

# **DATA PROCESSING METHOD AND APPARATUS**

The present application claims priority to Chinese Patent Application Serial No. 202010567428.X, titled “DATA PROCESSING 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 processing method. One or more embodiments of this specification also relate to a data processing apparatus, a computing device, and a computer-readable storage medium.

## **BACKGROUND**

In Internet database solutions, the technology of read-write separation is generally used to realize the read and write operations of data. The traditional read-write separation provides a read-write separation connection string. For non-transactional read requests, the read requests are sent to a read-only database according to a read weight ratio configured by a user, but transactional read requests are directly sent to a primary instance (that is, a primary database, which provides complete services, including read and write links). If a transaction contains an on-line analytical processing (OLAP) type read request, the OLAP type read request occupies too many resources of the primary instance, affecting the performance and stability of the primary instance.

Therefore, there is an urgent need to provide a data processing method that can reduce and avoid the preemption of resources of the primary instance by OLAP type read requests, thereby ensuring the stability of the database.

## **SUMMARY**

In view of this, embodiments of this specification provide a data processing method. One or more embodiments of this specification also relate to a data processing apparatus, a computing device, and a computer-readable storage medium, so as to solve the technical defects in the prior art.

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

receiving a data processing request carrying a query statement;

converting the query statement into a corresponding relational algebra tree based on the data processing request, and determining an operation type corresponding to the query statement based on the relational algebra tree; and

delivering the query statement to a first database in response to the operation type being a first type, and completing the data processing request in the first database based on the query statement.

Optionally, the converting the query statement into a corresponding relational algebra tree includes:

parsing the query statement into a corresponding abstract syntax tree, and converting the abstract syntax tree into the corresponding relational algebra tree, where the relational algebra tree consists of tree nodes and edges.

Optionally, the determining an operation type corresponding to the query statement based on the relational algebra tree includes:

determining structural information of the query statement based on the relational algebra tree; and

determining that the operation type corresponding to the query statement is a first type in response to the structural information of the query statement satisfying a first preset rule.

Optionally, the determining an operation type corresponding to the query statement based on the relational algebra tree includes:

determining logical information of the query statement based on the relational algebra tree; and

determining that the operation type corresponding to the query statement is a first type in response to the logical information of the query statement satisfying a second preset rule.

Optionally, after the determining structural information of the query statement based on the relational algebra tree, the method further includes:

determining logical information of the query statement based on the relational algebra tree in response to the structural information of the query statement not satisfying the first preset rule; and

determining that the operation type corresponding to the query statement is the first type in response to the logical information of the query statement satisfying a second preset rule.

Optionally, the determining an operation type corresponding to the query statement based on the relational algebra tree includes:

determining structural information and logical information of the query statement based

on the relational algebra tree;

determining an execution plan of the query statement based on the relational algebra tree and calculating a physical execution cost of the query statement according to the execution plan, in response to the structural information of the query statement not satisfying a first preset rule and the logical information of the query statement not satisfying a second preset rule; and

determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold.

Optionally, the determining an operation type corresponding to the query statement based on the relational algebra tree includes:

determining an execution plan of the query statement based on the relational algebra tree and calculating a physical execution cost of the query statement according to the execution plan; and

determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold.

Optionally, the determining an execution plan of the query statement based on the relational algebra tree and calculating a physical execution cost of the query statement according to the execution plan includes:

determining tree nodes of the relational algebra tree and determining the execution plan of the query statement based on the tree nodes;

calculating a physical execution cost of each of the tree nodes corresponding to the query statement based on the execution plan of the query statement; and

performing a weighted summation of the physical execution costs of the tree nodes to obtain the physical execution cost of the query statement.

Optionally, the determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold includes:

determining that the operation type corresponding to the query statement is the first type in response to the physical execution cost of any of the tree nodes corresponding to the query statement being less than or equal to the preset cost threshold.

Optionally, the method further includes:

determining that the operation type corresponding to the query statement is a second type in response to the physical execution cost being greater than the preset cost threshold.

Optionally, after the determining that the operation type corresponding to the query statement is a second type, the method further includes:

delivering the query statement to a second database in response to the operation type corresponding to the query statement being the second type, and completing the data processing request in the second database based on the query statement.

Optionally, the first database is a distributed read-write database for processing query statements of the first type, and the second database is a distributed read-only database for processing query statements of the second type.

Optionally, the first type is an on-line transaction processing (OLTP) type; and

correspondingly, the delivering the query statement to a first database in response to the operation type being a first type includes:

delivering the query statement to the distributed read-write database in response to the operation type being the OLTP type.

Optionally, the second type is an on-line analytical processing (OLAP) type; and

correspondingly, the delivering the query statement to a second database in response to the operation type corresponding to the query statement being the second type includes:

delivering the query statement to the distributed read-only database in response to the operation type corresponding to the query statement being the OLAP type.

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

a processing request receiving module, configured to receive a data processing request carrying a query statement;

an operation type determination module, configured to convert the query statement into a corresponding relational algebra tree based on the data processing request, and determine an operation type corresponding to the query statement based on the relational algebra tree; and

a database determination module, configured to deliver the query statement to a first database in response to the operation type being a first type, and complete the data processing request in the first database based on the query statement.

According to a third aspect of the embodiments of this specification, a data processing system is provided, including:

a routing module, a read-write distributed data node, and a read-only distributed data node, where,

the routing module is configured to receive a data processing request carrying a query statement, and determine an operation type corresponding to the query statement based on the data processing request;

deliver the query statement to the read-write distributed data node in response to the

operation type being an on-line transaction processing (OLTP) type, and complete the data processing request at the read-write distributed data node based on the query statement; and

deliver the query statement to the read-only distributed data node in response to the operation type being an on-line analytical processing (OLAP) type, and complete the data processing request at the read-only distributed data node based on the query statement.

