# Notice



## DESCRIPTION CN102917072A

Device, system and method for data migration between data server clusters

### [0001]

Technical Field

### [0002]

The present invention relates to the field of data storage technology, and in particular to a migration device, system and method for migrating data between data server clusters.

### [0003]

Background Art

### [0004]

MongoDB (Data Base) is a product between relational databases and non-relational databases. It is the most feature-rich non-relational database and the most relational database-like.
The data structure it supports is very loose, so it can store more complex data types.
Since MongoDB itself has good performance, in the early stages of business development, developers will deploy multiple small businesses on a small Mongodb cluster consisting of a small number of servers.
As the business gradually develops, the traffic increases. For example, the traffic of one or several existing businesses grows rapidly, forming a large-scale business. As a result, a small

cluster consisting of a small number of servers can no longer meet business needs. At this time, you need to consider how to increase system capacity to solve performance problems.

## [0005]

The first solution at present is that based on the dynamic expansion supported by Mongodb itself, performance can be improved simply by adding data servers. Therefore, the performance problem can be solved by directly adding Mongod servers to the Mongodb cluster currently composed of a small number of data servers.
By adding new Mongod servers and turning it into a cluster of more data servers, the performance of MongoDB will basically improve linearly. However, this solution also has side effects. Because the existing data server cluster serves multiple businesses, and there may be both small-scale businesses and rapidly growing large-scale businesses among the multiple businesses, and in the process of accessing the Mongod server, large-scale businesses may occupy the access resources of the Mongod server for a long time, and small-scale businesses are bound to be unable to compete with large-scale businesses, which will eventually cause these small-scale businesses to be affected by the rapidly growing large-scale businesses.

## [0006]

In order to avoid the side effects of the first solution mentioned above, the second solution gradually emerged.
That is, consider migrating the expanded business from the original MongoDB cluster to a new MongoDB cluster. For larger businesses, use a separate new cluster instead of sharing the Mongod server with other businesses. In this way, the existing multiple smaller businesses are still in the original data server cluster, while the rapidly developing larger-scale businesses use a new data server cluster alone. Therefore, the larger-scale businesses will no longer compete with the smaller-scale businesses for server access resources.

## [0007]

During the migration process using the above solution, the business first stops all write operations to the mongod server, then backs up the existing mongodb database information and imports it into the new data server cluster.
Finally, after the business data is completely switched to the new data server cluster, enable the business to write to the mongod server. Since all business write services need to be stopped during the switching process, and when the data volume is large, the downtime backup migration process may take several hours. Therefore, the normal operation of

related businesses will be greatly affected during this period of downtime backup migration, affecting the normal service provided to users. Similarly, similar problems also exist in other non-MongoDB data server cluster application environments.

## [0008]

Summary of the invention

## [0009]

In view of the above problems, the present invention is proposed to provide a migration device for performing data migration between data server clusters and a corresponding migration device and method for performing data migration between data server clusters that overcome the above problems or at least partially solve the above problems.

## [0010]

According to one aspect of the present invention, there is provided a migration device for migrating data between data server clusters, wherein the data is target data related to the business to be migrated, and comprises: an initial data import module, configured to import the target data stored in the first data server cluster and written at a first time point and before the first time point into the second data server cluster; a synchronization module, configured to obtain an operation log associated with the target data written to the first data server cluster after the first time point, the operation log at least including the content of the target data written to the first data server cluster; and an update module, configured to update the target data in the second data server cluster according to the operation log obtained by the synchronization module.

## [0011]

Optionally, the initial data import module includes: a first initial data import sub-module, configured to import the target data stored at the first time point in the first data server cluster and written before the first time point into a storage medium; and a second initial data import sub-module, configured to import the target data imported into the storage medium into the second data server cluster.

## [0012]

Optionally, the first initial data import submodule is configured to back up the target data stored in the first data server cluster at the first time point and written before the first time point to a storage medium through mongodump; and the second initial data import

submodule is configured to import the target data backed up to the storage medium into the second data server cluster through mongorestore.

