CLAIMS

What is claimed is:

1. A blockchain-based data processing method, performed by a computer device, comprising:

acquiring a cache application instruction for a target custom contract, and creating a cache region associated with the target custom contract in a total cache region according to custom cache creation parameters in the cache application instruction, the total cache region comprising at least two cache regions, wherein each cache region stores different types of cached contract data, and a cache eviction mechanism for each cache region in the total cache region being independent of each other, the cache eviction mechanism releasing partial cached contract data after a target cache region is determined;

receiving to-be-cached contract data, including a to-be-cached capacity of the to-be cached contract data, corresponding to the target custom contract;

searching for the cache region associated with the target custom contract in the at least two cache regions;

taking the cache region associated with the target custom contract as a target cache region;

determining an available cache capacity of the target cache region;

evicting cached contract data in the target cache region according to the cache eviction mechanism for the target cache region when the available cache capacity of the target cache region is less than the to-be-cached capacity of the to-be-cached contract data;

obtaining a new available cache capacity, the new available cache capacity being greater than the to-be-cached capacity; and

storing the to-be-cached contract data into the target cache region according to the new available cache capacity.

2. The method according to claim 1, wherein acquiring a cache application instruction for a target custom contract comprises:

acquiring a cache application request, the cache application request comprising a cache application contract identifier and custom cache creation parameters associated with a target custom contract;

invoking a cache application contract indicated by the cache application contract identifier in the cache application request; and

executing, in a contract virtual machine, the cache application contract according to the custom cache creation parameters to obtain a cache application instruction for the target custom contract, the cache application instruction comprising the custom cache creation parameters.

3. The method according to claim 1, wherein:

creating the cache region associated with the target custom contract in the total cache region according to the custom cache creation parameters in the cache application instruction is performed when it is verified that the cache application instruction satisfies a cache application condition,

verifying whether the cache application instruction satisfies the cache application condition comprises:

acquiring one or more cache approval results for the cache application instruction through a cache approval component when a consensus on the cache application instruction is reached in a consensus network, each cache approval result being data on which a consensus is reached in the consensus network; and

determining that the cache application instruction satisfies the cache application condition when among the one or more cache approval results, a number of cache approval results indicating that cache approval succeeds is greater than an approval threshold value.

4. The method according to claim 1, wherein the custom cache creation parameters comprise a contract identifier and a pre-allocated cache capacity of the target custom contract; and

creating a cache region associated with the target custom contract in a total cache region according to custom cache creation parameters in the cache application instruction comprises:

acquiring a pre-allocated cache region corresponding to the pre-allocated cache capacity from a total cache region; and

mapping the pre-allocated cache region and the contract identifier of the target custom contract to obtain a cache region associated with the target custom contract.

5. The method according to claim 1, wherein evicting cached contract data in the target cache region according to the cache eviction mechanism for the target cache region to obtain a new available cache capacity comprises:

acquiring a first access record list corresponding to the target cache region, the first access record list comprising a recent data access time respectively corresponding to at least two pieces of cached contract data, and the target cache region comprising the at least two pieces of cached contract data;

taking cached contract data whose recent data access time is the earliest as to-be-evicted target contract data in the target cache region; and

releasing the to-be-evicted target contract data, and taking an available cache capacity of the target cache region from which the to-be-evicted target contract data is released as a new available cache capacity.

6. The method according to claim 1, wherein evicting cached contract data in the target cache region according to the cache eviction mechanism for the target cache region to obtain a new available cache capacity comprises:

acquiring a second access record list corresponding to the target cache region, the second access record list comprising a number of data accesses respectively corresponding to at least two pieces of cached contract data within a target time period, and the target cache region comprising the at least two pieces of cached contract data;

taking cached contract data with the least number of data accesses as to-be-evicted target contract data in the target cache region; and

releasing the to-be-evicted target contract data, and taking an available cache capacity of the target cache region from which the to-be-evicted target contract data is released as a new available cache capacity.

7. The method according to claim 1, wherein evicting cached contract data in the target cache region according to the cache eviction mechanism for the target cache region to obtain a new available cache capacity comprises:

acquiring a data storage record list corresponding to the target cache region, the data storage record list comprising a data cache time respectively corresponding to at least two pieces of cached contract data, and the target cache region comprising the at least two pieces of cached contract data;

taking cached contract data with an earliest data cache time as to-be-evicted target contract data in the target cache region; and

releasing the to-be-evicted target contract data, and taking an available cache capacity of the target cache region from which the to-be-evicted target contract data is released as a new available cache capacity.