According to a fourth 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, and the computer-executable instructions, when executed by the processor, implement the operations of the data processing method.

According to a fifth aspect of the embodiments of this specification, a computer-readable storage medium is provided, storing computer-executable instructions, where the computer-executable instructions, when executed by a processor, implement the operations of the data processing method.

An embodiment of this specification provides a data processing method and apparatus. The method includes: receiving a data processing request carrying a query statement; converting the query statement into a corresponding relational algebra tree based on the data processing request, and determining an operation type corresponding to the query statement based on the relational algebra tree; and delivering the query statement to a first database in response to the operation type being a first type, and completing the data processing request in the first database based on the query statement. By the data processing method, it can be intelligently identified whether an operation type corresponding to an SQL statement sent by a client is a first type, i.e., an OLTP type, or a second type, i.e., an OLAP type. Only when the operation type corresponding to the SQL statement is the OLTP type, the SQL statement is delivered to a primary instance for processing, thereby ensuring low latency and high throughput of OLTP workload traffic and ensuring the stability of the database.

## **BRIEF DESCRIPTION OF THE DRAWINGS**

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

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

FIG. 3 is a flowchart of an application of a data processing method to a hybrid transactional/analytical processing (HTAP) distributed database according to an embodiment of this specification.

FIG. 4 is a detailed flowchart of analysis of an operation type corresponding to an SQL statement by an intelligent router in a data processing method according to an embodiment of this specification.

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

FIG. 6 is a schematic structural diagram of a data processing system according to an embodiment of this specification.

FIG. 7 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 embodiments disclosed below.

The terms used in one or more embodiments of this specification are merely used to describe the 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. Depending on the context, for example, the term "if" used herein may be interpreted as "while" or "when", or "in response to

determination".

First, the terms involved in one or more embodiments of this specification are explained.

Workload: Workload in a database mainly includes OLTP workload and OLAP workload.

OLTP: On-Line Transaction Processing, mainly including SQL such as addition, deletion, modification and query involving a small range of data.

OLAP: On-Line Analytical Processing, mainly including complex query analytical SQL involving a large range of data.

HTAP: Hybrid Transactional/Analytical Processing, workload in which OLTP workload and OLAP workload coexist.

Read-write instance: A database instance on which read and write operations can be performed.

Read-only instance: A database instance on which only read operations can be performed.

SQL: Structured Query Language, which is a special-purpose programming language, is a programming language for database query, and is used to access data and query, update and manage relational database systems.

Execution plan: One SQL statement corresponds to multiple equivalent execution plans. The execution plan includes how to actually execute one SQL statement.

Cost: The cost of the execution plan includes information such as CPU, Memory, and IO.

RT: Response Time, which is the time from requesting a transaction to returning a result after the system completes processing of the transaction.

This specification provides a data processing method. One or more embodiments of this specification also relate to a data processing apparatus, a computing device, and a computer-readable storage medium, which will be described in detail one by one in the following embodiments.

Referring to FIG. 1, FIG. 1 is a schematic architectural diagram of a data processing apparatus according to an embodiment of this specification. The apparatus includes: a client 102, an intelligent router 104 (i.e. read-write separation link string), a first database 106, and a second database 108. The first database 106 may be construed as a distributed read-write database, and the second database 108 may be construed as a distributed read-only database. In practical applications, the first database 106 may also be construed as a read-write distributed data node of a distributed read-write database, that is, an RW node, and the second database 108 may also be construed as a read-only distributed data node of a distributed read-only database, that is, an RO node.

In a specific implementation, the client 102 sends a query statement for the first database

106 or the second database 108 for data processing. After receiving the query statement sent by the client 102, the intelligent router 104 parses the query statement to determine an execution plan corresponding to the query statement. Then, the intelligent router 104 calculates a physical execution cost of the query statement based on the execution plan. The physical execution cost is, for example, the number of instructions to be executed by a CPU when executing the query statement, memory required to execute these instructions, the number of IO accesses, how many network interactions are required, etc. Finally, the intelligent router 104 determines an operation type corresponding to the query statement according to the physical execution cost of the query statement. The intelligent router 104 sends the query statement to the first database 106 when the operation type is an OLTP type, and sends the query statement to the second database 108 when the operation type is an OLAP type, so as to complete the data processing request of the client in the first database 106 or the second database 108. The query statement is an SQL statement.

A detailed processing process of the data processing method will be described in detail below. In a specific implementation, the data processing method is applied to an HTAP distributed database. The following data processing methods are all described with the intelligent router 104 as the execution entity.

Referring to FIG. 2, FIG. 2 is a flowchart of a data processing method according to an embodiment of this specification. The method includes the following steps.

Step 202: Receive a data processing request carrying a query statement.

The data processing request includes but is not limited to a data query request, a data update request, a data read request, a data write request, etc. The query statement is an SQL statement.

The query statement carried in the data processing request is a specific data requirement of the data processing request. For example, the data processing request is a data insertion request, and the query statement carried in the data processing request may specifically reflect which data is to be read from which table of which database and inserted into which table of which database.

For example, the data processing request sent by the client is a data insertion request, and correspondingly the query statement carried in the data insertion request may be: insert into T select name from S, that is, select a name field from a table S and insert the name field into a table T.

Step 204: convert the query statement into a corresponding relational algebra tree based on the data processing request, and determine an operation type corresponding to the query

statement based on the relational algebra tree.

The converting the query statement into a corresponding relational algebra tree based on the data processing request may be construed as parsing the query statement based on the data processing request. A specific process of parsing the query statement is the process of converting the query statement into the corresponding relational algebra tree.

Specifically, the converting the query statement into a corresponding relational algebra tree includes:

parsing the query statement into a corresponding abstract syntax tree, and converting the abstract syntax tree into the corresponding relational algebra tree, where the relational algebra tree consists of tree nodes and edges.