## [0013]

Optionally, the first data server cluster includes a master data server and several slave data servers, and the migration device also includes: a deactivation processing module, configured to stop the write operation of the first slave data server in the first data server cluster after a first time point; and an initial data import module, configured to import the target data stored in the first slave data server of the first data server cluster at the first time point and written before the first time point into the second data server cluster.

## [0014]

Optionally, it also includes: a synchronization detection module, configured to detect whether the target data of the first data server cluster and the second data server cluster are synchronized; an address update module, configured to change the entry address of the connection data server from the entry address of the first data server cluster to the entry address of the second data server cluster after the synchronization detection module detects that the target data of the first data server cluster and the second data server cluster are synchronized.

## [0015]

Optionally, the operation log also includes one or more of the following information: a timestamp of writing the data; and when the written data is an update of the original data, the old value of the data before the update.

## [0016]

Optionally, the first data server cluster is a first MongoDB cluster, the second data server cluster is a second MongoDB cluster, and the operation log is an oplog in MongoDB.

## [0017]

According to another embodiment of the present invention, a system for performing data migration between data server clusters is provided. The system includes at least a first data server cluster and a second data server cluster, and the migration device described above.

## [0018]

According to another embodiment of the present invention, a method for migrating data between data server clusters is provided, wherein the data is target data related to the business to be migrated, and comprises: importing the target data stored in the first data server cluster and written at and before a first time point into the second data server cluster; obtaining an operation log associated with the target data written to the first data server cluster after the first time point, the operation log at least including the content of the target data written to the first data server cluster; and updating the target data in the second data server cluster according to the obtained operation log.

## [0019]

According to the migration device, system and method of the present invention, on the one hand, the target data before a certain time point is directly backed up to the new data server cluster by means of backup, and on the other hand, the target data written to the old data server cluster after the first time point is synchronously written to the second data server cluster by means of operation logs, so that the new and old data server clusters basically achieve synchronization of the target data, and then the subsequent migrated business can directly connect to the new data server cluster to write and read data, and there is no need to stop the business to be migrated during this process, thereby solving the existing problem that the business data migration must be achieved by stopping the backup, and achieving the beneficial effect of completing the business data migration without affecting the normal external service of the migrated business.

## [0020]

The above description is only an overview of the technical solution of the present invention. In order to more clearly understand the technical means of the present invention, it can be implemented according to the contents of the specification. In order to make the above and other purposes, features and advantages of the present invention more obvious and easy to understand, the specific implementation methods of the present invention are listed below.

## [0021]

BRIEF DESCRIPTION OF THE DRAWINGS

## [0022]

Various additional advantages and benefits will become apparent to those of ordinary skill in the art from a reading of the following detailed description of the preferred embodiments.

The drawings are only for the purpose of illustrating the preferred embodiments and are not to be construed as limiting the invention.
Also, throughout the drawings, the same reference symbols are used to denote the same parts.
In the attached picture:

## [0023]

FIG1 shows a first system schematic diagram for performing data migration between data server clusters according to an embodiment of the present invention;

## [0024]

FIG2 shows a second system schematic diagram for performing data migration between data server clusters according to an embodiment of the present invention; and

## [0025]

FIG3 shows a flow chart of a method for performing data migration between data server clusters according to an embodiment of the present invention.

## [0026]

DETAILED DESCRIPTION

## [0027]

Exemplary embodiments of the present disclosure will be described in more detail below with reference to the accompanying drawings.
Although exemplary embodiments of the present disclosure are shown in the drawings, it should be understood that the present disclosure can be implemented in various forms and should not be limited to the embodiments set forth herein.
On the contrary, these embodiments are provided to enable a more thorough understanding of the present disclosure and to fully convey the scope of the present disclosure to those skilled in the art.

## [0028]

Please refer to FIG. 1 , which is a schematic diagram of a first system for performing data migration between data server clusters according to an embodiment of the present invention.

The system includes a migration device 100 for performing data migration between data server clusters, a first data server cluster 200, and a second data server cluster 300.
The migration device 100 includes an initial data import module 102 , a synchronization module 104 and an update module 106 .
The first data server cluster 200 includes multiple data servers, among which only the first data server 202 and the second data server 204 are schematically shown in the figure. In actual application, more data servers may be included as needed, and the present invention has no limitation to this.
Similarly, the second data server cluster 300 also includes multiple data servers, of which a third data server 302, a fourth data server 304 and a fifth data server 306 are schematically shown in the figure.

