



US 20120185432A1

(19) **United States**

(12) **Patent Application Publication**  
**Ding**

(10) **Pub. No.: US 2012/0185432 A1**

(43) **Pub. Date: Jul. 19, 2012**

(54) **METHOD, DEVICE AND SYSTEM FOR IMPLEMENTING DATA SYNCHRONIZATION BETWEEN SOURCE DATABASE AND TARGET DATABASE**

**Publication Classification**

(51) **Int. Cl.**  
**G06F 7/00** (2006.01)

(52) **U.S. Cl.** ..... **707/620; 707/E17.005**

(57) **ABSTRACT**

(75) Inventor: **Qipeng Ding**, Shenzhen (CN)

(73) Assignee: **ZTE CORPORATION**, Shenzhen, Guangdong (CN)

(21) Appl. No.: **13/258,244**

(22) PCT Filed: **Apr. 13, 2010**

(86) PCT No.: **PCT/CN2010/071725**

§ 371 (c)(1),  
(2), (4) Date: **Mar. 29, 2012**

The disclosure discloses a method for implementing data synchronization between a source database and a target database, which includes: a synchronization device acquires at least one first Logical Change Record (LCR) and a keyword of the first LCR from the source database; the synchronization device determines whether there is a second LCR with a keyword as same as the keyword of the first LCR in the synchronization device itself; the synchronization device processes the first LCR to generate a processed LCR according to the determination result, so that the synchronization device stores only one LCR containing the keyword of the first LCR; and the synchronization device sends the processed LCR to the target database. Accordingly, the disclosure also discloses a device and system for implementing data synchronization between a source database and a target database. The disclosure can reduce the data transmission load when implementing the synchronization between the source database and the target database.

