CLAIMS

What is claimed is:

1. A data synchronization method of a distributed system, the distributed system comprising a plurality of nodes, and the data synchronization method comprising:

obtaining data synchronization history of adjacent nodes connected to a target node;

determining selection weight values of the adjacent nodes according to the data synchronization history, the selection weight values being used for measuring efficiency of the adjacent nodes synchronizing data to the target node;

selecting, according to the selection weight values of the adjacent nodes, a target adjacent node for synchronizing data to the target node; and

synchronizing target data from the target adjacent node with the target node;

wherein the distributed system comprises a blockchain system, the target data comprises block header data and complete block data, and

the selecting, according to the selection weight values of the adjacent nodes, a target adjacent node for synchronizing data to the target node, and synchronizing target data in the target adjacent node to the target node comprises:

selecting, according to the selection weight values of the adjacent nodes, a first target adjacent node for synchronizing the block header data to the target node, and synchronizing the block header data from the first target adjacent node; and

selecting, according to the selection weight values of the adjacent nodes after the block header data is synchronized, a second target adjacent node for synchronizing the complete block data to the target node, and synchronizing the complete block data from the second target adjacent node.

2. The method according to claim 1, wherein the selecting, according to the selection weight values of the adjacent nodes, a target adjacent node for synchronizing data to the target node comprises:

detecting whether the adjacent nodes connected to the target node have data that needs to be synchronized to the target node, to obtain candidate adjacent nodes that need to synchronize data to the target node; and

selecting, according to the selection weight values of the candidate adjacent nodes, the target adjacent node for synchronizing data to the target node.

3. The method according to claim 1, wherein the obtaining data synchronization history of adjacent nodes connected to a target node comprises:

detecting whether the adjacent nodes connected to the target node have data that needs to be synchronized to the target node, to obtain candidate adjacent nodes that need to synchronize data to the target node; and

obtaining data synchronization history of the candidate adjacent nodes.

4. The method according to claim 3, wherein the distributed system comprises a blockchain system, the target data comprises block data, and

the detecting whether the adjacent nodes connected to the target node have data that needs to be synchronized to the target node comprises:

obtaining a height of a first block header of a blockchain ledger in the target node;

obtaining a height of a second block header of the blockchain ledger in the adjacent nodes connected to the target node; and

comparing the height of the first block header with the height of the second block header to detect whether the adjacent nodes connected to the target node have data that needs to be synchronized to the target node.

5. The method according to claim 1, wherein the data synchronization history comprises error counts of the target node and the adjacent nodes in a historical data synchronization process, and the determining selection weight values of the adjacent nodes according to the data synchronization history comprises:

detecting adjacent nodes having error counts not exceeding an error count threshold; and

determining, according to the data synchronization history, selection weight values of the adjacent nodes having the error counts not exceeding the error count threshold.

6. The method according to claim 1, wherein the data synchronization history comprises a historical data transmission rate and a historical data request moment, and the determining selection weight values of the adjacent nodes according to the data synchronization history comprises: the selection weight values of the adjacent nodes are calculated by the following formula:

$$W_i = \frac{\sum_{j=m+1}^{j=m+n} S_j + \sum_{j=m+1}^{j=m+n} (t - t_j)}{n}$$

wherein W_i represents the selection weight value of the ith adjacent node; j represents the historical data synchronization between the target node and the ith adjacent node for the jth time; m represents the historical data synchronization between the target node and the ith adjacent node for the mth time; n represents the number of times of the historical data synchronization between the target node and the ith adjacent node in the last n times; S_j represents the historical data transmission rate when the target node and the ith adjacent node perform historical data synchronization for the jth time; t_j represents the historical data request moment at which the target node and the ith adjacent node perform the historical data synchronization for the jth time; and t represents the moment at which the selection weight value of the ith adjacent node is calculated.

7. The method according to claim 1, wherein the synchronizing target data in the target adjacent node to the target node comprises:

transmitting a download request for the target data to the target adjacent node, to cause the target adjacent node to transmit the target data to the target node according to the download request; and

wherein the method further comprises: in the process of synchronizing target data in the target adjacent node to the target node, when synchronization of the target data fails, recording an error count in the data synchronization between the target node and the target adjacent node.

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

before the selecting a second target adjacent node for synchronizing the complete block data to the target node,

obtaining a height of a third block header with the complete block data of a blockchain ledger in the target node;

obtaining a height of a fourth block header of the blockchain ledger in the target node; and

comparing the height of the third block header with the height of the fourth block header to detect whether to select the second target adjacent node for synchronizing the complete

block data to the target node.

9. The method according to claim 1, the synchronizing the complete block data from the second target adjacent node comprising:

downloading the complete block data in the second target adjacent node;

verifying validity of the complete block data; and

adding valid complete block data to a block data buffer of the target node and synchronizing the complete block data to the target node according to the data stored in the block data buffer.

10. The method according to claim 9, the synchronizing the complete block data to the target node according to the data stored in the block data buffer comprising:

reading, according to a block sequence indicated by a blockchain ledger in the target node, corresponding block data from the block data buffer and writing the corresponding block data into the blockchain ledger in the target node.