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

receiving a data query request through a cache routing component, the data query request comprising a contract identifier of a to-be-executed contract; and

searching for queried cached data associated with the contract identifier of the to-be-executed contract in the at least two cache regions through the cache routing component.

9. The method according to claim 8, wherein the at least two cache regions comprise a general-purpose contract cache region and N custom contract cache regions; the N custom contract cache regions comprise the target cache region; N is an integer greater than or equal to 0; the general-purpose contract cache region comprises cached contract data corresponding to one or more general-purpose contracts; cached contract data stored in one custom contract cache region corresponds to a custom contract associated with the custom contract cache region; and

searching for queried cached data associated with the contract identifier of the to-be-executed contract in the at least two cache regions through the cache routing component comprises:

acquiring custom contract identifiers corresponding to custom contracts that are associated with the N custom contract cache regions through the cache routing component;

searching for the contract identifier of the to-be-executed contract in the custom contract identifiers;

searching for queried cached data associated with the contract identifier of the to-be-executed contract in a custom contract cache region that is associated with a custom contract corresponding to the contract identifier of the to-be-executed contract through the cache routing component when a custom contract identifier that is the same as the contract identifier of the to-be-executed contract is found out from the custom contract identifiers; and

searching for queried cached data associated with the contract identifier of the to-be-executed contract in the general-purpose contract cache region through the cache routing component when the custom contract identifier that is the same as the contract identifier of the to-be-executed contract is not found out from the custom contract identifiers.

10. The method according to claim 8, wherein:

the data query request further comprises a to-be-executed transaction for invoking the to-be-executed contract;

the at least two cache regions further comprise a block cache region and a transaction cache region;

the block cache region is used for storing cached block data corresponding to a to-be-uploaded block that is not written into a blockchain ledger;

the transaction cache region is used for storing cached transaction data corresponding to a to-be-uploaded transaction that is not written into the blockchain ledger; and

searching for queried cached data associated with the contract identifier of the to-be-executed contract in the at least two cache regions through the cache routing component comprises:

traversing, in the cache routing component, cached transaction data in the transaction cache region according to the contract identifier of the to-be-executed contract and the to-be-executed transaction;

taking traversed cached transaction data as queried cached data when cached transaction data associated with the contract identifier of the to-be-executed contract and the to-be-executed transaction is traversed in the transaction cache region;

traversing, in the cache routing component, cached block data in the block cache region according to the contract identifier of the to-be-executed contract and the to-be-executed transaction when the cached transaction data associated with the contract identifier of the to-be-executed contract and the to-be-executed transaction is not traversed in the transaction cache region;

taking traversed cached block data as queried cached data when cached block data associated with the contract identifier of the to-be-executed contract and the to-be-executed transaction is traversed in the block cache region; and

acquiring queried cached data associated with the contract identifier of the to-be-executed contract from a blockchain database when the cached block data associated with the contract identifier of the to-be-executed contract and the to-be-executed transaction is not traversed in the block cache region.

11. The method according to claim 10, wherein the to-be-uploaded block comprises one or more to-be-uploaded transactions; and

the method further comprises:

releasing cached block data corresponding to the to-be-uploaded block from the block cache region when the to-be-uploaded block is written into the blockchain ledger; and

releasing cached transaction data corresponding to the one or more to-be-uploaded transactions in the to-be-uploaded block from the transaction cache region.

12. The method according to claim 8, wherein:

the at least two cache regions further comprise a system cache region;

the system cache region comprises cached contract data respectively corresponding to one or more system contracts; and

searching for queried cached data associated with the contract identifier of the to-be-executed contract in the at least two cache regions through the cache routing component comprises:

acquiring system contract identifiers corresponding to the one or more system contracts;

searching for the contract identifier of the to-be-executed contract in the system contract identifiers; and

searching for queried cached data associated with the contract identifier of the to-be-executed contract in the system cache region that is associated with a system contract corresponding to the contract identifier of the to-be-executed contract through the cache routing component when a system contract identifier that is the same as the contract identifier of the to-be-executed contract is found out from the system contract identifiers.

13. A non-transitory computer readable medium storing a plurality of instructions, wherein the plurality of instructions, when executed by a processor, configure the processor to:

acquire a cache application instruction for a target custom contract;

create a cache region associated with the target custom contract in a total cache region according to custom cache creation parameters in the cache application instruction, the total cache region comprising at least two cache regions, wherein each cache region stores different types of cached contract data, and a cache eviction mechanism for each cache region in the total cache region being independent of each other, the cache eviction mechanism releasing partial cached contract data after a target cache region is determined;

receive to-be-cached contract data, including a to-be-cached capacity of the to-be cached contract data, corresponding to the target custom contract;

search for the cache region associated with the target custom contract in the at least two cache regions;

take the cache region associated with the target custom contract as a target cache region;

determine an available cache capacity of the target cache region;

evict cached contract data in the target cache region according to the cache eviction mechanism for the target cache region when the available cache capacity of the target cache region is less than the to-be-cached capacity of the to-be-cached contract data;

obtain a new available cache capacity, the new available cache capacity being greater than the to-be-cached capacity; and

store the to-be-cached contract data into the target cache region according to the new available cache capacity.