(30) **Foreign Application Priority Data**

Oct. 23, 2009 (CN) ..... 200910236502.3

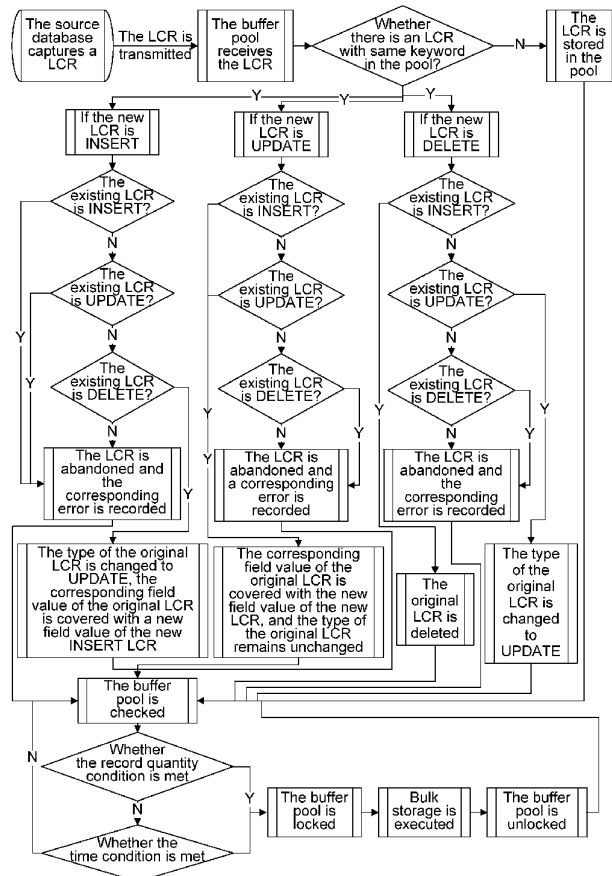


Fig. 1

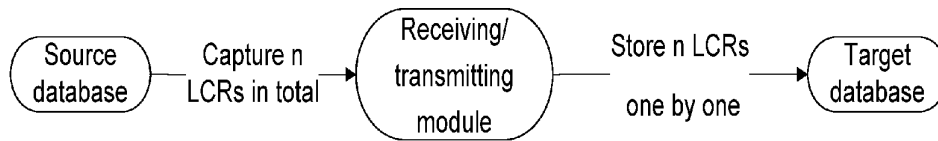


Fig. 2

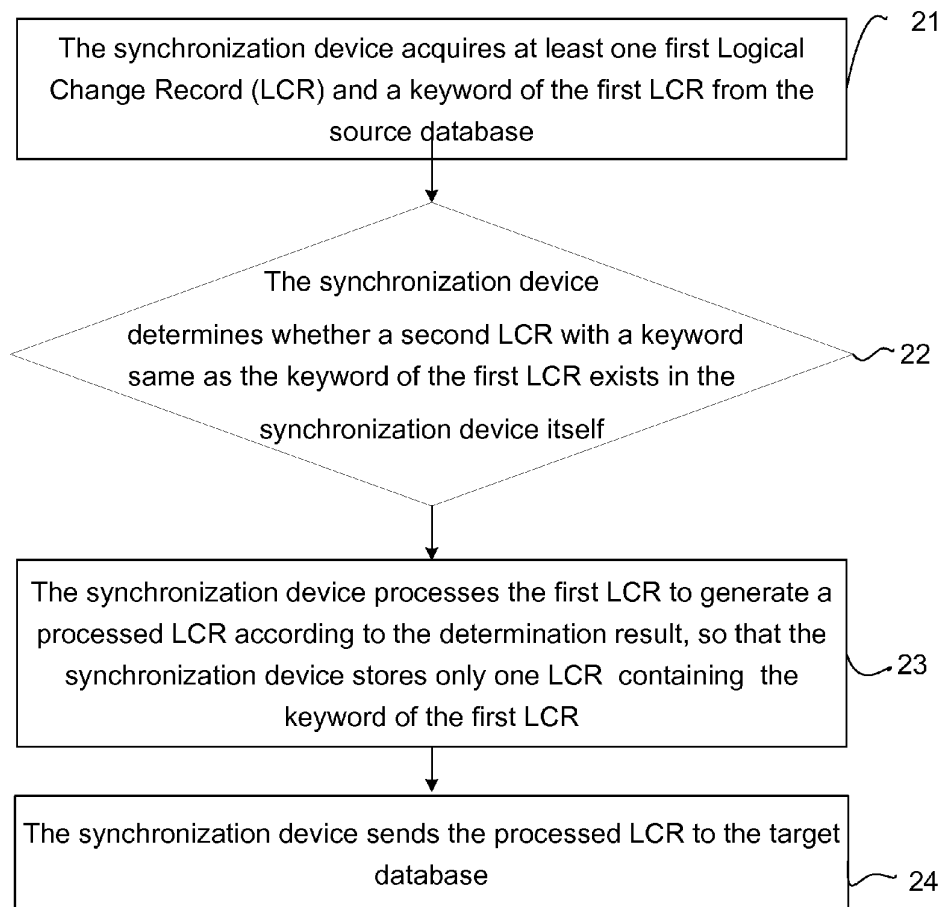


Fig. 3

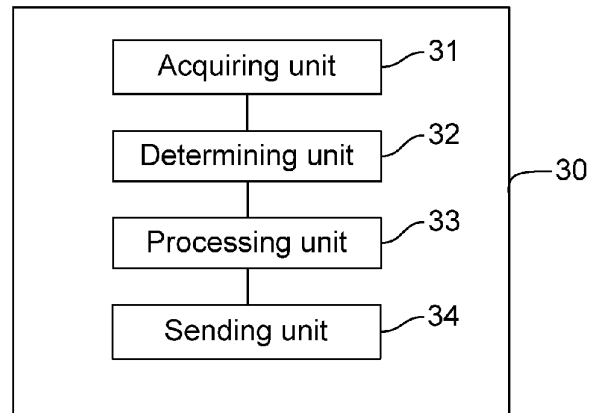


Fig. 4

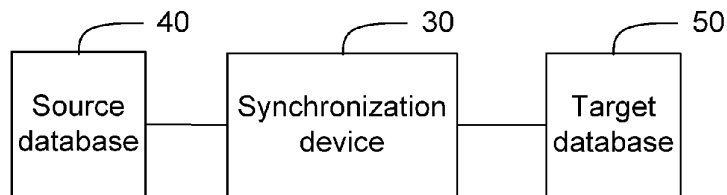


Fig.5

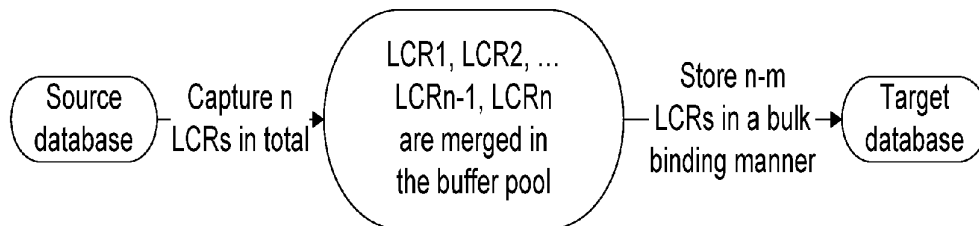
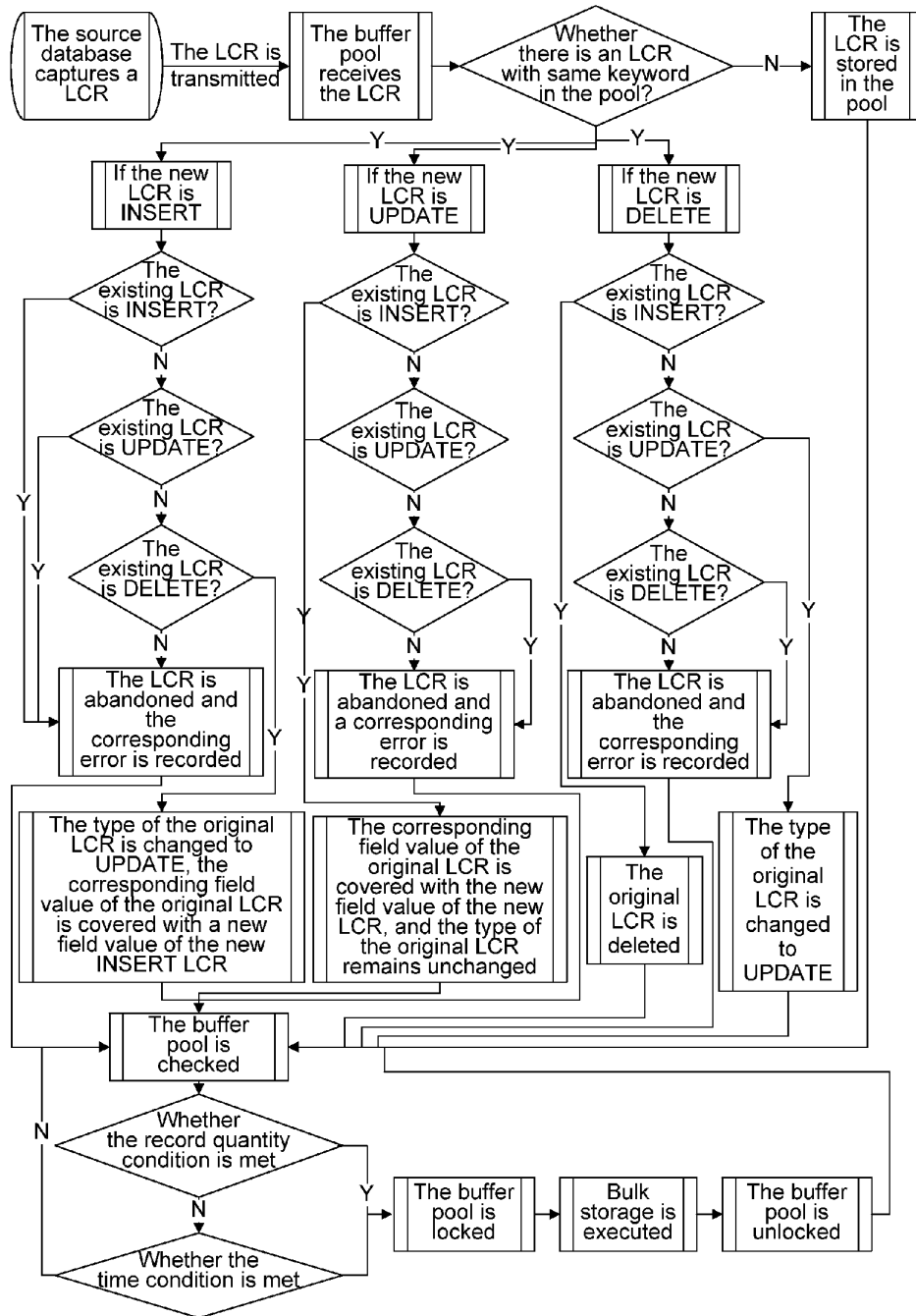