## [0029]

Generally speaking, the data servers in each data server cluster back up each other's data. Generally, one of the multiple data servers is a master data server, and the rest are slave data servers. In most cases, the application server 400 directly writes data to the master data server, and the other slave data servers generally do not directly accept data written by the application server 400, but back up data from the master data server.
The following is a detailed explanation of the data processing process of each component and the relationship between the components.
For the convenience of subsequent description, the business data that needs to be migrated from the first data server cluster 200 to the second data server cluster 300 is referred to as target data related to the business to be migrated.

## [0030]

In one embodiment, before the target data is migrated, the application server 400 writes the data of the business to be migrated, that is, the target data, into the first data server cluster 200 at any time.
For example, the first data server 202 and the second data server 204 in the first data server cluster 200 both store the target data that has been written.
Furthermore, when preparing for data migration, first select a time point that is already in the past, referred to as the first time point, and then the initial data import module 102 imports the target data written by the application server 400 to the first data server cluster 200 at the first time point and before the first time into the second data server 300.
For the sake of convenience in the following description, the second data server in the first data server 200 is taken as the master data server and the first data server is taken as the slave data server.

## [0031]

Specifically, first, the initial data export module 102 selects a data server from multiple data servers in the first data server cluster 200 to export data. Usually, a slave data server can be selected to export data so that during the data export process, the subsequent writing of the application server 400 to the main data server is not affected. For example, the first data server 202 as a slave data server is selected to export data.
In addition, since the application server 400 does not directly write data to the first data server 202 as the slave data server, but the first data server 202 obtains data from the second data server 204 after the application server 400 writes data to the second data server 204 as the master data server, and since the amount of target data exported from the first data server 202 and written to the first data server cluster 200 before the first time point may be relatively large, in order to better avoid various unexpected situations that may occur when data is exported and written simultaneously in the first data server 202, the migration device 100 may include a deactivation processing module, which is used to stop the write operation of the first data server 202 after the first time point, and then start the data export operation.
After the initial data import module 102 successfully exports the data written to the first data server cluster 200 at the first time point and before the first time from the first data server 202, the write operation of the first data server 202 can be resumed.
It should be noted that the above example of stopping the write operation of the first data server 202 before exporting the target data is only optional, and it is entirely possible to continue synchronizing data from the second data server 204 to the first data server 202 while exporting the target data.

## [0032]

Then, the initial data import module 102 imports the target data stored in the first data server 202 and written at the first time point and before the first time point into the second data server cluster 300.
For example, the initial data import module 102 specifically includes a first initial data import sub-module and a second initial data import sub-module. The first initial data import sub-module first retrieves the target data written at and before the first time point from the first data server 202, and imports the target data into a storage medium, such as a disk (for example, in the form of a file).
Then, the second initial data import submodule imports the target data imported into the storage medium into each data server in the second data server cluster 300 (302-306).

In the process of the initial data import module 102 importing the target data into the second data server cluster 300, the target data can be first imported into the master data server in the second data server cluster 300, and then other slave data servers obtain the target data from the master data server. In this way, the master data server and the slave data servers in the second data server cluster 300, that is, the third data server 302, the fourth data server 304 and the fifth data server 306, all successfully obtain the target data written into the first data server cluster 100 at the first time point and before the first time point.

## [0033]

During the process in which the initial data import module 102 successfully imports the target data written into the first data server cluster 200 at the first time point and before the first time point into the second data server cluster 300 and after the successful import, the application server 400 does not stop writing data to the second data server 204 serving as the main data server in the first data server cluster 200, so after the first time point, there is still target data written into the first data server cluster 200.
Specifically, the target data after the first time point is first written into the second data server 204 by the application server 400, and then the first data server 202 can synchronously obtain the target data written after the first time point from the second data server 204.

