

CLAIMS

What is claimed is:

1. A method for managing primary and secondary databases, comprising the following steps:

5 determining that a lock held by a primary database has expired, wherein the primary database and the secondary databases share the lock, and wherein the lock is updated to extend a validity period of the lock before the validity period expires, wherein:

the lock is updated during an update cycle having a first time period T1,

10 the primary database sends, at a cycle with a second time period T2, a lock request for querying a status of the lock, and

a particular secondary database from the secondary databases sends, at a cycle with a third time period T3, a lock request for querying the status of the lock, wherein the second time period T2 is smaller than the first time period T1,

15 and the third time period T3 is greater than or equal to the first time period T1;

in response to determining that the lock held by the primary database has expired, determining that a lock renewal request of the primary database has not been received after the lock has expired; and

in response to determining that the lock renewal request of the primary database has not been received, selecting the particular secondary database as a new primary database, and controlling the primary database to be switched to the secondary database.

20

2. The method for managing primary and secondary databases according to claim 1, further comprising:

25 sending a lock expiration notification to the particular secondary database; and receiving a lock request from the particular secondary database based on the lock expiration notification, and recording a receiving time of the lock request.

3. The method for managing primary and secondary databases according to claim 2, wherein selecting the particular secondary database comprises:

30 selecting one of the secondary databases with an earliest receiving time.

4. The method for managing primary and secondary databases according to claim 1, wherein a priority of the primary database is higher than a priority of the particular

secondary database.

5. The method for managing primary and secondary databases according to claim 1, wherein the particular secondary database comprises a hot standby database and a disaster

recovery database, the primary database and the hot standby database are located in a same data center, and the primary database and the disaster recovery database are located in different data centers.

6. The method for managing primary and secondary databases according to claim 1, further comprising:

performing operations comprising at least one of a maintenance or an upgrade on the primary database.

7. The method for managing primary and secondary databases according to claim 6, further comprising:

determining whether the operations comprising at least one of the maintenance and the upgrade on the primary database are completed; and

in response to determining that the operations comprising at least one of the maintenance and the upgrade on the primary database are completed, restoring the primary database.

8. The method for managing primary and secondary databases according to claim 7, wherein after the primary database is restored, the method further comprises:

receiving a lock request sent by an original primary database, and controlling the lock to be held by the original primary database after the lock expires, wherein the original primary database is restored to the primary database, and a current primary database is restored to one of the secondary databases.

9. The method for managing primary and secondary databases according to claim 8, further comprising:

continuing to determine whether a lock held by the current primary database expires;

in response to determining that the lock held by the current primary database has expired, determining whether lock requests of the original primary database, the current primary database, and other secondary databases are received; and

in response to determining that the lock request of the original primary database is received, restoring the original primary database to the primary database, and restoring the current primary database to one of the secondary databases; or

in response to determining that the lock request of the original primary database is not received and the lock requests of the current primary database and the other secondary databases are received, maintaining the current primary database as the primary database; or

in response to determining that the lock requests of the original primary database and the current primary database are not received and the lock requests of the other secondary databases are received, selecting one of the other secondary databases as
 5 the primary database, and restoring the current primary database to one of the secondary databases.

10. A lock arbitration server, comprising:

a first determining module, configured to determine that a lock held by a primary database has expired, wherein the primary database and the secondary databases share
 10 the lock, and wherein the lock is updated to extend a validity period of the lock before the validity period expires, wherein:

the lock is updated during an update cycle having a first time period T1,

the primary database sends, at a cycle with a second time period T2, a lock request for querying a status of the lock, and

15 a particular secondary database from the secondary databases sends, at a cycle with a third time period T3, a lock request for querying the status of the lock, wherein the second time period T2 is smaller than the first time period T1, and the third time period T3 is greater than or equal to the first time period T1;

a second determining module, configured to: in response to determining that the

lock held by the primary database has expired, determine that a lock renewal request of the primary database has not been received after the lock has expired; and

a first processing module, configured to: in response to determining that the lock renewal request of the primary database has not been received, select the particular
5 secondary database as a new primary database, and control the primary database to be switched to the secondary database.

11. The lock arbitration server according to claim 10, further comprising:

a first sending module, configured to send a lock expiration notification to the particular secondary database; and

10 a record module, configured to: receive a lock request from the particular secondary database based on the lock expiration notification, and record a receiving time of the lock request.

12. The lock arbitration server according to claim 11, wherein the first processing module is further configured to:

15 select, as the particular secondary database, one of the secondary databases with an earliest receiving time.

13. The lock arbitration server according to claim 10, wherein a priority of the primary database is higher than a priority of the particular secondary database.

14. The lock arbitration server according to claim 10, wherein
20 the particular secondary database comprises a hot standby database and a disaster recovery database, the primary database and the hot standby database are located in a same data center, and the primary database and the disaster recovery database are located in different data centers.

15. The lock arbitration server according to claim 13, further comprising:

25 a first receiving module, configured to: after the primary database is restored, receive a lock request sent by an original primary database, wherein

the first processing module is further configured to control the lock to be held by the original primary database after the lock expires, wherein the original primary

database is restored to the primary database, and a current primary database is restored to one of the secondary databases.

16. The lock arbitration server according to claim 15, wherein

the first determining module is further configured to continue to determine
5 whether a lock held by the current primary database expires;

the second determining module is further configured to: in response to determining that the lock held by the current primary database has expired, determine whether lock requests of the original primary database, the current primary database, and other secondary databases are received;

10 a second processing module is configured to: in response to determining that the lock request of the original primary database is received, restore the original primary database to the primary database, and restore the current primary database to one of the secondary databases;

a third processing module is configured to: in response to determining that the
15 lock request of the original primary database is not received and the lock requests of the current primary database and the other secondary databases are received, maintain the current primary database as the primary database; and

a fourth processing module is configured to: in response to determining that the lock requests of the original primary database and the current primary database are
20 not received and the lock requests of the other secondary databases are received, select one of the other secondary databases as the primary database, and restore the current primary database to one of the secondary databases.

17. A primary database server, comprising:

a second sending module, configured to send a lock renewal request to a lock
25 arbitration server when a lock held by a primary database expires, wherein the lock is updated to extend a validity period of the lock before the validity period expires, wherein:

the lock is updated during an update cycle having a first time period T1,

the primary database sends, at a cycle with a second time period T2, a lock
30 request for querying a status of the lock, and

a particular secondary database from the secondary databases sends, at a cycle with a third time period T3, a lock request for querying the status of the lock, wherein the second time period T2 is smaller than the first time period T1, and the third time period T3 is greater than or equal to the first time period T1;

wherein the lock arbitration server selects the particular secondary database as a new primary database when the lock arbitration server does not receives the lock renewal request after the lock has expired; and

- 5 a switching module, configured to control the primary database to be switched to the secondary database.

18. The primary database server according to claim 17, wherein a priority of the primary database is higher than a priority of the particular secondary database.

19. The primary database server according to claim 17, wherein
the particular secondary database comprises a hot standby database and a disaster
recovery database, the primary database and the hot standby database are located in a
same data center, and the primary database and the disaster recovery database are
5 located in different data centers.

20. The primary database server according to claim 18, wherein the primary
database server further comprises:

a third sending module, configured to: after the primary database is restored,
send a lock request to the lock arbitration server, so that the lock arbitration server
10 receives the lock request sent by an original primary database, and controls the lock to
be held by the original primary database after the lock expires, wherein the original
primary database is restored to the primary database, and a current primary database is
restored to one of the secondary databases.