11. A data synchronization apparatus of a distributed system, the apparatus comprising:

an obtaining module, configured to obtain data synchronization history of adjacent nodes connected to a target node;

a determining module, configured to determine selection weight values of the adjacent nodes according to the data synchronization history, the selection weight values being used for measuring efficiency of the adjacent nodes synchronizing data to the target node;

a selection module, configured to select, according to the selection weight values of the adjacent nodes, a target adjacent node for synchronizing data to the target node; and

a synchronization module, configured to synchronize target data from the target adjacent node with the target node;

wherein the distributed system comprises a blockchain system, the target data comprises block header data and complete block data, and the selection module is configured to: select, according to the selection weight values of the adjacent nodes, a first target adjacent node for synchronizing the block header data to the target node, and synchronizing the block header data from the first target adjacent node; and select, according to the selection weight values of the adjacent nodes after the block header data is synchronized, a second target adjacent node for synchronizing the complete block data to the target node, and synchronizing the complete

block data from the second target adjacent node.

12. The apparatus according to claim 11, wherein the selection module is configured to: detect whether the adjacent nodes connected to the target node have data that needs to be synchronized to the target node, to obtain candidate adjacent nodes that need to synchronize data to the target node; and select, according to the selection weight values of the candidate adjacent nodes, the target adjacent node for synchronizing data to the target node.

13. The apparatus according to claim 11, wherein the obtaining module is configured to: detect whether the adjacent nodes connected to the target node have data that needs to be synchronized to the target node, to obtain candidate adjacent nodes that need to synchronize data to the target node; and obtain data synchronization history of the candidate adjacent nodes.

14. The apparatus according to claim 13, wherein the distributed system comprises a blockchain system, the target data comprises block data, and the detecting module is configured to: obtain a height of a first block header of a blockchain ledger in the target node; obtain a height of a second block header of the blockchain ledger in the adjacent nodes connected to the target node; and compare the height of the first block header with the height of the second block header to detect whether the adjacent nodes connected to the target node have data that needs to be synchronized to the target node.

15. The apparatus according to claim 11, wherein the data synchronization history comprises error counts of the target node and the adjacent nodes in a historical data synchronization process, and the determining module is configured to: detect adjacent nodes having error counts not exceeding an error count threshold; and determine, according to the data synchronization history, selection weight values of the adjacent nodes having the error counts not exceeding the error count threshold.

16. The apparatus according to claim 11, wherein the data synchronization history comprises a historical data transmission rate and a historical data request moment, and the determining module is configured to: calculate the selection weight values of the adjacent nodes by the following formula:

$$W_i = \frac{\sum_{j=m+1}^{j=m+n} S_j + \sum_{j=m+1}^{j=m+n} (t - t_j)}{n}$$

wherein W_i represents the selection weight value of the ith adjacent node; j represents the historical data synchronization between the target node and the ith adjacent node for the jth time; m represents the historical data synchronization between the target node and the ith adjacent node for the mth time; n represents the number of times of the historical data synchronization between the target node and the ith adjacent node in the last n times; S_j represents the historical data transmission rate when the target node and the ith adjacent node perform historical data synchronization for the jth time; t_j represents the historical data request moment at which the target node and the ith adjacent node perform the historical data synchronization for the jth time; and t represents the moment at which the selection weight value of the ith adjacent node is calculated.

17. The apparatus according to claim 11, wherein the synchronizing module is configured to: transmit a download request for the target data to the target adjacent node, to cause the target adjacent node to transmit the target data to the target node according to the download request; and the synchronizing module is further configured to: in the process of synchronizing target data in the target adjacent node to the target node, when synchronization of the target data fails, record an error count in the data synchronization between the target node and the target adjacent node.

18. The apparatus according to claim 11, wherein the selecting module is configured to: before the selecting a second target adjacent node for synchronizing the complete block data to the target node, obtain a height of a third block header with the complete block data of a blockchain ledger in the target node; obtain a height of a fourth block header of the blockchain ledger in the target node; and compare the height of the third block header with the height of the fourth block header to detect whether to select the second target adjacent node for synchronizing the complete block data to the target node.

19. The apparatus according to claim 11, wherein the synchronizing module is configured to: download the complete block data in the second target adjacent node; verify validity of the complete block data; and add valid complete block data to a block data buffer of the target node and synchronizing the complete block data to the target node according to the data stored in the block data buffer.

20. The apparatus according to claim 19, wherein the synchronizing module is configured to: read, according to a block sequence indicated by a blockchain ledger in the target node, corresponding block data from the block data buffer and writing the

corresponding block data into the blockchain ledger in the target node.

21.A computer-readable storage medium, storing a computer program, the computer program comprising executable instructions, and the executable instructions, when executed by a processor, implementing the method according to any one of claims 1 to 10.

22. An electronic device, comprising:

a processor; and

a memory, configured to store executable instructions of the processor,

the processor being configured to execute the executable instructions to implement the method according to any one of claims 1 to 10.