The abstract syntax tree may be abbreviated as AST. Parsing the query statement into the corresponding abstract syntax tree is to parse the query statement and express its grammatical structure in the form of a tree.

Stilling taking the query statement carried in the data insertion request "insert into T select name from S" as an example, parsing the query statement into the corresponding abstract syntax tree is to express the query statement in the form of a tree. For example, "insert" is used as the first node of the tree, under which two input branch nodes are provided: one node is "T", and the other node is "select name from S". Two branch nodes may further be provided under the node "select name from S": node "S" and node "name".

After the query statement is parsed into the corresponding abstract syntax tree, the abstract syntax tree is converted into the corresponding relational algebra tree. The relational algebra tree consists of tree nodes and edges. Stilling taking the query statement carried in the data insertion request as an example, the tree nodes of the relational algebra tree include at least a table scan node, a mapping node and a write node. The query statement may be determined based on the tree nodes of the relational algebra tree and a connection relationship between the edges connecting the tree nodes: need to scan the table S, map the name field in the table S, and then write the mapped name field to the table T.

Specifically, after the query statement is specifically converted into the corresponding relational algebra tree, the operation type corresponding to the query statement may be determined based on the relational algebra tree.

In the embodiments of this specification, through the optimization strategy of converting the query statement into the relational algebra tree structure, the data processing method greatly improves the data processing efficiency during data processing on the database based on the query statement having the relational algebra tree structure.

In a specific implementation, the operation type of the query statement may be quickly determined based on structural information of the query statement that is based on the relational algebra tree corresponding to the query statement. A specific implementation method is as follows:

the determining an operation type corresponding to the query statement based on the relational algebra tree includes:

determining the structural information of the query statement based on the relational algebra tree; and

determining that the operation type corresponding to the query statement is a first type in response to the structural information of the query statement satisfying a first preset rule.

The first preset rule is a preset requirement that structural information corresponding to a query statement of the first type needs to meet.

Specifically, the relational algebra tree of the query statement may reflect specific structural information of the corresponding SQL statement, e.g., whether the SQL statement includes a primary key, whether the primary key has an index, whether the index has an equivalent condition, etc. If the first preset rule is that the query statement needs to include a primary key, the primary key has a corresponding index, and the index has an equivalent condition, and it is determined based on the relational algebra tree corresponding to the query statement that the structural information of the query statement includes a primary key, the primary key has a corresponding index, and the index has an equivalent condition, it may be determined that the structural information of the query statement satisfies the first preset rule. In this case, it may be determined that the operation type corresponding to the query statement is the first type. The first type is the OLTP type described above.

In the embodiments of this specification, by the data processing method, the operation type corresponding to the query statement can be quickly determined by determining whether the structural information of the query statement based on the relational algebra tree structure matches the preset rule, and then based on the accurate operation type, the query statement can be delivered to a suitable database for data processing, thereby greatly improving the efficiency of data processing.

In a specific application, the relational algebra tree corresponding to each query statement is expressed by tree nodes, that is, each query statement may be expressed by the corresponding tree nodes. Therefore, the operation type corresponding to the query statement may be determined more quickly based on the tree nodes corresponding to the query statement. A specific implementation is as follows:

The determining structural information of the query statement based on the relational algebra tree includes:

determining structural information of the query statement based on the relational algebra tree; and

correspondingly, the determining that the operation type corresponding to the query statement is a first type in response to the structural information of the query statement satisfying a first preset rule includes:

determining that the operation type corresponding to the query statement is a first type in response to the tree nodes in the structural information of the query statement satisfying a first preset rule.

The tree nodes of the relational algebra tree corresponding to the query statement can express the specific meaning of each query statement. After the tree nodes of the relational algebra tree are determined, the structural information of the corresponding query statement can be seen according to the tree nodes. That is, it may be considered that the structural information of the query statement includes a plurality of tree nodes.

Still taking the query statement carried in the data insertion request as an example, it may be determined by parsing the query statement that the relational algebra tree corresponding to the query statement includes at least three tree nodes: a table scan node, a mapping node, a write node, etc. In this case, it may be determined that the structural information of the query statement includes at least: the table scan node, the mapping node, the write node, etc. It is assumed that the first preset rule is that the number of tree nodes in the structural information of the query statement is less than or equal to a preset number threshold of, for example 3, it is determined that the operation type corresponding to the query statement is the first type. Because it can be known from the above description that the structural information of the query statement includes three tree nodes, i.e., satisfies the first preset rule, it may be determined that the operation type corresponding to the query statement is the first type.

By the data processing method provided in the embodiments of this specification, the operation type corresponding to the query statement can be determined more quickly by determining whether the tree nodes of the structural information of the query statement based on the relational algebra tree structure match the preset rule, and then based on the accurate operation type, the query statement can be delivered to a suitable database for data processing, thereby greatly improving the efficiency of data processing.

In addition, logical information corresponding to the query statement may also be determined based on the relational algebra tree corresponding to the query statement. In another

possible implementation, the operation type corresponding to the query statement may also be determined by determining whether the logical information corresponding to the query statement matches a preset rule. A specific implementation is as follows:

the determining an operation type corresponding to the query statement based on the relational algebra tree includes:

determining the logical information of the query statement based on the relational algebra tree; and

determining that the operation type corresponding to the query statement is a first type in response to the logical information of the query statement satisfying a second preset rule.

The second preset rule is a preset requirement that logical information corresponding to a query statement of the first type needs to meet. For example, the second preset rule includes a requirement on the number of rows in the table and the number of rows after filtering.

Specifically, the logical information of the query statement may be determined based on the relational algebra tree. Still taking the query statement carried in the data insertion request as an example, it may be determined based on the relational algebra tree of the query statement that the logical information of the query statement is: the total number of rows in the table S is 1000, and if there is a where condition after the query statement: where age=20, the number of rows in the name field in the table S filtered according to the where condition is 300.

