CLAIMS

1. A data processing method based on distributed storage, executed by a block chain node and comprising:

detecting, after writing first block data of a first block into a local cache and a local storage, a first block processing space of a storage buffer pool associated with the local cache to obtain a first space detection result, the storage buffer pool comprising N pieces of block data to be stored, N being a non-negative integer, wherein the storage buffer pool and the local cache are both included in a storage layer of the block chain node;

writing, if the first space detection result indicates that a number of block data to be stored in the first block processing space does not reach a buffer block processing threshold of the first block processing space, the first block data into the storage buffer pool based on the N pieces of block data to be stored;

detecting a second block processing space of a storage processing pool associated with the storage buffer pool to obtain a second space detection result, the storage processing pool being configured to store M pieces of block data to be fed back associated with a distributed database, M being a non-negative integer, wherein the M pieces of block data to be fed back in the storage processing pool are block data that are currently being stored in the distributed database, but have not received stored feedback information returned by the distributed database; and

determining, if the second space detection result indicates that a number of block data to be fed back in the second block processing space does not reach a feedback block processing threshold of the second block processing space, block data to be combined in the storage buffer pool based on the number of the block data to be fed back and the feedback block processing threshold, performing duplication detection on the block data to be combined and the M pieces of block data to be fed back to determine data that are not duplicated as retained data, and writing the retained data into the distributed database.

2. The method according to claim 1, further comprising:

acquiring K1 transaction services in the first block in a case that an executive layer receives the first block transmitted by a consensus layer, the first block comprising first block header information and a transaction list associated with the K1 transaction services, K1 being a positive integer;

acquiring, for each of the K1 transaction services, a service contract for executing the transaction

service, and executing the transaction service through the service contract of the transaction service to obtain a transaction execution result of the transaction service;

using the first block header information, the transaction list, and the transaction execution result of each of the K1 transaction services as first block data of the first block; and

writing the first block data of the first block into the local cache, and writing the first block data of the first block into the local storage corresponding to the local cache.

3. The method according to claim 2, wherein the acquiring the service contract for executing the transaction service and executing the transaction service through the service contract of the transaction service to obtain the transaction execution result of the transaction service comprises:

acquiring, based on a contract calling address of the transaction service, a service contract for executing the transaction service;

searching, through the service contract of the transaction service, the local cache for target data associated with the transaction service;

using the target data as target read data of the transaction service if the target data associated with the transaction service is found from the local cache; and

executing the transaction service based on the target read data of the transaction service, to obtain a transaction execution result of the transaction service.

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

searching, through the service contract, the distributed database for the target data associated with the transaction service if the target data associated with the transaction service is not found from the local cache, and using the target data as the target read data of the transaction service.

5. The method according to claim 2, further comprising:

notifying the executive layer to execute K2 transaction services in a second block after writing the first block data into the local cache and the local storage, the second block being a next block of the first block, and the second block being obtained by packaging the K2 transaction services by the consensus layer, K2 being a positive integer.

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

receiving second block data of the second block transmitted by the executive layer after writing the first block data into the storage buffer pool, the second block data being obtained by the executive layer after executing the K2 transaction services;

performing, through a first refreshing thread, buffer refreshing on the first block processing space of the storage buffer pool to obtain a buffer refreshing result, and updating the first space detection result by using the buffer refreshing result; and

writing, if the updated first space detection result indicates that the number of block data to be stored in the first block processing space does not reach the buffer block processing threshold, the second block data into the storage buffer pool based on the block data to be stored in the first block processing space.

7. The method according to claim 1, before writing the first block data into the storage buffer pool, the method further comprising:

notifying an executive layer to wait for the first block data to be stored if the first space detection result indicates that the number of block data to be stored in the first block processing space has reached the buffer block processing threshold.

8. The method according to claim 1, wherein the writing, based on the N pieces of block data to be stored, the first block data into the storage buffer pool comprises:

acquiring the N pieces of block data to be stored from the storage buffer pool, and searching the N pieces of block data to be stored for block data to be stored that duplicates with the first block data;

using, if the block data to be stored that duplicates with the first block data is found from the N pieces of block data to be stored, the found block data to be stored as duplicated block data; and

replacing the duplicated block data with the first block data, using the replaced N pieces of block data to be stored as first combined data associated with the first block data, and writing the first combined data into the storage buffer pool.