14. The non-transitory computer readable medium according to claim 13, wherein the plurality of instructions further configures the processor to:

acquire a cache application request, the cache application request comprising a cache application contract identifier and the custom cache creation parameters associated with the target custom contract;

invoke a cache application contract indicated by the cache application contract identifier in the cache application request; and

execute, in a contract virtual machine, the cache application contract according to the custom cache creation parameters to obtain a cache application instruction for the target custom contract, the cache application instruction comprising the custom cache creation parameters.

15. The non-transitory computer readable medium according to claim 13, wherein the plurality of instructions further configures the processor to:

acquire one or more cache approval results for the cache application instruction through a cache approval component when a consensus on the cache application instruction is reached in a consensus network, each cache approval result being data on which a consensus is reached in the consensus network; and

determine that the cache application instruction satisfies a cache application condition when among the one or more cache approval results, a number of cache approval results indicating that cache approval succeeds exceeds an approval threshold value, and

create the cache region associated with the target custom contract in the total cache region according to the custom cache creation parameters in the cache application instruction when the cache application instruction satisfies the cache application condition.

16. The non-transitory computer readable medium according to claim 13, wherein the custom cache creation parameters comprise a contract identifier and a pre-allocated cache capacity of the target custom contract; and

the plurality of instructions further configures the processor to:

acquire a pre-allocated cache region corresponding to the pre-allocated cache capacity from the total cache region; and

map the pre-allocated cache region and the contract identifier of the target custom contract to obtain a cache region associated with the target custom contract.

17. The non-transitory computer readable medium according to claim 13, wherein the plurality of instructions further configures the processor to:

receive a data query request through a cache routing component, the data query request comprising a contract identifier of a to-be-executed contract; and

search for queried cached data associated with the contract identifier of the to-be-executed contract in the at least two cache regions through the cache routing component.

18. A computer device, comprising:

a processor;

a network interface in communication with the processor; and

a memory in communication with the processor and storing a plurality of instructions, the plurality of instructions, when executed by the processor, configure the processor to:

acquire a cache application instruction for a target custom contract;

create a cache region associated with the target custom contract in a total cache region according to custom cache creation parameters in the cache application instruction, the total cache region comprising at least two cache regions, wherein each cache region stores different types of cached contract data, and a cache eviction mechanism for each cache region in the total cache region being independent of each other, the cache eviction mechanism releasing partial cached contract data after a target cache region is determined;

receive to-be-cached contract data, including a to-be-cached capacity of the to-be cached contract data, corresponding to the target custom contract;

search for the cache region associated with the target custom contract in the at least two cache regions;

take the cache region associated with the target custom contract as a target cache region;

determine an available cache capacity of the target cache region;

evict cached contract data in the target cache region according to the cache eviction mechanism for the target cache region when the available cache capacity of the target cache region is less than the to-be-cached capacity of the to-be-cached contract data;

obtain a new available cache capacity, the new available cache capacity being greater than the to-be-cached capacity; and

store the to-be-cached contract data into the target cache region according to the new available cache capacity.

19. The computer device according to claim 18, wherein the plurality of instructions further configures the processor to:

acquire a cache application request, the cache application request comprising a cache application contract identifier and the custom cache creation parameters associated with the target custom contract;

invoke a cache application contract indicated by the cache application contract identifier in the cache application request; and

execute, in a contract virtual machine, the cache application contract according to the custom cache creation parameters to obtain a cache application instruction for the target custom contract, the cache application instruction comprising the custom cache creation parameters.

20. The computer device according to claim 18, wherein the plurality of instructions further configures the processor to:

acquire one or more cache approval results for the cache application instruction through a cache approval component when a consensus on the cache application instruction is reached in a consensus network, each cache approval result being data on which a consensus is reached in the consensus network; and

determine that the cache application instruction satisfies a cache application condition when among the one or more cache approval results, a number of cache approval results indicating that cache approval succeeds exceeds an approval threshold value, and

create the cache region associated with the target custom contract in the total cache region according to the custom cache creation parameters in the cache application instruction when the cache application instruction satisfies the cache application condition.