In this case, if the second preset rule is that the number of rows in the table in the query statement is less than or equal to a first preset row number threshold of, for example, 1000, and the number of rows after filtering is less than or equal to a second preset row number threshold of, for example, 500, it may be determined that the logical information of the query statement matches the second preset rules match. In this case, it may be determined that the operation type corresponding to the query statement is the first type.

By the data processing method provided in the embodiments of this specification, the operation type corresponding to the query statement can be determined quickly and conveniently by determining whether the logical information of the query statement matches preset logical information, thereby improving user experience.

In another possible embodiment, to ensure the accuracy of judgment of query statements of the first type and ensure that the query statement will not be directly defined as the second type even if the structural information of the query statement does not meet the first preset rule, a further judgment may be made based on the logical information of the query statement. A specific implementation is as follows:

after the determining structural information of the query statement based on the relational

algebra tree, the method further includes:

- determining logical information of the query statement based on the relational algebra tree in response to the structural information of the query statement not satisfying the first preset rule; and

- determining that the operation type corresponding to the query statement is a first type in response to the logical information of the query statement satisfying a second preset rule.

Specifically, when the structural information of the query statement determined based on the relational algebra tree does not meet the first preset rule, but it is determined that the logical information of the query statement meets the second preset rule, the operation type corresponding to the query statement is still defined as the first type, so that data processing of the query statement may subsequently be performed in the first database corresponding to the first type, thereby improving the low-latency and high-throughput processing of the query statement of the first type.

In practical applications, even if the structural information of the query statement does not meet the first preset rule and the logical information of the query statement does not meet the second preset rule, a further judgment may be made on the query statement, to realize lossless screening of query statements of the first type. A specific implementation is as follows:

- the determining an operation type corresponding to the query statement based on the relational algebra tree includes:

- determining structural information and logical information of the query statement based on the relational algebra tree;

- determining an execution plan of the query statement based on the relational algebra tree and calculating a physical execution cost of the query statement according to the execution plan, in response to the structural information of the query statement not satisfying a first preset rule and the logical information of the query statement not satisfying a second preset rule; and

- determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold.

The execution plan may be construed as a plan for actually executing an SQL statement. For example, reading of a field of a table in the SQL statement is executed first, and then writing of the read field in the SQL statement is executed. The physical execution cost is defined as Cost, and is the number of instructions to be executed by a CPU when executing an SQL statement, memory required to execute these instructions, the number of IO accesses, how many network interactions are required, etc.

The preset cost threshold may be set according to practical applications, for example, set

to 20,000, 30,000, etc., which is not limited herein.

Specifically, first, the execution plan of the corresponding query statement is determined based on the relational algebra tree, and then the physical execution cost required for executing the SQL statement is calculated according to the execution plan. Then, when the physical execution cost is less than or equal to the preset cost threshold of, for example, 20,000, it is determined that the operation type corresponding to the query statement is the first type.

In another possible implementation of this specification, after the query statement is received, the operation type corresponding to the query statement may be determined directly based on the above-mentioned cost rule, thereby improving the efficiency of data processing. A specific implementation is as follows:

the determining an operation type corresponding to the query statement based on the relational algebra tree includes:

determining an execution plan of the query statement based on the relational algebra tree and calculating a physical execution cost of the query statement according to the execution plan; and

determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold.

Specifically, the determining an execution plan of the query statement based on the relational algebra tree and calculating a physical execution cost of the query statement according to the execution plan includes:

determining tree nodes of the relational algebra tree and determining the execution plan of the query statement based on the tree nodes;

calculating a physical execution cost of each of the tree nodes corresponding to the query statement based on the execution plan of the query statement; and

performing a weighted summation of the physical execution costs of the tree nodes to obtain the physical execution cost of the query statement.

In a specific implementation, the physical execution cost of each tree node in the query statement may be calculated according to the execution plan of the query statement, and then a weighted summation of the physical execution costs of all the tree nodes in the query statement is performed to obtain the final physical execution cost of the final query statement.

In practical applications, a different weight is set for each physical execution cost. Therefore, the physical execution cost of the query statement is obtained by multiplying each physical execution cost by the corresponding weight and summing the products, so as to ensure the precision of the final physical execution cost of the query statement.

In specific applications, the comparison between the physical execution cost and the preset cost threshold includes two cases. In a first case,

the determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold includes:

determining that the operation type corresponding to the query statement is the first type in response to the physical execution cost of any of the tree nodes corresponding to the query statement being less than or equal to the preset cost threshold.

That is, after the physical execution cost of each tree node of the query statement is calculated, the operation type corresponding to the corresponding query statement may be determined based on the physical execution cost of each tree node. For example, the physical execution cost of each tree node has a corresponding preset cost threshold. For example, if the query statement includes tree node 1 and tree node 2, the preset cost threshold corresponding to tree node 1 is 2000, and the preset cost threshold corresponding to tree node 2 is 3000, it may be determined that the operation type corresponding to the query statement is the first type, regardless whether the physical execution cost of the tree node 1 or the tree node 2 in the query statement is less than or equal to its corresponding preset cost threshold.

In a second case, the determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold includes:

determining that the operation type corresponding to the query statement is the first type in response to a sum of the physical execution costs of all the tree nodes corresponding to the query statement being less than or equal to the preset cost threshold.

That is, the total physical execution cost of the query statement is obtained through the weighted summation of the physical execution costs of all the tree nodes, and when the total physical execution cost is less than or equal to the preset cost threshold, it is determined that the operation type corresponding to the query statement is the first type.