## [0034]

Moreover, each time the application server 400 writes a piece of target data to the second data server 204, based on the characteristics of the data server cluster itself, the data server cluster will simultaneously generate an operation log (oplog) associated with the target data, and the content of the specific data written each time will be recorded in the operation log. Moreover, the general operation log also records the time information of each written data, that is, the timestamp. If the written data is an update of the previous data, not only the new value after the data update will be recorded, but also the old value before the data update will be recorded.
In other words, according to the operation log, it is possible to know at what time and what content data was written by the application server 400.

## [0035]

Furthermore, the synchronization module 104 in the migration device 100 can obtain the operation log associated with the target data written to the first data server cluster 200 after the first time point.
Specifically, since the application server 400 writes target data to the first data server cluster 200, data is mostly written one by one, and accordingly, operation logs are generated one by one, and the operation logs include the timestamp of the written data.
Therefore, optionally, the synchronization module 104 can obtain in real time the operation log associated with the target data written to the first data server cluster 200 after the first time point.
Of course, it is not necessary to obtain the operation log in real time, but to obtain the operation log once every certain time interval. However, in order to achieve the fastest synchronization of the two data server clusters, the synchronization module 104 obtains the operation log at a time interval as short as possible.

## [0036]

Then, the synchronization module 104 provides the obtained operation log associated with the target data written to the first data server cluster 200 after the first time point to the update module 106.
After the update module 106 obtains the operation log associated with the data written into the first data server cluster 200 after the first time point, the target data in the second data server cluster 300 can be updated according to the obtained operation log.
As mentioned before, the operation log at least includes the content of specific data written by the application server 400 to the first data server cluster 200 each time after the first time point. Therefore, the update module 106 can update the corresponding target data in the second data server cluster 300 according to each operation log, so that the second data server cluster 300 also stores the target data written to the first data server cluster 200 after the first time point.

## [0037]

Since a single operation log involves very little data update content, the update module 106 can update the associated target data in the second data server cluster 300 in a timely manner based on an operation log. Furthermore, it can basically be achieved that every time the application server 400 writes one or several target data to the first data server cluster 200 after the first time point, the update module 106 can correspondingly write the same target data to the second data server cluster 300. Therefore, after a very short period of time, the second data server 300 can achieve the purpose of consistency and synchronization with the target data in the first data server 200.

## [0038]

At this point, all target data related to the business to be migrated stored in the first data server cluster 200 has been migrated to the second data server cluster 300.

## [0039]

In another embodiment, after the target data is successfully migrated, the subsequent application server 400 can directly write data related to the business to be migrated to the second data server cluster 300, and the migration device 100 can also include a synchronization detection module and an address update module.
Specifically, after the update module 106 updates the target data in the second data server cluster 300 according to the operation log obtained by the synchronization module 104 and associated with the target data written to the first data server cluster 200 after the first time point, the synchronization detection module can compare the target data in the first data server cluster 200 and the second data server cluster 300 to detect whether the target data in the two data server clusters have been successfully synchronized.
To detect whether the synchronization is successful, in addition to referring to the content of the target data in the two data server clusters, you can also refer to the operation log associated with the target data. This is because the operation log generally contains auxiliary information such as the timestamp when the target data is written and the values before and after the target data is updated. Therefore, the synchronization detection module can also refer to this auxiliary information to more quickly and accurately detect whether the two data server clusters have been successfully synchronized.

## [0040]

After the synchronization detection module determines that the first data server cluster 200 and the second data server cluster 300 have been synchronized successfully, the address update module can be notified that the two data server clusters have been synchronized successfully, and then the address update module can change the entry address of the application server 400 connecting to the data server from the entry address of the first data server cluster 200 to the entry address of the second data server cluster 300.
If the application server 400 subsequently needs to write data related to the business to be migrated to the data server cluster or read data related to the business, it will directly write and read data to the second data server cluster 300 because the entry address for connecting to the data server has been changed to the entry address of the second data server cluster.

At this point, the business that needs to be migrated has been successfully migrated from the first data server cluster 200 to the second data server cluster 300.
Then, the data related to the migrated business in the first data server cluster 200 can be deleted.