9. The method according to claim 8, further comprising:

using, if the block data to be stored that duplicates with the first block data is not found from the

N pieces of block data to be stored, the first block data and the N pieces of block data to be stored as second combined data associated with the first block data, writing the second combined data into the storage buffer pool, and performing incremental processing on the number of block data to be stored in the first block processing space.

10. The method according to claim 1 or 8 or 9, further comprising:

using the combined data in the storage buffer pool as data to be stored; and

notifying the storage buffer pool to wait for the storage of the data to be stored if the second space detection result indicates that a number of block data to be fed back in the second block processing space has reached a feedback block processing threshold.

11. The method according to claim 1, wherein the data to be stored in the storage buffer pool comprises the N pieces of block data to be stored and the first block data, and the first block data has no data duplication with the N pieces of block data to be stored, and

the determining the block data to be combined from the storage buffer pool based on the number of the block data to be fed back and the feedback block processing threshold, performing duplication detection on the block data to be combined and the M pieces of block data to be fed back, and writing the data retained after duplication detection into the distributed database comprises:

determining a difference between the number of the block data to be fed back and the feedback block processing threshold;

determining the block data to be combined from the N pieces of block data to be stored and the first block data based on the number difference and a block height of the data to be stored in the storage buffer pool; performing duplication detection on the block data to be combined and the M pieces of block data to be fed back to obtain an duplication detection result;

writing the block data to be combined into the storage processing pool if the duplication detection result indicates that the M pieces of block data to be fed back has no data to be fed back that duplicates with the block data to be combined; and

using the block data to be combined and the M pieces of block data to be fed back in the storage processing pool as data retained after duplication detection, and writing the data retained after duplication detection into the distributed database.

12. The method according to claim 11, further comprising:

receiving a data completion amount of the target block data fed back by the distributed database, the target block data being block data to which data successfully written into the distributed database belongs in the data retained after the duplication detection; and

performing, based on the data completion amount, decremental processing on the amount of data to be stored corresponding to the target block data in a block storage state mapping table until the amount of data to be stored reaches a decrement threshold, and determining that a complete block corresponding to the target block data has been successfully written into the distributed database.

13. A data processing apparatus based on distributed storage, comprising:

a first detection module configured to detect, after writing first block data of a first block into a local cache and a local storage, a first block processing space of a storage buffer pool associated with the local cache to obtain a first space detection result, the storage buffer pool comprising N pieces of block data to be stored, N being a non-negative integer, wherein the storage buffer pool and the local cache are both included in a storage layer of the block chain node;

a first writing module configured to write, if the first space detection result indicates that a number of block data to be stored in the first block processing space does not reach a buffer block processing threshold of the first block processing space, the first block data into the storage buffer pool based on the N pieces of block data to be stored;

a second detection module configured to detect a second block processing space of a storage processing pool associated with the storage buffer pool to obtain a second space detection result, the storage processing pool being configured to store M pieces of block data to be fed back associated with a distributed database, M being a non-negative integer, wherein the M pieces of block data to be fed back in the storage processing pool are block data that are currently being stored in the distributed database, but have not received stored feedback information returned by the distributed database; and

a second writing module configured to determine, if the second space detection result indicates that a number of block data to be fed back in the second block processing space does not reach a feedback block processing threshold of the second block processing space, block data to be combined in the storage buffer pool based on the number of the block data to be fed back and the feedback block processing threshold, performing duplication detection on the block data to be combined and the M pieces of block data to be fed back to determine data that are not duplicated as retained data, and writing the retained data into the distributed database.

14. A computer device, comprising: a processor and a memory,

the processor being connected to the memory, the memory being configured to store a computer program, and the processor being configured to invoke the computer program to cause the computer device to perform the method according to any one of claims 1 to 12.

15. A computer-readable storage medium, storing a computer program, the computer program, applicable to be loaded and executed by a processor, causing a computer device comprising the processor to perform the method according to any one of claims 1 to 12.

16. A computer program product, comprising instructions, the instructions, when run on a computer, causing the computer to implement the method according to any one of claims 1 to 12.