By the data processing method provided in the embodiments of this specification, after the physical execution cost of each tree node is determined according to the execution plan of the query statement, the operation type corresponding to the query statement may be determined as the first type in a case that the physical execution cost of any one of the tree nodes corresponding to the query statement is less than or equal to the preset cost threshold. This can greatly improve the screening rate of query statements of the first type and ensure that processing of query statements of the OLTP type can subsequently be executed in the first

database, thereby improving improve the efficiency of data processing.

Step 206: Deliver the query statement to a first database in response to the operation type being a first type, and complete the data processing request in the first database based on the query statement.

Specifically, when the operation type corresponding to the query statement is determined as the first type by the various methods described above, the query statement may be delivered to the first database, and the data processing request may be executed in the first database according to the query statement.

In a specific implementation, the first type is an OLTP type. In this case, the first database is a distributed read-write database; and

correspondingly, the delivering the query statement to a first database in response to the operation type being a first type includes:

delivering the query statement to the distributed read-write database in response to the operation type being the OLTP type.

In another embodiment of this specification, after the judgment made on the query statement by all the above methods, if the physical execution cost of the query statement is greater than the preset cost threshold, it may be determined that the operation type corresponding to the query statement is the second type. A specific implementation is as follows:

The method further includes:

determining that the operation type corresponding to the query statement is a second type in response to the physical execution cost being greater than the preset cost threshold.

The second type is the OLAP type. That is, only when the structural information of the query statement does not meet the first preset rule, the logical information of the query statement does not meet the second preset rule, and the physical execution cost of the query statement is greater than the preset cost threshold, it is determined that the operation type corresponding to the query statement is the OLAP type. Therefore, the separation of reading and writing of data processing is achieved, thereby ensuring the stability of the database.

The query statement is delivered to a second database in response to the operation type corresponding to the query statement being the second type, and the data processing request is completed in the second database based on the query statement. A specific implementation is as follows:

after the determining that the operation type corresponding to the query statement is a second type, the method further includes:

delivering the query statement to a second database in response to the operation type

corresponding to the query statement being the second type, and completing the data processing request in the second database based on the query statement.

In a specific implementation, the second type is an OLAP type. In this case, the second database is a distributed read-only database; and

correspondingly, the delivering the query statement to a second database in response to the operation type corresponding to the query statement being the second type includes:

delivering the query statement to the distributed read-only database in response to the operation type corresponding to the query statement being the OLAP type.

Specifically, query statements of the OLAP type are generally complex analytic query statements. If query statements of both the OLAP type and the OLTP type are delivered to the same database for data processing, resource preemption occurs. To ensure the stability of the database, the good read-write separation strategy implemented in the data processing method in the embodiments of the specification is adopted, which can reduce and avoid preemption of resources of OLTP workloads by OLAP workloads, thereby making the database more stable.

Specifically, the first database is a distributed read-write database for processing query statements of the first type, and the second database is a distributed read-only database for processing query statements of the second type. That is, the first database may be construed as a primary database, and the second database may be construed as a standby database. Data in the primary database is synchronized to the standby database, but the data in the standby database has a certain delay. Data processing of query statements of the OLTP type generally involves changing the data in the database, and data processing of query statements of the OLAP type is generally query and search of the data in the database. To ensure the integrity of the data and the efficiency of data processing, the data processing of query statements of the OLTP type is distributed to the first database for execution to ensure the read-write integrity of the data, the data processing of query statements of the OLAP type is distributed to the second database for execution, to avoid the preemption of resources of OLTP workloads by OLAP workloads, thereby making the database more stable and improving the efficiency of data processing.

Referring to Fig. 3, taking the application of the data processing method provided in this specification to an HTAP distributed database as an example, the case where the operation type corresponding to the SQL statement is the OLTP type in the data processing method is further described, specifically including the following steps.

Step 302: A client sends a data processing request carrying an SQL statement to an HTAP distributed database.

Step 304: An intelligent router in the distributed database parses the SQL statement and determines an operation type corresponding to the SQL statement.

Step 306: When determining that the operation type corresponding to the SQL statement is an OLTP type, the intelligent router in the distributed database delivers the SQL statement to the corresponding RDS1 data node.

An RDS data node is a distributed data node of a read-write database. For example, both the RDS1 and the RDS2 in FIG. 3 are distributed data nodes of a read-write databases. The only difference between the RDS1 and the RDS2 is that data stored in the RDS1 is different from that stored in the RDS2. For example, the RDS1 stores data of name and age fields, and the RDS2 stores data of duty and address fields. After the SQL statement is parsed, the SQL statement that requires data processing of the fields in the RDS1 is delivered to the RDS1, and the SQL statement that requires data processing of the fields in the RDS2 is delivered to the RDS2.

Step 308: The RDS1 receives the SQL statement and executes the SQL statement on a data node thereof.

Step 310: The RDS1 returns a result of the execution of the SQL statement to the distributed database.

Step 312: The distributed database delivers the SQL statement to the corresponding RD2 data node.

Step 314: The RDS2 receives the SQL statement and executes the SQL statement on a data node thereof.

Step 316: The RDS2 returns a result of the execution of the SQL statement to the distributed database.

Step 318: The distributed database integrates the results returned by the RDS1 and the RDS2.

Step 320: The distributed database returns a result of the integration to the client.

Specifically, steps 306, 308 and 310 and steps 312, 314 and 316 are not limited to a particular execution order, but may be executed concurrently in practical use. The execution order may be set according to practical applications.

In addition, an SQL statement may be delivered to multiple RDS data nodes of a distributed database for execution at the same time. The specific number of RDS data nodes required for executing the SQL statement depends on specific contents to be executed in the parsed SQL statement, and is not limited herein. For example, if the data to be executed in the SQL statement is stored in five RDS data nodes respectively, the execution of the SQL statement

requires delivering the SQL statement to the corresponding five RDS data nodes for data processing at the same time.