## [0041]

It should be noted that, in the specific implementation process, the migration device 100 can be implemented independently of the first data server cluster 200 and the second data server cluster 300, or it can be implemented in a certain data server cluster, such as the data management server of the second data server cluster 300.
The data management server can be a server that is currently available in many data server clusters and plays a role in cluster management, such as mangos in a MongoDB cluster.

## [0042]

In one embodiment, the first data server cluster 200 of the first system above is a first MongoDB cluster 500, and the second data server cluster 300 is a second MongoDB cluster 600, wherein the migration device 100 is placed in the mangos of the second MongoDB cluster for implementation.
Please refer to FIG. 2 for details, which is a schematic diagram of a second system for performing data migration between data server clusters according to an embodiment of the present invention. The second system can be understood as a specific application example of the above first system applied to a data server cluster of the MongoDB type. Therefore, the first MongoDB cluster 500 in the second system is similar to the first data server cluster 200 in the first system. Similarly, the second MongoDB cluster 600 in the second system is similar to the second data server cluster 300 in the first system. The initial data import module 702 in the second system is similar to the initial data import module 102 in the first system. The synchronization module 704 in the second system is similar to the synchronization module 104 in the first system, and the update module 706 in the second system is similar to the update module 106 in the first system. Therefore, the specific implementation of each component in the second system is basically not repeated, and reference can be made to the specific implementation of the relevant components in the first system. Only individual components related to specific MongoDB cluster features are described.

## [0043]

For example, the first initial data import submodule in the initial data import module 702 can first call mongodump, a backup tool provided by mongo, to back up the target data stored at

the first time point and before the first time point in the first mongod cluster 500 to the disk to generate a data file.
Then, the second initial data submodule in the initial data import module 702 uses mongorestore, a recovery tool provided by mongo, to import the data file into the second mongod cluster 600 . The operation log obtained by the synchronization module 704 is specifically the oplog in MongoDB. For example, the content of a specific oplog instance is as follows:

## [0044]

{″ts":{″t":1339660240000,"i″:8},

## [0045]

"h":NumberLong("-79360722582655136 67"),"op":"i","ns":"test.method",

## [0046]

"o":{″_id":"testid","v":"test"}}

## [0047]

Among them, "ts" records the timestamp of the operation; "o p" records the type of the operation, for example, type "i" indicates an insert operation; "h" records the hash value of this oplog, and "ns" records the namespace of the operation; "o" records the file content, that is, the content of the specific data written.

## [0048]

It can be seen from the above oplog example that the oplog of the MongoDB cluster not only contains the content of the written target data, but also includes other auxiliary information such as timestamps. Therefore, the update module 706 can obtain the operation log associated with the target data written to the first MongoDB cluster 500 after the first time point according to the synchronization module 704, and update the content of the target data in the second MongoDB cluster 600.

## [0049]

Those skilled in the art can understand that in other distributed data storage systems other than MongoDB clusters, such as Cassandra and other distributed data storage systems, there are similar problems of data migration between data server clusters, and there are

also operation logs similar to the operation logs of MongoDB clusters. Therefore, the technical solution of the present invention is not only applicable to data migration between MongoDB clusters, but also to data migration between other types of data server clusters.

## [0050]

Please refer to Figure 3, which is a schematic diagram of a method for migrating data between data server clusters according to an embodiment of the present invention. The data is target data related to the business to be migrated. The two data server clusters performing data migration may be, for example, the first data server cluster 200 and the second data server cluster 300 described in Figure 1 above, or the first MongoDB cluster 500 and the second MongoDB cluster 600 described in Figure 2 above.

## [0051]

The data migration method begins with step S310. In step S310, target data stored in the first data server cluster and written at a first time point and before the first time point is imported into the second data server cluster.
Specifically, when preparing for data migration, first select a first time point that is in the past, and then import the first time point stored in the first data server cluster and the target data written before the first time point into the second data server cluster.
For example, the target data in the first data server cluster can be backed up to a storage medium such as a disk. For example, in a MongoDB cluster, the backup tool mongodump can be used to import the data. Then, the target data in the storage medium can be imported into the second data server cluster. For example, in a MongoDB cluster, the recovery tool mongorestore can be used.
Optionally, if there are a master data server and slave data servers in the first data server cluster, it is preferred to export the target data from a slave data server after stopping the write operation of the slave data server.
This step can be performed by the initial data import module 102 in Figure 1 or the initial data import module 702 in Figure 2. The relevant technical implementation can refer to the relevant description of the initial data import module in each embodiment, which will not be repeated here.