Fig. 6



**METHOD, DEVICE AND SYSTEM FOR  
IMPLEMENTING DATA SYNCHRONIZATION  
BETWEEN SOURCE DATABASE AND  
TARGET DATABASE**

TECHNICAL FIELD

**[0001]** The disclosure relates to the field of databases, in particular to a method, device and system for implementing data synchronization between a source database and a target database.

BACKGROUND

**[0002]** Since Dr. E. F. Codd proposed an initial theoretical model in 1970, a relation-type database has become an essential infrastructure in all information systems, and ranks only second to operating systems. Today, large-scale systems, which can operate without databases, have not existed any more.

**[0003]** In actual application, due to the complicity of service logics and the modularized design of functions, one information system is always deployed with multiple databases, and the multiple databases have to collaborate and share data with each other, and this requires data synchronization among the databases. In order to achieve this purpose, mainstream database vendors have developed their respective data synchronization technologies, such as, Streams from Oracle and Replication Server from Sybase. FIG. 1 shows a diagram of a traditional data synchronization process, and as shown in FIG. 1, Logical Change Records (LCRs) are captured from a database redo log in a source database, and are applied one by one to a target database according to the occurrence sequence of the LCRs, so that the synchronization of data in the target database with data in the source database is achieved.

**[0004]** LCR is a general name of a database change record in the technical field of databases, one LCR shows one specific Database Manipulation Language (DML) operation on one specific record in a source database, the contents of the LCR include: an operation type, a changed field list, a field value before change, a field value after change, a field name of a primary key, a field value of the primary key and so on; according to these pieces of information, the DML operation can be repeated.

**[0005]** The current data synchronization technologies have the following defects:

**[0006]** a source database sends all changed LCRs to a target database, therefore, when the source database changes frequently, the work load of LCR transmission is large.

SUMMARY

**[0007]** The technical problem to be solved in the disclosure is to provide a method, device and system for implementing data synchronization between a source database and a target database, to save resources of the target database.

**[0008]** In order to solve the above technical problem, embodiments of the disclosure provide the following technical solution.

**[0009]** A method for implementing data synchronization between a source database and a target database includes:

**[0010]** acquiring, by a synchronization device, at least one first Logical Change Record (LCR) and a keyword of the first LCR from a source database;

**[0011]** determining, by the synchronization device, whether there is a second LCR with a keyword as same as the keyword of the first LCR in the synchronization device itself, and generating, by the synchronization device, a determination result;

**[0012]** processing, by the synchronization device, the first LCR to generate a processed LCR according to the determination result, so that the synchronization device stores only one LCR containing the keyword of the first LCR;

**[0013]** sending, by the synchronization device, the processed LCR to a target database.

**[0014]** The processing, by the synchronization device, the first LCR to generate a processed LCR according to the determination result may include:

**[0015]** when the determination result is YES, merging, by the synchronization device, the first LCR with the second LCR to generate the processed LCR, according to an operation type of the first LCR and an operation type of the second LCR;

**[0016]** when the determination result is NO, storing, by the synchronization device, the first LCR in the synchronization device as the processed LCR.

**[0017]** The merging, by the synchronization device, the first LCR and the second LCR to generate the processed LCR may include:

**[0018]** when the operation type of the first LCR is INSERT and the operation type of the second LCR is INSERT or UPDATE, taking, by the synchronization device, the second LCR as the processed LCR;

**[0019]** when the operation type of the first LCR is INSERT and the operation type of the second LCR is DELETE, changing, by the synchronization device, the operation type of the second LCR to UPDATE, and covering, by the synchronization device, a corresponding field value of the second LCR with each field value of the first LCR, to generate the processed LCR;

**[0020]** when the operation type of the first LCR is UPDATE and the operation type of the second LCR is INSERT or UPDATE, covering, by the synchronization device, a corresponding field value of the second LCR with each field value of the first LCR, to generate the processed LCR;

**[0021]** when the operation type of the first LCR is UPDATE or DELETE and the operation type of the second LCR is DELETE, taking, by the synchronization device, the second LCR as the processed LCR;

**[0022]** when the operation type of the first LCR is DELETE and the operation type of the second LCR is INSERT, taking, by the synchronization device, the first LCR as the processed LCR; and

**[0023]** when the operation type of the first LCR is DELETE and the operation type of the second LCR is UPDATE, changing the type of the second LCR to DELETE and taking the second LCR as the processed LCR by the synchronization device.

**[0024]** The sending, by the synchronization device, the processed LCR to a target database may include:

**[0025]** sending, by the synchronization device, the processed LCR to the target database in a bulk binding way.

**[0026]** The sending, by the synchronization device, the processed LCR to the target database in a bulk binding way may include:

[0027] when the number of the processed LCRs reaches a predetermined threshold, sending, by the synchronization device, the processed LCRs to the target database in a bulk binding way; or

[0028] sending, by the synchronization device, the processed LCRs to the target database in a bulk binding way every fixed period.

[0029] The keyword of the first LCR may be a primary key or an identifier of the first LCR.

