CLAIMS

What is claimed is:

1. A method for updating a database by using a two-phase commit distributed transaction, the method being applied to a participant, and comprising:

performing an update operation according to an instruction on updating a target database record in a current transaction after adding a write lock to the target database record;

using another transaction, in a transaction information record, whose state is to release the write lock in advance and whose update object involves the target database record as a predecessor transaction of the current transaction, and recording the predecessor transaction of the current transaction in the transaction information record;

writing a log of the update operation of the current transaction to a log buffer after receiving a phase 1 prepare request message sent by a coordinator, and sending a write lock advance release request to the coordinator after the writing succeeds;

releasing the write lock after receiving a write lock advance release response returned by the coordinator, and setting a state of the current transaction in the transaction information record to releasing the write lock in advance;

after persisting the log of the update operation of the current transaction in the log buffer succeeds, sending a phase 1 prepare response message of preparation success to the coordinator after all predecessor transactions of the current transaction are in a commit state or sending a phase 1 prepare response message of preparation failure to the coordinator after any predecessor transaction of the current transaction is in an abort state; and

setting the current transaction in the transaction information record to a commit state or an abort state based on an execution result of the current transaction in phase 2,

wherein the recording the predecessor transaction of the current transaction in the transaction information record includes: writing each predecessor transaction of the current transaction and a status of the predecessor transaction to an entry of the current transaction in the transaction information record, and recording, in an entry of each predecessor transaction of the current transaction, the current transaction as a successor transaction of the predecessor transaction; and

the setting the current transaction in the transaction information record to the commit state or the abort state based on the execution result of the current transaction in phase 2 includes: based on the execution result of the current transaction in a commit/abort phase, updating a state in the entry of the current transaction in the transaction information record into the commit state or the abort state, and updating a state of the current transaction in an entry of a successor transaction of the current transaction to the commit state or the abort state.

2. The method according to claim 1, further comprising: when the current transaction has no predecessor transaction, sending the phase 1 prepare response message of preparation success to the coordinator after the persisting the log of the update operation of the current transaction in the log buffer succeeds.

3. The method according to claim 1, wherein the write lock advance release response carries a commit timestamp generated by the coordinator;

the method further comprises: recording a commit timestamp of the current transaction in the transaction information record; and

the performing the update operation according to the instruction includes: when a difference between a timestamp of the current update operation and a commit timestamp of the predecessor transaction of the current transaction reaches or exceeds a first threshold, performing the update operation according to the instruction after the predecessor transaction is in the commit state or the abort state.

4. The method according to claim 1, wherein the write lock advance release response carries a commit timestamp generated by the coordinator;

the method further comprises: recording a commit timestamp of the current transaction in the transaction information record; and

the writing the log of the update operation of the current transaction to the log buffer includes: when a difference between a timestamp of writing the update operation of the current transaction to the log buffer and a commit timestamp of the predecessor transaction of the current transaction reaches or exceeds a second threshold, writing the log of the update operation of the current transaction to the log buffer after the predecessor transaction is in the commit state or the abort state.

5. The method according to claim 1, wherein the write lock advance release request carries a timestamp of writing the log of the update operation of the current transaction to the log buffer, for the coordinator to generate a commit timestamp of the current transaction.

6. The method according to claim 1, further comprising: persisting states of the predecessor transaction of the current transaction and the current transaction in the transaction information record.

7. An apparatus for updating a database by using a two-phase commit distributed transaction, the apparatus being applied to a participant, and comprising:

a processor; and

a memory storing instructions executable by the processor,

wherein the processor is configured to:

perform an update operation according to an instruction on updating a target database record in a current transaction after adding a write lock to the target database record;

use another transaction, in a transaction information record, whose state is to release the write lock in advance and whose update object involves the target database record as a predecessor transaction of the current transaction, and record the predecessor transaction of the current transaction in the transaction information record;

write a log of the update operation of the current transaction to a log buffer after receiving a phase 1 prepare request message sent by a coordinator, and send a write lock advance release request to the coordinator after the writing succeeds;

release the write lock after receiving a write lock advance release response returned by the coordinator, and set a state of the current transaction in the transaction information record to releasing the write lock in advance;

after persisting the log of the update operation of the current transaction in the log buffer succeeds, send a phase 1 prepare response message of preparation success to the coordinator after all predecessor transactions of the current transaction are in a commit state or send a phase 1 prepare response message of preparation failure to the coordinator after any predecessor transaction of the current transaction is in an abort state; and

set the current transaction in the transaction information record to a commit state or an abort state based on an execution result of the current transaction in phase 2,

wherein in recording the predecessor transaction of the current transaction in the transaction information record, the processor is further configured to: write each predecessor transaction of the current transaction and a status of the predecessor transaction to an entry of the current transaction in the transaction information record,

and record, in an entry of each predecessor transaction of the current transaction, the current transaction as a successor transaction of the predecessor transaction; and

in setting the current transaction in the transaction information record, the processor is further configured to: based on the execution result of the current transaction in a commit/abort phase, update a state in the entry of the current transaction in the transaction information record into the commit state or the abort state, and update a state of the current transaction in an entry of a successor transaction of the current transaction to the commit state or the abort state.

8. The apparatus according to claim 7, wherein the processor is further configured to: when the current transaction has no predecessor transaction, send the phase 1 prepare response message of preparation success to the coordinator after the persisting the log of the update operation of the current transaction in the log buffer succeeds.

9. The apparatus according to claim 7, wherein the write lock advance release response carries a commit timestamp generated by the coordinator;

the processor is further configured to:

record a commit timestamp of the current transaction in the transaction information record; and

when a difference between a timestamp of the current update operation and a commit timestamp of the predecessor transaction of the current transaction reaches or exceeds a first threshold, perform the update operation according to the instruction after the predecessor transaction is in the commit state or the abort state.

10. The apparatus according to claim 7, wherein the write lock advance release response carries a commit timestamp generated by the coordinator;

the processor is further configured to:

record a commit timestamp of the current transaction in the transaction information record; and

when a difference between a timestamp of writing the update operation of the current transaction to the log buffer and a commit timestamp of the predecessor transaction of the current transaction reaches or exceeds a second threshold, write the log of the update operation of the current transaction to the log buffer after the predecessor transaction is in the commit state or the abort state.

11. The apparatus according to claim 7, wherein the write lock advance release request carries a timestamp of writing the log of the update operation of the current

transaction to the log buffer, for the coordinator to generate a commit timestamp of the current transaction by using the timestamp.

12. The apparatus according to claim 7, wherein the processor is further configured to persist states of the predecessor transaction of the current transaction and the current transaction in the transaction information record.