## [0052]

In the above step S310, the target data backed up from the first data server cluster to the second data server cluster is written to the first data server cluster at and before the first time point, so what needs to be migrated subsequently is the target data written to the first data server cluster after the first time point.
Furthermore, in step S320, the operation log associated with the target data written to the first data server cluster after the first time point is first obtained. The operation log is an information record that records the specific content of each data writing operation, including the content of the target data written to the first data server cluster each time, and usually also includes a timestamp of the write. If the operation is an update of the data content written previously, then the operation log records not only the new value after the update, but also the old value before the update.
It can be seen that the specific content of each written data can be known based on the operation log.
In the process of obtaining operation logs, since data is written one by one and operation logs are generated one by one, the operation logs can be obtained in real time, that is, each time one or several operation logs are written, they are obtained once; the operation logs can also be obtained periodically.
Step S320 may be executed by the synchronization module 104 in FIG. 1 or the synchronization module 704 in FIG. 2 . For the related technical implementation, reference may be made to the related description of the synchronization module in each embodiment, which will not be repeated here.

## [0053]

Then, in step S330, the target data in the second data server cluster is updated according to the operation log written to the first data server cluster after the first time point obtained in step S320.
Specifically, the corresponding data can be written to the second data server cluster according to the specific content of each data write recorded in the operation log, so that the second data server cluster also successfully stores the target data written to the first data server cluster after the first time point.
Since the content of each operation log is relatively small, the speed at which the update operation is completed in step S330 is very fast, and almost every time a target data is written to the first data server cluster, it is also written to the second data server cluster. At this point, all the target data in the first data server cluster has been migrated to the second data server cluster, and the purpose of synchronizing the target data in the two data server clusters has been achieved. Step S330 may be executed by the update module 106 in FIG. 1 or the update module 706 in FIG. 2 . For the related technical implementation, reference may

be made to the related description of the update module in each embodiment, which will not be described in detail here.

## [0054]

After that, it is also possible to detect whether the target data stored in the two data server clusters are consistent, and use auxiliary information such as the timestamp in the operation log and the numerical content before and after the update to determine whether the two data server clusters have been successfully synchronized.
This step may be performed by the synchronization detection module described in the migration device 100 above. If it is determined after detection that the synchronization has been successful, then the entry address for connecting to the data server can be changed from the entry address of the first data server cluster to the entry address of the second data server cluster. This step can be performed by the address update module in the migration device mentioned above. Thereafter, no matter whether the application server needs to write to the target data or read the target data, the second data server cluster will perform the corresponding operation. After observing that the operation of the two data servers is completely stable for a period of time, the target data in the first data server cluster can be deleted.

## [0055]

It should be noted that the order of the steps in the above method can be adjusted. For example, step S320 does not have to wait until step S310 is completely executed before starting. It is possible that these two steps are executed simultaneously or partially simultaneously.
Since the operation log is generated in real time, step S320 and step S330 can also be repeatedly executed, that is, each time one or more operation logs are newly generated, step S320 and step S330 are executed once, so as to obtain the latest operation log in time and update the target data in the second data server cluster in time according to the operation log, so that the two data servers can achieve synchronous writing of the target data as soon as possible. For another example, the synchronization detection step mentioned above can also be repeatedly executed. For example, if it is found that the synchronization is not successful after the first execution, it can be detected again after a period of time until the two data server clusters are detected to be synchronized.