[0030] A synchronization device includes:

[0031] an acquiring unit, configured to acquire at least one first LCR and a keyword of the first LCR from a source database;

[0032] a determining unit, configured to determine whether there is a second LCR with a keyword as same as the keyword of the first LCR in the determining unit itself, and to generate a determination result;

[0033] a processing unit, configured to process the first LCR to generate a processed LCR according to the determination result; and

[0034] a sending unit, configured to send the processed LCR to a target database.

[0035] The processing unit may include:

[0036] a merging unit, configured to, when the determination result is YES, merge the first LCR with the second LCR to generate the processed LCR according to an operation type of the first LCR and an operation type of the second LCR;

[0037] a storage unit, configured to, when the determination result is NO, store the first LCR as the processed LCR.

[0038] A synchronization system includes: a synchronization device, a source database and a target database, wherein the synchronization device includes:

[0039] an acquiring unit, configured to acquire at least one first LCR and a keyword of the first LCR from the source database;

[0040] a determining unit, configured to determine whether there is a second LCR with a keyword as same as the keyword of the first LCR in the determining unit itself, and to generate a determination result;

[0041] a processing unit, configured to process the first LCR to generate a processed LCR according to the determination result; and

[0042] a sending unit, configured to send the processed LCR to the target database.

[0043] The synchronization device may be arranged on the source database or the target database, or may be independent of the source database or the target database.

[0044] The embodiments of the present disclosure have the following advantageous effects:

[0045] In above solution, when data synchronization is implemented between a source database and a target database, not all changed LCRs are sent to the target database, instead, they are buffered in a synchronization device; in this way, even if the source database changes frequently, each time the synchronization device acquires the first LCR from the source database, it processes the first LCR in the synchronization device first, that is, the synchronization device determines whether a second LCR with the keyword of the first LCR exists in the synchronization device itself and generates a determination result; the synchronization device processes the first LCR to generate a processed LCR according to the determination result, so that the synchronization device stores only one LCR containing the keyword of the first LCR, then

the synchronization device sends the processed LCR to the target database. Compared with the prior art, the work load of LCR transmission is reduced.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0046] FIG. 1 shows a diagram of a traditional data synchronization process;

[0047] FIG. 2 shows a flow diagram of a method for implementing data synchronization between a source database and a target database in the disclosure;

[0048] FIG. 3 shows a structural diagram of a synchronization device in the disclosure;

[0049] FIG. 4 shows a structural diagram of a synchronization system in the disclosure;

[0050] FIG. 5 shows a process diagram of the method for implementing data synchronization between a source database and a target database in the disclosure; and

[0051] FIG. 6 shows a process diagram of an embodiment of the method for implementing data synchronization between a source database and a target database in the disclosure.

#### DETAILED DESCRIPTION

[0052] In order to make the technical problem to be solved, technical solution and advantages in embodiments of the disclosure more clear, detailed description will be given below in conjunction with drawings and specific embodiments.

[0053] As to the problem of large LCR transmission load during data synchronization between a source database and a target database in the prior art, the embodiments of the disclosure provide a method, device and system for implementing data synchronization between a source database and a target database.

[0054] FIG. 2 shows a flow diagram of the method for implementing data synchronization between a source database and a target database in the disclosure; as shown in FIG. 2, the method for implementing data synchronization between a source database and a target database comprises:

[0055] step 21: a synchronization device acquires at least one first Logical Change Record (LCR) and a keyword of the first LCR from a source database;

[0056] step 22: the synchronization device determines whether a second LCR whose keyword is as same as the keyword of the first LCR exists in the synchronization device itself, and then generates a determination result;

[0057] step 23: the synchronization device processes the first LCR to generate a processed LCR according to the determination result, so that the synchronization device stores only one LCR containing the keyword of the first LCR; and

[0058] step 24: the synchronization device sends the processed LCR to a target database.

[0059] In the above solution, when data synchronization is implemented between the source database and the target database, not all changed LCRs are sent to the target database, instead, they are buffered in the synchronization device; in this way, even if the source database changes frequently, each time the synchronization device acquires the first LCR from the source database, it processes the first LCR in the synchronization device first, that is, the synchronization device determines whether there is a second LCR with the keyword of the first LCR in the synchronization device itself and generates a determination result; the synchronization device processes

the first LCR to generate the processed LCR according to the determination result, so that the synchronization device stores only one LCR containing the keyword of the first LCR, then the synchronization device sends the processed LCR to the target database. Compared with the prior art, the work load of LCR transmission is reduced.

[0060] Wherein, the step 23 comprises:

[0061] when the determination result is YES, the synchronization device merges the first LCR with the second LCR to generate the processed LCR, according to the operation type of the first LCR and the operation type of the second LCR;

[0062] when the determination result is NO, the synchronization device stores the first LCR in the synchronization device and takes the first LCR as the processed LCR.

[0063] Specifically, the step of the synchronization device merging the first LCR with the second LCR to generate the processed LCR according to the operation type of the first LCR and the operation type of the second LCR may comprise:

[0064] when the operation type of the first LCR is INSERT and the operation type of the second LCR is INSERT or UPDATE, the synchronization device takes the second LCR as the processed LCR;

[0065] when the operation type of the first LCR is INSERT and the operation type of the second LCR is DELETE, the synchronization device changes the operation type of the second LCR to UPDATE, and covers a corresponding field value of the second LCR with each field value of the first LCR, to generate the processed LCR;

[0066] when the operation type of the first LCR is UPDATE and the operation type of the second LCR is INSERT or UPDATE, the synchronization device covers a corresponding field value of the second LCR with each field value of the first LCR, to generate the processed LCR;

[0067] when the operation type of the first LCR is UPDATE or DELETE and the operation type of the second LCR is DELETE, the synchronization device takes the second LCR as the processed LCR;

[0068] when the operation type of the first LCR is DELETE and the operation type of the second LCR is INSERT, the synchronization device takes the first LCR as the processed LCR;

[0069] when the operation type of the first LCR is DELETE and the operation type of the second LCR is UPDATE, the synchronization device changes the type of the second LCR to DELETE and takes the second LCR as the processed LCR.