By the data processing method provided in the embodiments of this specification, after receiving a data processing request of an SQL statement from a user, an intelligent router module analyzes a workload type of the SQL statement, and when determining according to a result of the analysis that the workload type of the SQL statement is an OLTP type, executes the SQL statement in a distributed data node of a read-write database, thereby achieving low-latency and high-throughput data processing.

Referring to FIG. 4, FIG. 4 is a detailed flowchart of analysis of an operation type corresponding to an SQL statement by an intelligent router in a data processing method according to an embodiment of this specification, specifically including the following steps.

Step 402: Parse an SQL statement into an abstract syntax tree.

Specifically, after an SQL statement sent by a client is received, the SQL statement is parsed into an abstract syntax tree.

Step 404: Convert the abstract syntax tree into a relational algebra tree.

Step 406: Determine a route according to structural information and logical information by using a preset rule.

The preset rule may be construed as the first preset rule and the second preset rule in the foregoing embodiments.

Specifically, determining the route according to the structural information and the logical information by using the preset rule may be construed as determining whether the structural information and the logical information of the SQL statement matches the preset rule. If yes, step 412 is executed. If not, step 408 is executed.

Step 408: Optimize a physical execution plan using a cost-based optimizer.

Specifically, optimizing the physical execution plan using the cost-based optimizer is formulating the physical execution plan by the cost-based optimizer for the SQL statement to be executed. The physical execution plan includes how to actually execute the SQL statement.

Step 410: Determine a route according to the cost information by using a cost rule.

Specifically, a physical execution cost of executing the SQL statement is calculated according to the physical execution plan formulated for the SQL statement. For the physical execution cost, reference may be made to the foregoing embodiments, and the details will not be repeated herein. Then, a database to which the SQL statement needs to be routed is determined according to the calculated physical execution cost.

Step 412: Routing result.

Specifically, when the structural information and the logical information of the SQL statement match the preset rule, it may be determined that the SQL statement needs to be routed to the distributed data node of the read-write database. When the structural information and the logical information of the SQL statement do not match the preset rule, whether the SQL statement needs to be routed to a read-write database or a read-only database may be determined according to the physical execution cost of the SQL statement, so as to realize the read-write separation of the SQL statements. Therefore, finally, requests of the OLTP type are routed to an RW (read-write) node, and requests of the OLAP type are routed to an RO (read-only) node, thereby ensuring stable RT and throughput for highly concurrent SQL requests of the OLTP type, without being affected by SQL requests of the OLAP type that consume a lot of resources, thereby ensuring the stability of the database.

The data processing method provided in the embodiments of this specification designs a read-write separation routing algorithm that can intelligently identify OLTP and OLAP workloads, is used for read-write separation to prevent OLAP workload traffic from affecting OLTP workload traffic, thereby ensuring low latency and high throughput of OLTP workload traffic. Therefore, the problem of mutual interference between OLAP workloads and OLTP workloads can be well solved without manual intervention.

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

- a processing request receiving module 502, configured to receive a data processing request carrying a query statement;

- an operation type determination module 504, configured to convert the query statement into a corresponding relational algebra tree based on the data processing request, and determine an operation type corresponding to the query statement based on the relational algebra tree; and

- a database determination module 506, configured to deliver the query statement to a first database in response to the operation type being a first type, and complete the data processing request in the first database based on the query statement.

Optionally, the operation type determination module 504 is further configured to execute operations of:

- parsing the query statement into a corresponding abstract syntax tree, and converting the abstract syntax tree into the corresponding relational algebra tree, where the relational algebra

tree consists of tree nodes and edges.

Optionally, the operation type determination module 504 is further configured to execute operations of:

- determining structural information of the query statement based on the relational algebra tree; and

- determining that the operation type corresponding to the query statement is a first type in response to the structural information of the query statement satisfying a first preset rule.

Optionally, the operation type determination module 504 is further configured to execute operations of:

- determining logical information of the query statement based on the relational algebra tree; and

- determining that the operation type corresponding to the query statement is a first type in response to the logical information of the query statement satisfying a second preset rule.

Optionally, the apparatus further includes:

- a logical information determination module, configured to logical information of the query statement based on the relational algebra tree in response to the structural information of the query statement not satisfying the first preset rule; and

- a transaction type determination module, configured to determine that the operation type corresponding to the query statement is the first type in response to the logical information of the query statement satisfying a second preset rule.

Optionally, the operation type determination module 504 is further configured to execute operations of:

- determining structural information and logical information of the query statement based on the relational algebra tree;

- determining an execution plan of the query statement based on the relational algebra tree and calculating a physical execution cost of the query statement according to the execution plan, in response to the structural information of the query statement not satisfying a first preset rule and the logical information of the query statement not satisfying a second preset rule; and

- determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold.

Optionally, the operation type determination module 504 is further configured to execute operations of:

- determining an execution plan of the query statement based on the relational algebra tree and calculating a physical execution cost of the query statement according to the execution

plan; and

determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold.

Optionally, the operation type determination module 504 is further configured to execute operations of:

determining tree nodes of the relational algebra tree and determining the execution plan of the query statement based on the tree nodes;

calculating a physical execution cost of each of the tree nodes corresponding to the query statement based on the execution plan of the query statement; and

performing a weighted summation of the physical execution costs of the tree nodes to obtain the physical execution cost of the query statement.

Optionally, the operation type determination module 504 is further configured to execute operations of:

determining that the operation type corresponding to the query statement is the first type in response to the physical execution cost of any of the tree nodes corresponding to the query statement being less than or equal to the preset cost threshold.

Optionally, the apparatus further includes:

a second type determination module, configured to determine that the operation type corresponding to the query statement is a second type in response to the physical execution cost being greater than the preset cost threshold.

Optionally, the apparatus further includes:

a data processing module, configured to deliver the query statement to a second database in response to the operation type corresponding to the query statement being the second type, and complete the data processing request in the second database based on the query statement.