## [0056]

It can be seen from the description of the above embodiments that by adopting the technical solution provided by the embodiments of the present invention, in the process of migrating the target data at the first time point and before the first time point, which may have a large amount of data, it is not necessary to stop the normal writing and reading of the data of the service to be migrated. For example, the application server can still write and read data to a data server in the old data server cluster (such as the first data server cluster described above), and at the same time export the target data at the first time point and before the first time point from another data server in the old data server cluster to the new data server cluster (such as the second data server cluster described above); in the process of migrating the target data written to the old data server cluster after the first time point, the target data after the first time point is synchronously written to the new data server according to the operation log of the old data server cluster. In this process, it is still not necessary to stop the service to be migrated. After the two new and old data server clusters are completely synchronized, the service to be migrated can directly write and read data to the new data server cluster normally.
It can be seen that by adopting the technical solution of the present invention, during the entire business migration process, there is no need to stop the business to be migrated, and thus it will not affect the normal external service and operation of the business, thereby achieving the beneficial effect of realizing data migration without stopping for backup.

## [0057]

The algorithms and displays presented herein are not inherently related to any particular computer, virtual system, or other apparatus.
Various general-purpose systems can also be used with the teachings based thereon. The structure required to construct such a system should be apparent from the above description. Furthermore, the present invention is not directed to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the present invention as described herein and that the above descriptions using specific languages are intended to disclose the best mode for carrying out the present invention.

## [0058]

In the description provided herein, numerous specific details are set forth.
However, it is understood that embodiments of the invention may be practiced without these specific details. In some instances, well-known methods, structures and techniques have not been shown in detail in order not to obscure the understanding of this description.

## [0059]

Similarly, it should be appreciated that in order to streamline the disclosure and aid in understanding one or more of the various inventive aspects, in the above description of exemplary embodiments of the invention, various features of the invention are sometimes grouped together into a single embodiment, figure, or description thereof.
This method of disclosure, however, is not to be interpreted as reflecting an intention that the claimed invention requires more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive aspects lie in less than all features of a single foregoing disclosed embodiment. Thus, the claims following the Detailed Description are hereby expressly incorporated into this Detailed Description, with each claim standing on its own as a separate embodiment of this invention.

## [0060]

Those skilled in the art will appreciate that the modules in the device in the embodiment may be adaptively changed and arranged in one or more devices different from the embodiment. The modules or units or components in the embodiments may be combined into one module or unit or component, and further may be divided into a plurality of sub-modules or sub-units or sub-components. All features disclosed in this specification (including accompanying claims, abstract and drawings) and all processes or units of any method or apparatus so disclosed may be combined in any combination, except that at least some of such features and/or processes or units are mutually exclusive. Unless expressly stated otherwise, each feature disclosed in this specification (including accompanying claims, abstract and drawings) may be replaced by alternative features serving the same, equivalent or similar purpose.

## [0061]

Furthermore, those skilled in the art will appreciate that although some embodiments herein include certain features included in other embodiments but not other features, the combination of features of different embodiments is meant to be within the scope of the present invention and to form different embodiments.
For example, in the following claims, any of the claimed embodiments can be used in any combination.

## [0062]

The various component embodiments of the present invention may be implemented in hardware, or in software modules running on one or more processors, or in a combination of these.
Those skilled in the art should understand that a microprocessor or a digital signal processor (DSP) can be used in practice to implement some or all functions of some or all components of the migration device for migrating data between data server clusters according to an embodiment of the present invention. The present invention may also be implemented as a device or device program (e.g., a computer program and a computer program product) for executing part or all of the methods described herein. Such a program for realizing the present invention may be stored on a computer-readable medium, or may have the form of one or more signals. Such a signal may be downloaded from an Internet website, or provided on a carrier signal, or in any other form.

## [0063]

It should be noted that the above-mentioned embodiments illustrate rather than limit the invention and that those skilled in the art will be able to design alternative embodiments without departing from the scope of the appended claims.
In the claims, any reference signs placed between parentheses shall not be constructed as limiting the claim. The word "comprising" does not exclude the presence of elements or steps not listed in a claim. The word "a" or "an" preceding an element does not exclude the presence of a plurality of such elements. The invention can be implemented by means of hardware comprising several distinct elements, and by means of a suitably programmed computer. In a unit claim enumerating several means, several of these means may be embodied by one and the same item of hardware. The usage of the words first, second, and third etc. do not indicate any order. These words may be interpreted as names.