[0070] The step 24 may comprise: the synchronization device sends the processed LCR to the target database in a bulk binding way.

[0071] The step of sending the processed LCR by the synchronization device to the target database in a bulk binding way comprises:

[0072] when the number of the processed LCRs reaches a predetermined threshold, the synchronization device sends the processed LCRs to the target database in a bulk binding way; or

[0073] the synchronization device sends the processed LCR to the target database in a bulk binding way every fixed period.

[0074] The keyword of the first LCR is a primary key or an identifier of the first LCR.

[0075] FIG. 3 shows a structural diagram of a synchronization device in the disclosure; as shown in FIG. 3, a synchronization device 30 of the disclosure comprises:

[0076] an acquiring unit 31, configured to acquire at least one first LCR and a keyword of the first LCR from a source database;

[0077] a determining unit 32, configured to determine whether there is a second LCR whose keyword is as same as the keyword of the first LCR, and to generate a determination result;

[0078] a processing unit 33, configured to process the first LCR to generate a processed LCR according to the determination result;

[0079] a sending unit 34, configured to send the processed LCR to a target database.

[0080] In the above solution, when data synchronization is implemented between the source database and the target database, not all changed LCRs are sent to the target database, instead, they are buffered in the synchronization device; in this way, even if the source database changes frequently, each time the synchronization device acquires the first LCR from the source database, it processes the first LCR in the synchronization device first, that is, the synchronization device determines whether there is a second LCR containing the keyword of the first LCR in the synchronization device itself and generates a determination result; the synchronization device processes the first LCR to generate a processed LCR according to the determination result, so that the synchronization device stores only one LCR containing the keyword of the first LCR, then the synchronization device sends the processed LCR to the target database. Compared with the prior art, the work load of LCR transmission is reduced.

[0081] The processing unit 33 comprises:

[0082] a merging unit, configured to, when the determination result is YES, merge the first LCR with the second LCR to generate the processed LCR according to the operation type of the first LCR and the operation type of the second LCR;

[0083] a storage unit, configured to, when the determination result is NO, store the first LCR which is taken as the processed LCR.

[0084] The merging unit is specifically configured to:

[0085] when the operation type of the first LCR is INSERT and the operation type of the second LCR is INSERT or UPDATE, take the second LCR as the processed LCR;

[0086] when the operation type of the first LCR is INSERT and the operation type of the second LCR is DELETE, change the operation type of the second LCR to UPDATE, cover a corresponding field value of the second LCR with each field value of the first LCR, to generate the processed LCR;

[0087] when the operation type of the first LCR is UPDATE and the operation type of the second LCR is INSERT or UPDATE, cover a corresponding field value of the second LCR with each field value of the first LCR, to generate the processed LCR;

[0088] when the operation type of the first LCR is UPDATE or DELETE and the operation type of the second LCR is DELETE, take the second LCR as the processed LCR;

[0089] when the operation type of the first LCR is DELETE and the operation type of the second LCR is INSERT, take the first LCR as the processed LCR; and

[0090] when the operation type of the first LCR is DELETE and the operation type of the second LCR is UPDATE, change the type of the second LCR to DELETE and take the second LCR as the processed LCR.

[0091] Wherein the sending unit 34 may send the processed LCRs to the target database in a bulk binding way.

**[0092]** Further, when the number of the processed LCRs reaches a predetermined threshold, the sending unit **34** may send the processed LCRs to the target database in a bulk binding way; or, the sending unit **34** sends the processed LCRs to the target database in a bulk binding way every fixed period. The keyword of the first LCR is a primary key or an identifier of the first LCR.

**[0093]** FIG. 4 shows a structural diagram of a synchronization system in the disclosure; as shown in FIG. 4, the synchronization system comprises: a synchronization device **30**, a source database **40** and a target database **50**, wherein the synchronization device **30** comprises:

**[0094]** an acquiring unit **31**, configured to acquire at least one first LCR and a keyword of the first LCR from a source database;

**[0095]** a determining unit **32**, configured to determine whether a second LCR whose keyword is as same as the keyword of the first LCR exists in the determining unit itself, and to generate a determination result;

**[0096]** a processing unit **33**, configured to process the first LCR to generate a processed LCR according to the determination result; and

**[0097]** a sending unit **34**, configured to send the processed LCR to a target database.

**[0098]** In the above solution, when data synchronization is implemented between the source database and the target database, not all changed LCRs are sent to the target database, instead, they are buffered in a synchronization device; in this way, even if the source database changes frequently, each time the synchronization device acquires the first LCR from the source database, it processes the first LCR in the synchronization device first, that is, the synchronization device determines whether there is a second LCR containing the keyword of the first LCR in the synchronization device itself and generates a determination result; the synchronization device processes the first LCR to generate a processed LCR according to the determination result, so that the synchronization device stores only one LCR containing the keyword of the first LCR, then the synchronization device sends the processed LCR to the target database. Compared with the prior art, the work load of LCR transmission is reduced.

**[0099]** The synchronization device is arranged on the source database or the target database, or it is independent from the source database or the target database.

**[0100]** FIG. 5 shows a process diagram of the method for implementing data synchronization between a source database and a target database in the disclosure; as shown in FIG. 5, the high-speed data synchronization method based on a desynchronizing mirroring mechanism in the disclosure comprises: LCRs of the source database are cached in a buffer pool, wherein the buffer pool is a synchronization device; if an LCR of a certain record has already existed, LCR merging is executed, and  $m$  is the number of LCRs merged during the merging process, so that only one LCR determined by a certain keyword exists in the buffer pool; after a period of time, the merged LCRs are sent to the target database to be stored; under this model, it looks like some LCRs are lost during the synchronization process, and actually, the LCRs existing in the buffer pool are mirror images of original records, the mirror images existing in special form, therefore this synchronization model is called "desynchronizing mirroring". By adopting this model, the number of LCRs to be sent to the target database to be executed is reduced, system resources are saved, and the influence on major services of the

target database is decreased. Meanwhile, since an LCR for a same record only exists in the buffer pool once, the technology of bulk binding can be employed to achieve bulk storage, thus the speed of storage is improved. To achieve above purpose, the disclosure, as to the case of merging in the buffer pool through different DMLs, creates a set of merging rules and executes bulk binding storage of the merged LCRs in the pool at a fixed time or a fixed quantity.

