commit timestamp for a write transaction of first databases, the data reading transaction request carrying the snapshot timestamp to the first databases for data reading.

[Claim 2] The data reading method according to claim 1, wherein that the snapshot timestamp is less than a maximum commit timestamp for a write transaction of first databases comprises:

the snapshot timestamp being less than a maximum commit timestamp for a write transaction of a second database to the first databases, wherein the first database is a backup database of the second database.

[Claim 3] The data reading method according to claim 1, wherein after the acquiring a snapshot timestamp based on the data reading transaction request, the method further comprises:

reacquiring, in a case that the snapshot timestamp is greater than the maximum commit timestamp for the write transaction of the first databases, a maximum commit timestamp for the write transaction of the first databases after waiting for a preset duration.

[Claim 4] The data reading method according to claim 1 or 3, wherein after the reacquiring a maximum commit timestamp for the write transaction of the first databases after waiting for a preset duration, the method further comprises:

acquiring a log sequence number (LSN) of a second database in a case that the snapshot timestamp is less than the maximum commit timestamp for the write transaction of the first databases;

sending the data reading transaction request carrying the LSN of the second database to the first database, and acquiring an LSN of the first database; and

reading data in the first database based on the data reading transaction request and the LSN of the first database in a case that it is determined that the LSN of the first database matches the LSN of the second database.

[Claim 5] The data reading method according to claim 4, wherein the acquiring an LSN of a second database comprises:

acquiring an LSN of the last complete database backup of the second database; and

the acquiring an LSN of the first database correspondingly comprises:

acquiring an LSN of the last complete database backup of the first database.

[Claim 6] The data reading method according to claim 4, wherein after the receiving a data reading transaction request, the method further comprises:

determining at least one corresponding second database based on the data reading transaction request.

[Claim 7] The data reading method according to claim 6, wherein the acquiring an LSN of a second database comprises:

acquiring an LSN of each second database; and

the sending the data reading transaction request carrying the LSN of the second database to the first database correspondingly comprises:

sending the data reading transaction request carrying the LSN of the second database to the first database corresponding to each second database.

[Claim 8] The data reading method according to any one of claims 1 to 7, wherein the data reading method is applicable to a distributed database.

[Claim 9] A data reading apparatus, comprising:

a request receiving module, configured to receive a data reading transaction request;

a timestamp acquisition module, configured to acquire a snapshot timestamp based on the data reading transaction request; and

a data reading module, configured to send, in a case that the snapshot timestamp is less than a maximum commit timestamp for a write transaction of first databases, the data reading transaction request carrying the snapshot timestamp to the first databases for data reading.

[Claim 10] A computing device, comprising:

a memory and a processor, wherein

the memory is configured to store computer-executable instructions, the processor is configured to execute the computer-executable instructions, and the processor, when executing the computer-executable instructions, implements the steps of the data reading method according to any one of claims 1 to 8.

[Claim 11] A computer-readable storage medium, storing computer instructions, the instructions, when executed by a processor, implementing the steps of the data reading method according to any one of claims 1 to 8.