Optionally, the first database is a distributed read-write database for processing query statements of the first type, and the second database is a distributed read-only database for processing query statements of the second type.

Optionally, the first type is an OLTP type; and

correspondingly, the database determination module is further configured to execute:

delivering the query statement to the distributed read-write database in response to the operation type being the OLTP type.

Optionally, the second type is an OLAP type; and

correspondingly, the data processing module is further configured to execute:

delivering the query statement to the distributed read-only database in response to the

operation type corresponding to the query statement being the OLAP type.

An embodiment of this specification implements a data processing apparatus, which can intelligently identify whether an operation type corresponding to an SQL statement sent by a client is an OLTP type or an OLAP type, and only when the operation type corresponding to the SQL statement is the OLTP type, deliver the SQL statement to a primary instance (that is, a read-write database) for processing, thereby ensuring low latency and high throughput of OLTP workload traffic and ensuring the stability of the database.

The above is a schematic solution of a data processing apparatus according to this embodiment. It should be noted that the technical solution of the data processing apparatus and the technical solution of the foregoing data processing method belong to the same concept, and for the details that are not set forth in the technical solution of the data processing apparatus, reference may be made to the description of the technical solution of the foregoing data processing method.

Referring to FIG. 6, FIG. 6 is a schematic structural diagram of a data processing system according to an embodiment of this specification.

The data processing system includes: a routing module 602, a read-write distributed data node 604, and a read-only distributed data node 606, where,

the routing module 602 is configured to receive a data processing request carrying a query statement, and determine an operation type corresponding to the query statement based on the data processing request;

deliver the query statement to the read-write distributed data node 604 in response to the operation type being an OLTP type, and complete the data processing request at the read-write distributed data node 604 based on the query statement; and

deliver the query statement to the read-only distributed data node 606 in response to the operation type being an OLAP type, and complete the data processing request at the read-only distributed data node 606 based on the query statement.

In the data processing system provided in the embodiments of this specification, after receiving a data processing request carrying a query statement sent by a client, the routing module 602 converts the query statement into a corresponding relational algebra tree based on the data processing request, and determines an operation type corresponding to the query statement based on the relational algebra tree; and delivers the query statement to the read-write distributed data node 604 in response to the operation type being an OLTP type, and completes the data processing request at the read-write distributed data node 604 based on the query statement; or delivers the query statement to the read-only distributed data node 606 in

response to the operation type being an OLAP type, and completes the data processing request at the read-only distributed data node 606 based on the query statement. The routing module 602 can intelligently identify whether an operation type corresponding to a query statement sent by a client is an OLTP type or an OLAP type, and only when the operation type corresponding to the query statement is the OLTP type, deliver the query statement to a primary instance for processing, thereby ensuring low latency and high throughput of OLTP workload traffic and ensuring the stability of the database.

FIG. 7 is a schematic block diagram of a computing device 700 according to an embodiment of this specification. Components of the computing device 700 include, but are not limited to, a memory 710 and a processor 720. The processor 720 and the memory 710 are connected through a bus 730. A database 750 is configured to store data.

The computing device 700 further includes an access device 740. The access device 740 is configured to enable the computing device 700 to communicate via one or more networks 760. Examples of the networks include a public switched telephone network (PSTN), a local area network (LAN), a wide area network (WAN), a personal area network (PAN), and/or a combination of communication networks such as the internet. The access device 740 may include one or more of wired or wireless network interfaces (for example, a network interface card (NIC)) of any types, such as an IEEE 802.11 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. 7 of the computing device 700 may be connected, for example, through the bus. It should be understood that the structural block diagram of the computing device shown in FIG. 7 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 700 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 700 may alternatively be a mobile or stationary server.

The processor 720 is configured to execute the following computer-executable instructions:

receiving a data processing request carrying a query statement;

converting the query statement into a corresponding relational algebra tree based on the data processing request, and determining an operation type corresponding to the query statement based on the relational algebra tree; and

delivering the query statement to a first database in response to the operation type being a first type, and completing the data processing request in the first database based on the query statement.

The above is a schematic solution of a computing device according to this embodiment. It should be noted that the technical solution of the computing device and the technical solution of the foregoing data processing method belong to the same concept, and for the details that are not set forth in the technical solution of the computing device, reference may be made to the description of the technical solution of the foregoing data processing method.

An embodiment of this specification further provides a computer-readable storage medium, storing computer-executable instructions, where the computer-executable instructions, when executed by a processor, implement operations of:

receiving a data processing request carrying a query statement sent by a client;

converting the query statement into a corresponding relational algebra tree based on the data processing request, and determining an operation type corresponding to the query statement based on the relational algebra tree; and

delivering the query statement to a first database in response to the operation type being a first type, and completing the data processing request in the first database based on the query statement.

The above is a schematic solution of a 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 foregoing data processing method belong to the same concept, and for the details that are not set forth in the technical solution of the storage medium, reference may be made to the description of the technical solution of the foregoing data processing method.

Specific embodiments of this specification are described above. Other embodiments are within the scope of the appended claims. In some cases, the actions or steps recited in the claims may be performed in a different order than in the embodiments and the desired result may still be achieved. In addition, the processes depicted in the accompanying drawings do not necessarily require the particular order shown or sequential order to achieve the desired results. In some embodiments, multitasking and parallel processing are also possible or may be advantageous.

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.

The foregoing method embodiments are expressed as a series of action combinations for the purpose of brief description, but it is to be understood by a person skilled in the art that because some steps may be performed in other sequences or simultaneously according to the embodiments of this specification, the embodiments of this specification are not limited to a described action sequence. Secondly, those skilled in the art should also know that the embodiments described in the specification are all preferred embodiments, and the actions and modules involved are not necessarily required by 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 subject only to the claims and the full scope and equivalents thereof.