**[0101]** The method for implementing data synchronization between a source database and a target database in the disclosure is described below, which comprises the following steps.

**[0102]** Step 1: a buffer pool with a proper size is allocated in a machine which is connected with a source database and a target database at the same time, wherein the machine is the synchronization device described above, and the buffer pool can completely reside in a system memory, or can be stored in a disk by means of expansion when the system memory is not large enough to accommodate the buffer pool.

**[0103]** Step 2: a set of data management mechanism is created in the buffer pool, wherein the mechanism can realize:

**[0104]** storing one LCR;

**[0105]** determining whether there is an LCR meeting the requirement according to a value of a keyword; that is to say, determining whether an LCR with a keyword of a new LCR exists in the buffer pool; the keyword can be a primary key of the new LCR;

**[0106]** finding out the LCR meeting the requirement in the pool according to the value of the keyword of the new LCR;

**[0107]** locating the LCR meeting the requirement according to the value of the keyword and updating the LCR;

**[0108]** quickly locating the LCR meeting the requirement according to the value of the keyword and deleting it.

**[0109]** Step 3: LCRs are acquired from a database log (i.e. redo log) of the source database and sending the LCRs to the buffer pool.

**[0110]** Step 4: after the buffer pool receives the LCRs, the buffer pool stores or merges the LCRs using its own LCR processing function.

**[0111]** Step 5: two types of thresholds are set in order to guarantee the timeliness of the synchronization process.

**[0112]** The first type is a quantitative threshold, that is to say, when the number of the LCRs cached in the buffer pool reaches a certain value, the data is sent to the target database for application;

**[0113]** the second type is a time threshold, that is to say, when the time determined by the time threshold has passed since the last time of sending the data to the target database for application, the data is sent to the target database for application.

**[0114]** Step 6: when any one of the two types of thresholds involved in the step 5 is met, the data is applied to the target database in a bulk binding way.

**[0115]** In this method, it is needed to establish a buffer pool and create a set of LCR management mechanisms in the buffer pool; and during the period when LCRs exist in the buffer pool, in the case that there are LCRs with same keyword, the LCRs are merged, thus the total number of LCRs is reduced; LCRs are first cached in the buffer pool for a period of time and then applied to the target database; there are two storage trigger conditions including: triggering at a fixed time and triggering with a fixed quantity, thus guaranteeing the timeliness of LCR application; and the technology of bulk binding realizes high-speed storage.



[0116] FIG. 6 shows a process diagram of an embodiment of the method for implementing data synchronization between a source database and a target database in the disclosure; as shown in FIG. 6, the method for implementing data synchronization between a source database and a target database in another embodiment of the disclosure comprises the following steps.

[0117] Step 1: it is determined that a host can communicate with a source database and a target database at the same time, and is equipped with basic software for operating the source database and the target database, such as oci database and ct-lib database, the host acting as a synchronization device.

[0118] Step 2: a buffer pool with a proper size is allocated, wherein the buffer pool can completely reside in system memory, or the buffer pool can be stored in a disk by means of expansion when the system memory is not large enough to accommodate the buffer pool.

[0119] Step 3: a set of quick index mechanisms is created in the buffer pool based on technologies such as hash or b-tree or rbt, through which an LCR can be added, deleted, modified and found quickly.

[0120] Step 4: a set of LCR merging mechanisms is created in the pool, specifically as follows:

[0121] case 1: when a new INSERT LCR is acquired, and it is found in the buffer pool at this time that:

[0122] case 1.1: if there is no LCR which has the same keyword as the new LCR, then the new LCR is stored in the buffer pool;

[0123] case 1.2: if there has existed one INSERT LCR which has the same keyword as the new LCR, then the new LCR is abandoned and the abnormal condition is recorded;

[0124] case 1.3: if there has existed one UPDATE LCR which has the same keyword as the new LCR, then the new LCR is abandoned and the abnormal condition is recorded;

[0125] case 1.4: if there has existed one DELETE LCR which has the same keyword as the new LCR, then the type of the original LCR is changed to UPDATE, and a corresponding field value of the original LCR is covered with each new field value of the new INSERT LCR.

[0126] case 2: when a new UPDATE LCR is acquired, and it is found in the buffer pool at this time that:

[0127] case 2.1: if there is no LCR which has the same keyword as the new LCR, then the new LCR is stored into the buffer pool;

[0128] case 2.2: if there has existed one INSERT LCR which has the same keyword as the new LCR, then a corresponding field value of the original LCR is covered with each new field value of the new LCR, and the type of the original LCR remains INSERT;

[0129] case 2.3: if there has existed one UPDATE LCR which has the same keyword as the new LCR, then a corresponding field value of the original LCR is covered with each new field value of the new LCR, and the type of the original LCR remains UPDATE;

[0130] case 2.4: if there has existed one DELETE LCR which has the same keyword as the new LCR, then the new LCR is abandoned and the abnormal condition is recorded.

[0131] case 3: when a new DELETE LCR is acquired, and it is found in the buffer pool at this time that:

[0132] case 3.1: if there is no LCR which has the same keyword as the new LCR, then the new LCR is stored into the buffer pool;

[0133] case 3.2: if there has existed one INSERT LCR which has the same keyword as the new LCR, then the original LCR is deleted;

[0134] case 3.3: if there has existed one UPDATE LCR which has the same keyword as the new LCR, then the type of the original LCR is changed to DELETE;

[0135] case 3.4: if there has existed one DELETE LCR which has the same keyword as the new LCR, then the new LCR is abandoned and the abnormal condition is recorded.

