

CLAIMS

What is claimed is:

1. A method for managing a primary and secondary databases, the primary database and the
 5 secondary databases sharing a lock and contending for ownership of the lock, the lock having
 a mutual exclusion feature so only one database can hold the lock at a same time, the one
 database holding the lock being the primary database, wherein the primary database is
 configured to continuously update the lock to extend a validity period of the lock before the
 validity period expires, and wherein the primary database cannot complete updating the lock
 10 when the primary database is performing an upgrade or maintenance operation within the
 validity period;

wherein the method is performed by a lock arbitration server and comprises:

determining (S110; S310), by a first determining module (510) of the lock arbitration server,
 15 whether the lock held by the primary database has expired, wherein the lock held by the primary
 database expires when the validity period expires before the lock is updated to extend the
 validity period; and

in response to determining that the lock held by the primary database has expired:

20

sending (S320), by a first sending module (540) of the lock arbitration server, a lock
 expiration notification to the secondary databases,

receiving (S330), by a record module (550) of the lock arbitration server, lock requests
 25 sent by the secondary databases based on the lock expiration notification, and recording a
 receiving time of the lock requests,

determining (S120; S340), by a second determining module (520) of the lock arbitration
 server, whether a lock renewal request of the primary database is received, and

30

in response to determining that the lock renewal request of the primary database is not
 received, selecting (S130; S350), by a first processing module (530) of the lock arbitration
 server, a secondary database of the secondary databases as a new primary database seizing the
 lock, the selecting (S130; S350) comprising selecting, as the new primary database, the

secondary database with the earliest receiving time, and controlling, by the first processing module, the primary database to be switched to the secondary database of the secondary databases.

5 2. The method according to claim 1, wherein a priority of the primary database is higher than a priority of the secondary databases.

3. The method according to claim 1, further comprising:
performing operations comprising at least one of maintenance and upgrade on the primary
10 database.

4. The method according to claim 3, further comprising:

determining whether the operations comprising at least one of the maintenance and the
15 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 to continue to serve a user.

20 5. The method according to claim 4, 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, so that the original primary
25 database is restored to the primary database, and a current primary database is restored to a
secondary database.

6. The method according to claim 5, wherein after the selecting the secondary database of the
secondary databases as the new primary database, the method further comprises:

30 continuing to determine, by the first determining module (510), whether a lock held by
the current primary database expires; and

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.

5 7. The method according to claim 6, further comprising:

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 a secondary database; or

10

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

15

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 the new primary database, and restoring the current primary database to a secondary database.

20

8. A lock arbitration server for managing a primary and secondary databases, the primary database and the secondary databases sharing a lock and contending for ownership of the lock, the lock having a mutual exclusion feature so only one database can hold the lock at a same time, the one database holding the lock being the primary database, wherein the primary database is configured to continuously update the lock to extend a validity period of the lock before the validity period expires, and wherein the primary database cannot complete updating the lock when the primary database is performing an upgrade or maintenance operation within the validity period;

25

wherein the lock arbitration server comprises:

30

a first determining module (510), configured to determine (S110; S310) whether the lock held by the primary database has expired, wherein the lock held by the primary database expires when the validity period expires before the lock is updated to extend the validity period;

a second determining module (520), configured to determine (S120; S340), in response to determining that the lock held by the primary database has expired, whether a lock renewal request of the primary database is received;

5 a first sending module (540), configured to send (S320) a lock expiration notification to the secondary databases after the first determining module determines that the lock held by the primary database has expired;

10 a record module (550) configured to receive (S330) lock requests sent by the secondary databases based on the lock expiration notification, and recording a receiving time of the lock requests; and

15 a first processing module (530), configured to, in response to determining that the lock renewal request of the primary database is not received:

select (S130; S350) a secondary database of the secondary databases as a new primary database seizing the lock, the selecting (S130; S350) comprising selecting, as the new primary database, the secondary database with the earliest receiving time, and

20 control the primary database to be switched to the secondary database of the secondary databases.

9. The lock arbitration server according to claim 8, wherein a priority of the primary database is higher than a priority of the secondary databases.

25

10. The lock arbitration server according to any one of claims 8 or 9, further configured to:

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

30

in response to determining that the operations comprising at least one of the maintenance and the upgrade on the primary database are completed, restore the primary database to continue to serve a user.

11. The lock arbitration server according to claim 10, further configured to, after the primary database is restored:

receive 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, so that the original primary database is restored to the primary database, and a current primary database is restored to a secondary database.

12. The lock arbitration server according to claim 11, further configured to, after selecting the secondary database of the secondary databases as the new primary database:

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

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.

13. The lock arbitration server according to claim 12, further 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 a secondary database; 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 the new primary database, and restoring the current primary database to a secondary database.

14. A system for managing a primary and secondary databases, the system comprising:

the primary database;

the secondary databases; and

a lock arbitration server according to any one of claims 8 to 13.