## CLAIMS

### What is claimed is:

[Claim 1] A data processing method, comprising:  
receiving a data processing request carrying a query statement;  
converting the query statement into a corresponding relational algebra tree based on the data processing request, and determining an operation type corresponding to the query statement based on the relational algebra tree; and  
delivering the query statement to a first database in response to the operation type being a first type, and completing the data processing request in the first database based on the query statement.

[Claim 2] The data processing method according to claim 1, wherein the converting the query statement into a corresponding relational algebra tree includes:  
parsing the query statement into a corresponding abstract syntax tree, and converting the abstract syntax tree into the corresponding relational algebra tree, wherein the relational algebra tree consists of tree nodes and edges.

[Claim 3] The data processing method according to claim 2, wherein the determining an operation type corresponding to the query statement based on the relational algebra tree includes:  
determining structural information of the query statement based on the relational algebra tree; and  
determining that the operation type corresponding to the query statement is a first type in response to the structural information of the query statement satisfying a first preset rule.

[Claim 4] The data processing method according to claim 2, wherein the determining an operation type corresponding to the query statement based on the relational algebra tree includes:  
determining logical information of the query statement based on the relational algebra tree;  
and  
determining that the operation type corresponding to the query statement is a first type in response to the logical information of the query statement satisfying a second preset rule.

[Claim 5] The data processing method according to claim 3, wherein after the determining structural information of the query statement based on the relational algebra tree, the method further comprises:  
determining logical information of the query statement based on the relational algebra tree in response to the structural information of the query statement not satisfying the first preset rule; and  
determining that the operation type corresponding to the query statement is the first type in response to the logical information of the query statement satisfying a second preset rule.

[Claim 6] The data processing method according to claim 2, wherein the determining an operation type corresponding to the query statement based on the relational algebra tree includes:  
determining structural information and logical information of the query statement based on the relational algebra tree;  
determining an execution plan of the query statement based on the relational algebra tree

and calculating a physical execution cost of the query statement according to the execution plan, in response to the structural information of the query statement not satisfying a first preset rule and the logical information of the query statement not satisfying a second preset rule; and determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold.

[Claim 7] The data processing method according to claim 2, wherein the determining an operation type corresponding to the query statement based on the relational algebra tree includes:

- determining an execution plan of the query statement based on the relational algebra tree and calculating a physical execution cost of the query statement according to the execution plan; and

- determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold.

[Claim 8] The data processing method according to claim 6 or 7, wherein the determining an execution plan of the query statement based on the relational algebra tree and calculating a physical execution cost of the query statement according to the execution plan includes:

- determining tree nodes of the relational algebra tree and determining the execution plan of the query statement based on the tree nodes;

- calculating a physical execution cost of each of the tree nodes corresponding to the query statement based on the execution plan of the query statement; and

- performing a weighted summation of the physical execution costs of the tree nodes to obtain the physical execution cost of the query statement.

[Claim 9] The data processing method according to claim 8, wherein the determining that the operation type corresponding to the query statement is a first type in response to the physical execution cost being less than or equal to a preset cost threshold includes:

- determining that the operation type corresponding to the query statement is the first type in response to the physical execution cost of any of the tree nodes corresponding to the query statement being less than or equal to the preset cost threshold.

[Claim 10] The data processing method according to claim 8, further comprising:

- determining that the operation type corresponding to the query statement is a second type in response to the physical execution cost being greater than the preset cost threshold.

[Claim 11] The data processing method according to claim 10, wherein after the determining that the operation type corresponding to the query statement is a second type, the method further comprises:

- delivering the query statement to a second database in response to the operation type corresponding to the query statement being the second type, and completing the data processing request in the second database based on the query statement.

[Claim 12] The data processing method according to claim 11, wherein the first database is a distributed read-write database for processing query statements of the first type, and the second database is a distributed read-only database for processing query statements of the second type.

[Claim 13] The data processing method according to claim 12, wherein the first type is an on-line transaction processing (OLTP) type; and

correspondingly, the delivering the query statement to a first database in response to the operation type being a first type includes:

delivering the query statement to the distributed read-write database in response to the operation type being the OLTP type.

[Claim 14] The data processing method according to claim 13, wherein the second type is an on-line analytical processing (OLAP) type; and

correspondingly, the delivering the query statement to a second database in response to the operation type corresponding to the query statement being the second type includes:

delivering the query statement to the distributed read-only database in response to the operation type corresponding to the query statement being the OLAP type.

[Claim 15] A data processing apparatus, including:

a processing request receiving module, configured to receive a data processing request carrying a query statement;

an operation type determination module, configured to convert the query statement into a corresponding relational algebra tree based on the data processing request, and determine an operation type corresponding to the query statement based on the relational algebra tree; and

a database determination module, configured to deliver the query statement to a first database in response to the operation type being a first type, and complete the data processing request in the first database based on the query statement.

[Claim 16] A data processing system, comprising: a routing module, a read-write distributed data node, and a read-only distributed data node, wherein,

the routing module is configured to receive a data processing request carrying a query statement, and determine an operation type corresponding to the query statement based on the data processing request;

deliver the query statement to the read-write distributed data node in response to the operation type being an on-line transaction processing (OLTP) type, and complete the data processing request at the read-write distributed data node based on the query statement; and

deliver the query statement to the read-only distributed data node in response to the operation type being an on-line analytical processing (OLAP) type, and complete the data processing request at the read-only distributed data node based on the query statement.

[Claim 17] 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 computer-executable instructions, when executed by the processor, implement the operations of the data processing method according to any one of claims 1 to 14.

[Claim 18] A computer-readable storage medium, storing computer instructions, wherein the computer instructions, when executed by a processor, implement the operations of the data processing method according to any one of claims 1 to 14.