[0136] Step 5: a quantity threshold and a time threshold are set for the buffer pool;

[0137] Step 6: the number of LCRs in the buffer pool is checked; if it does not reach the quantity threshold, then the process jumps to step 7; if it has already reached the quantity threshold, then the buffer pool is locked to prohibit new LCRs from entering the pool, all LCRs in the pool are applied to the target database by using the bulk storage mechanism, and errors occurring during the bulk storage process are recorded. When the storage action is finished, the time point is recorded; and the process returns to step 7.

[0138] Step 7: it is checked whether the current time calculated from the time point of finishing the last storage action has already exceeded the time threshold; if it has not exceeded the time threshold, then the process returns to step 6; if it has exceeded the time threshold, then all LCRs in the pool are applied to the target database by using the bulk storage mechanism, and errors occurring during bulk storage process are recorded. When the storage action is finished, the time point is recorded; and the process returns to step 6.

[0139] The first application scene of the method for implementing data synchronization between a source database and a target database in the disclosure will be described below. In this application scene, one LCR is triggered based on the quantity condition and applied to the target database in the case that there is no same keyword in the buffer pool. The application scene comprises the following steps:

[0140] step 1: it is assumed that there is a table DEMO\_TAB in both the source database and the target database, which is defined as follows:

---

```
CREATE TABLE DEMO_TAB(
  FLD_A NUMBER PRIMARY KEY,
  FLD_B CHAR(100),
  FLD_C VARCHAR(100))
```

---

[0141] the four fields in this table represent three most common data types, wherein FLD\_A is a primary key;

[0142] step 2: INSERT actions are executed for N times in the source database:

---

```
INSERT INTO DEMO_TAB VALUES(1,'CHAR1','VARCHAR1');
INSERT INTO DEMO_TAB VALUES(2,'CHAR2','VARCHAR2');
...
INSERT INTO DEMO_TAB VALUES(N,'CHARN','VARCHARN');
```

---

[0143] step 3: the N INSERT actions are captured through the LCR capture mechanism, wherein the contents of the captured LCRs are as follows:

**[0144]** LCR1 contents:

Table name: DEMO_TAB	Operation type: INSERT
----------------------	------------------------

**[0145]** Original value list of fields: None (INSERT action has no original value list)

**[0146]** New value list of fields:

Field 1: name: FLD_A;	type: NUMBER;	value: 1
Field 2: name: FLD_B;	type: CHAR;	value: CHAR1
Field 3: name: FLD_C;	type: VARCHAR;	value: VARCHAR1
...		

(Subsequent LCR contents are similar to those of LCR 1, only the field values are different)

**[0147]** step 4: the N LCRs are sent to the buffer pool;

**[0148]** step 5: through searching, no LCR which has the same keyword as the new LCRs is found in the pool;

**[0149]** step 6: the N LCRs are stored into the buffer pool.

**[0150]** step 7: the number of LCRs in the pool are detected to have already exceeded the quantity threshold;

**[0151]** step 8: the buffer pool is locked to prohibit new LCRs from entering the pool;

**[0152]** step 9: the LCRs are fast stored into the pool for one time by using the technology of bulk binding; and

**[0153]** step 10: the buffer pool is unlocked.

**[0154]** The second application scene of the method for implementing data synchronization between a source database and a target database in the disclosure will be described below. In this application scene, one new LCR is triggered based on the time condition and applied to the target database in the case that an LCR having the same keyword as the new LCR exists in the buffer pool. The application scene comprises the following steps:

**[0155]** step 1: the table structure of the table in the step 1 in the last application scene is adopted, and a record is inserted into the table:

**[0156]** INSERT INTO DEMO\_TAB VALUES (1, 'CHAR1', 'VARCHAR1');

**[0157]** step 2: UPDATE actions are executed for N times regarding to this record in the source database:

```
UPDATE DEMO_TAB SET FLD_B='NEWCHAR1',
FLD_C='NEWVARCHAR1' WHERE FLD_A=1;
UPDATE DEMO_TAB SET FLD_B='NEWCHAR2',
FLD_C='NEWVARCHAR2' WHERE FLD_A=1;
...
UPDATE DEMO_TAB SET FLD_B='NEWCHARN',
FLD_C='NEWVARCHARN' WHERE FLD_A=1;
```

**[0158]** step 3: the N UPDATE actions are captured through the LCR capture mechanism, wherein the contents of the captured LCRs are as follows:

**[0159]** LCR1 contents:

Table name: DEMO_TAB	Operation type: UPDATE
----------------------	------------------------

**[0160]** Original value list of fields:

Field 2: name: FLD_B;	type: CHAR;	value: CHAR1
Field 3: name: FLD_C;	type: VARCHAR;	value: VARCHAR1

**[0161]** New value list of fields:

Field 2: name: FLD_B;	type: CHAR;	value: NEWCHAR1
Field 3: name: FLD_C;	type: VARCHAR;	value: NEWVARCHAR1

**[0162]** Keyword information:

Field 1: name: FLD_A;	type: NUMBER;	value: 1
...		

(Subsequent LCR contents are similar to those of LCR 1, only the field values in the new value list are different)

**[0163]** step 4: the N LCRs are sent to the buffer pool in turn;

**[0164]** step 5: through searching, there is no LCR which has a keyword the same as LCR1 in the pool;

**[0165]** step 6: LCR1 is stored into the buffer pool;

**[0166]** step 7: when LCR2 is stored into the pool and checked, if LCR1 having the same keyword as LCR2 is found in the pool, then the two LCRs are merged according to the merging rules: replacing the value of each field in the new value list of LCR1 in the pool with the corresponding field value in the new value list of LCR2, and then abandoning LCR2; this process is repeatedly performed for each subsequent LCR until LCR N is abandoned;

**[0167]** step 8: the number of LCRs in the pool are detected to have already exceeded the quantity threshold;

**[0168]** step 9: it is checked whether the current time calculated from the time of finishing last storage action exceeds the time threshold, and found out that the time has already exceeded the time threshold;

**[0169]** step 10: the buffer pool is locked to prohibit new LCRs from entering the pool;

**[0170]** step 11: the LCRs are stored into the pool for one time with high speed by using the technology of bulk binding; and

**[0171]** step 12: the buffer pool is unlocked.

**[0172]** The method, device and system for realizing high-speed data synchronization based on the desynchronizing mirroring mechanism can optimize quick storage by using the technology of bulk binding, and the target database does not need to apply all LCRs delivered from the source database one by one, thus the synchronization speed is improved.