# Notice



## CLAIMS CN102917072A

### 1.

A migration device for migrating data between data server clusters, wherein the data is target data related to a business to be migrated, and the migration device comprises:
an initial data import module, configured to import the target data stored in the first data server cluster and written at or before the first time point into the second data server cluster;
a synchronization module configured to obtain an operation log associated with target data written to the first data server cluster after the first time point, the operation log at least including content of the target data written to the first data server cluster; and
An update module is configured to update target data in the second data server cluster according to the operation log obtained by the synchronization module.

### 2.

According to the migration device of claim 1, the initial data import module comprises:
A first initial data import submodule is configured to import the target data written at the first time point and before the first time point stored in the first data server cluster into a storage medium;
The second initial data import submodule is configured to import the target data imported into the storage medium into the second data server cluster.

### 3.

The migration device according to claim 2, wherein the first initial data import submodule is configured to back up the target data written at and before the first time point stored in the first data server cluster to a storage medium through mongodump; and
The second initial data import submodule is configured to import the target data backed up in the storage medium into the second data server cluster through mongorestore.

## 4.

According to the migration device according to any one of claims 1 to 3, the first data server cluster includes a master data server and a plurality of slave data servers, and the migration device further includes:
a deactivation processing module configured to stop the write operation of the first slave data server in the first data server cluster after a first time point; and
The initial data import module is configured to import the target data stored at the first time point in the first slave data server of the first data server cluster and written before the first time point into the second data server cluster.

## 5.

The migration device according to any one of claims 1 to 4, further comprising:
a synchronization detection module, configured to detect whether target data of the first data server cluster and the second data server cluster are synchronized;
The address update module is configured to change the entry address of the data server connected from the entry address of the first data server cluster to the entry address of the second data server cluster after the synchronization detection module detects that the target data of the first data server cluster and the second data server cluster have been synchronized.

## 6.

According to the migration device according to any one of claims 1 to 5, the operation log further includes one or more of the following information:
The timestamp of when the data was written; and
When the data written is an update of the original data, the old value before the data update.

## 7.

According to the migration device according to any one of claims 1 to 6, the first data server cluster is a first MongoDB cluster, the second data server cluster is a second MongoDB cluster, and the operation log is an oplog in MongoDB.

## 8.

A method for migrating data between data server clusters, wherein the data is target data related to a business to be migrated, the method comprising:
Importing the target data stored in the first data server cluster and written at or before the first time point into the second data server cluster;
Obtaining an operation log associated with target data written to the first data server cluster after a first time point, the operation log at least including content of the target data written to the first data server cluster; and
The target data in the second data server cluster is updated according to the obtained operation log.

## 9.

According to the method of claim 8, the step of importing the target data stored in the first data server cluster and written at and before the first time point into the second data server cluster comprises:
Importing the target data stored in the first data server cluster at the first time point and written before the first time point into a storage medium; and
The target data imported into the storage medium is imported into the second data server cluster.

## 10.

The method according to claim 9, wherein
Backing up the target data stored at the first time point and written before the first time point in the first data server cluster to a storage medium through mongodump; and
Import the target data backed up in the storage medium into the second data server cluster through mongorestore.

## 11.

According to the method of any one of claims 8 to 10, the first data server cluster includes a master data server and a plurality of slave data servers, and the method further comprises:
stopping the write operation of the first slave data server in the first data server cluster after the first time point; and

The step of importing the target data stored in the first data server cluster and written before the first time point into the second data server cluster includes:
Import the target data stored at the first time point in the first slave data server in the first data server cluster and written before the first time point into the second data server cluster.

## 12.

The method according to any one of claims 8 to 11, further comprising:
Detecting whether target data of the first data server cluster and the second data server cluster are synchronized; and
After the target data of the first data server cluster and the second data server cluster are synchronized, the entry address for connecting to the data server is changed from the entry address of the first data server cluster to the entry address of the second data server cluster.

## 13.

According to the method according to any one of claims 8 to 12, the operation log further includes one or more of the following information:
The timestamp of when the data was written; and
When the data written is an update of the original data, the old value before the data update.

## 14.

According to the method according to any one of claims 8 to 13, the first data server cluster is a first MongoDB cluster, the second data server cluster is a second MongoDB cluster, and the operation log is an oplog in MongoDB.