**[0173]** The method embodiments correspond to the device embodiments, so those parts not described in details in the method embodiments can be referred to descriptions in relative parts in the device embodiments, and those parts not described in details in the device embodiments can be referred to descriptions in relative parts in the method embodiments.

**[0174]** It should be understood by those skilled in the art that all or part of the steps for realizing the method in the embodiments can be implemented by a program instructing relative hardware, wherein the program can be stored in a computer readable storage medium and execute the steps described in the method embodiments, the storage medium being disk, disc, Read-Only Memory (ROM) or Random Access Memory (RAM) or the like.

**[0175]** In the method embodiments of the disclosure, the sequence number of the steps are not used for limiting the sequence of the steps; for those skilled in the art, changes of the sequence of the steps without any creative work are also within the protection scope of the disclosure.

**[0176]** The above mentioned only describes the preferred embodiments of the disclosure, it should be noted that, for those skilled in the art, some improvements and elaborations can be made without departing from the principle of the disclosure, and those improvements and elaborations should be included within the protection scope of the disclosure.

1. A method for implementing data synchronization between a source database and a target database, comprising: acquiring, by a synchronization device, at least one first Logical Change Record (LCR) and a keyword of the first LCR from a source database;

determining, by the synchronization device, whether there is a second LCR with a keyword as same as the keyword of the first LCR in the synchronization device itself, and generating, by the synchronization device, a determination result;

processing, by the synchronization device, the first LCR to generate a processed LCR according to the determination result, so that the synchronization device stores only one LCR containing the keyword of the first LCR;

sending, by the synchronization device, the processed LCR to a target database.

2. The method for implementing data synchronization between a source database and a target database according to claim 1, wherein the processing, by the synchronization device, the first LCR to generate a processed LCR according to the determination result comprises:

when the determination result is YES, merging, by the synchronization device, the first LCR and the second LCR to generate the processed LCR, according to an operation type of the first LCR and an operation type of the second LCR;

when the determination result is NO, storing, by the synchronization device, the first LCR into the synchronization device as the processed LCR.

3. The method for implementing data synchronization between a source database and a target database according to claim 2, wherein the merging, by the synchronization device, the first LCR and the second LCR to generate the processed LCR comprises:

when the operation type of the first LCR is INSERT and the operation type of the second LCR is INSERT or UPDATE, taking, by the synchronization device, the second LCR as the processed LCR;

when the operation type of the first LCR is INSERT and the operation type of the second LCR is DELETE, changing, by the synchronization device, the operation type of the second LCR to UPDATE, and covering, by the synchronization device, a corresponding field value of the second LCR with each field value of the first LCR, to generate the processed LCR;

when the operation type of the first LCR is UPDATE and the operation type of the second LCR is INSERT or UPDATE, covering, by the synchronization device, a corresponding field value of the second LCR with each field value of the first LCR, to generate the processed LCR;

when the operation type of the first LCR is UPDATE or DELETE and the operation type of the second LCR is DELETE, taking, by the synchronization device, the second LCR as the processed LCR;

when the operation type of the first LCR is DELETE and the operation type of the second LCR is INSERT, taking, by the synchronization device, the first LCR as the processed LCR; and

when the operation type of the first LCR is DELETE and the operation type of the second LCR is UPDATE, changing the type of the second LCR to DELETE and taking the second LCR as the processed LCR by the synchronization device.

4. The method for implementing data synchronization between a source database and a target database according to claim 1, wherein the sending, by the synchronization device, the processed LCR to a target database comprises:

sending, by the synchronization device, the processed LCR to the target database in a bulk binding way.

5. The method for implementing data synchronization between a source database and a target database according to claim 4, wherein the sending, by the synchronization device, the processed LCR to the target database in a bulk binding way comprises:

when the number of the processed LCRs reaches a predetermined threshold, sending, by the synchronization device, the processed LCRs to the target database in a bulk binding way; or

sending, by the synchronization device, the processed LCRs to the target database in a bulk binding way every fixed period.

6. The method for implementing data synchronization between a source database and a target database according to claim 1, wherein the keyword of the first LCR is a primary key or an identifier of the first LCR.

7. A synchronization device, comprising:

an acquiring unit, configured to acquire at least one first Logical Change Record (LCR) and a keyword of the first LCR from a source database;

a determining unit, configured to determine whether there is a second LCR with a keyword as same as the keyword of the first LCR in the determining unit itself, and to generate a determination result;

a processing unit, configured to process the first LCR to generate a processed LCR according to the determination result; and

a sending unit, configured to send the processed LCR to a target database.

8. The synchronization device according to claim 7, wherein the processing unit comprises:

a merging unit, configured to, when the determination result is YES, merge the first LCR with the second LCR to generate the processed LCR according to an operation type of the first LCR and an operation type of the second LCR;

a storage unit, configured to, when the determination result is NO, store the first LCR as the processed LCR.

**9.** A synchronization system, comprising: a synchronization device, a source database and a target database, wherein the synchronization device comprises:

an acquiring unit, configured to acquire at least one first Logical Change Record (LCR) and a keyword of the first LCR from the source database;

a determining unit, configured to determine whether there is a second LCR with a keyword as same as the keyword of the first LCR in the determining unit itself, and to generate a determination result;

a processing unit, configured to process the first LCR to generate a processed LCR according to the determination result; and

a sending unit, configured to send the processed LCR to the target database.

**10.** The synchronization system according to claim **9**, wherein the synchronization device is arranged on the source database or the target database, or is independent of the source database or the target database.

\* \* \* \* \*

---