CLAIMS

1. A method of executing a smart contract, comprising:

receiving, by a blockchain node, a request for creating a smart contract, wherein the request comprises a bytecode of the smart contract;

starting, by the blockchain node when the blockchain node completes deployment of the smart contract, to turn a bytecode of the smart contract into a machine code through Just-In-Time JIT compilation and store the machine code; and

when the blockchain node is to execute the smart contract deployed, the machine code of the smart contract is stored locally and a number of blockchain nodes storing the machine code of the smart contract in a blockchain network is not less than a predetermined number, executing, by the blockchain node, the machine code, wherein a number of blockchain nodes in the blockchain network is not less than 3f+1, the predetermined number is not less than 2f+1, and the f is a positive integer.

2. The method according to claim 1, wherein starting, by the blockchain node when the blockchain node completes deployment of the smart contract, to turn the bytecode of the smart contract into the machine code through JIT compilation, comprises:

starting, by the blockchain node when the blockchain node completes creation of the smart contract, to perform JIT compilation on the bytecode of the smart contract; or,

using, by the blockchain node when the blockchain node completes the creation of the smart contract, relatively free periods of time to perform JIT compilation on the bytecode of the smart contract; or

performing, by the blockchain node when the blockchain node receives a transaction request to invoke the smart contract created, JIT compilation on the bytecode of the smart contract.

3. The method according to claim 1, wherein turning the bytecode of the smart contract into the machine code through JIT compilation and storing the machine code, comprises:

identifying, by the blockchain node, in a process of translating and/or executing the bytecode of the smart contract, a hotspot bytecode, and performing compilation and storing; and/or,

performing, by the blockchain node, in the process of translating and/or executing the bytecode of the smart contract, optimized compilation on the bytecode and storing.

4. The method according to claim 3, wherein granularity of identifying the hostpot bytecode comprises a function level or a code block level.

5. The method according to claim 1, wherein executing the machine code by the blockchain node comprises:

executing, by the blockchain node, a machine code corresponding to an invoked function and/or a code block in the smart contract stored.

6. The method according to claim 1, wherein when the blockchain node is to execute the smart contract deployed, no machine code of the smart contract is stored locally, or the number of blockchain nodes storing the machine code of the smart contract in the blockchain network is less than the predetermined number, interpreting and executing, by the blockchain node, the bytecode of the smart contract.

7. The method according to claim 6, wherein interpreting and executing, by the blockchain node, the bytecode of the smart contract, comprises:

interpreting and executing, by the blockchain node, a bytecode corresponding to an invoked function in the smart contract.

8. The method according to claim 6, wherein when the blockchain node is to execute the smart contract deployed, no machine code of the smart contract is stored locally, interpreting and executing, by the blockchain node, the bytecode of the smart contract, comprises:

when the blockchain node is to execute the smart contract deployed, no machine code of the smart contract is stored locally; further querying, by the blockchain node, current status of JIT compilation for the bytecode of the smart contract, and in a case that the status is under JIT compilation, interpreting and executing, by the blockchain node, the bytecode of the smart contract;

or,

when the blockchain node is to execute the smart contract deployed, no machine code of the smart contract is stored locally; further querying, by the blockchain node, current status of JIT compilation for the bytecode of the smart contract, and in a case that the JIT compilation is not started, starting the JIT compilation, and interpreting and executing the bytecode of the smart contract.

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

reporting, by the blockchain node when the blockchain node is to reach a consensus on a

transaction invoking the smart contract deployed, the compilation progress of the local JIT compilation against the smart contract deployed to other blockchain nodes in the blockchain network, and obtaining, by the blockchain node, the compilation progress reported by the other blockchain nodes; and

determining, by the blockchain node, a number of blockchain nodes storing the machine code in the blockchain network according to the compilation progress of the blockchain node and the compilation progress reported by the other blockchain nodes.

10. The method according to claim 9, wherein reporting, by the blockchain node when the blockchain node is to reach a consensus on the transaction invoking the smart contract deployed, the compilation progress of the local JIT compilation against the smart contract deployed to the other blockchain nodes in the blockchain network, comprises:

if a consensus mechanism based on a practical Byzantine fault-tolerant PBFT algorithm is adopted, and the blockchain node is a master node, sending, by the blockchain node, a pre-prepare message to the other blockchain nodes in a pre-prepare phase, wherein the pre-prepare message contains the compilation progress of the blockchain node; and if a consensus mechanism based on the PBFT algorithm is adopted, and the blockchain node is a slave node, sending, by the blockchain node, a prepare message to the other blockchain nodes in a prepare phase, wherein the prepare message contains the compilation progress of the blockchain node; and/or,

if a consensus mechanism based on the PBFT algorithm is adopted, sending, by the blockchain node, a commit message to the other blockchain nodes in a commit phase, wherein the commit message contains the compilation progress of the blockchain node.

11. The method according to any one of claims 1 to 10, wherein the storing is caching.

12. A method of executing a smart contract, comprising:

starting, by a blockchain node when the blockchain node deployed the smart contract, to turn a bytecode of the smart contract into a machine code through Just-In-Time JIT compilation and store the machine code; and

when the blockchain node is to execute the smart contract deployed, the machine code of the smart contract is stored locally, and a number of blockchain nodes storing the machine code of the smart contract in a blockchain network is not less than a predetermined number, executing, by the blockchain node, the machine code, wherein a number of blockchain nodes in the blockchain network is not less than 3f+1, the predetermined number is not less than

2f+1, and the f is a positive integer.

13. The method according to claim 12, wherein starting, by the blockchain node when the blockchain node deployed the smart contract, to turn the bytecode of the smart contract into the machine code through JIT compilation, comprises:

starting, by the blockchain node when the blockchain node completes creation of the smart contract, to perform JIT compilation on the bytecode of the smart contract; or,

using, by the blockchain node when the blockchain node completes the creation of the smart contract, relatively free periods of time to perform JIT compilation on the bytecode of the smart contract; or

performing, by the blockchain node when the blockchain node receives a transaction request to invoke the smart contract created, JIT compilation on the bytecode of the smart contract.

14. The method according to claim 12, wherein turning the bytecode of the smart contract into the machine code through JIT compilation and storing the machine code, comprises:

identifying, by the blockchain node, in a process of translating and/or executing the bytecode of the smart contract, a hotspot bytecode, and performing compilation and storing;

and/or,

performing, by the blockchain node, in the process of translating and/or executing the bytecode of the smart contract, optimized compilation on the bytecode and storing.

15. The method according to claim 14, wherein granularity of identifying the hostpot bytecode comprises a function level or a code block level.

16. The method according to claim 12, wherein executing, by the blockchain node, the machine code, comprises:

executing, by the blockchain node, a machine code corresponding to an invoked function and/or a code block in the smart contract stored.

17. The method according to claim 12, wherein when the blockchain node is to execute the smart contract deployed, no machine code of the smart contract is stored locally, or the number of blockchain nodes storing the machine code of the smart contract in the blockchain network is less than the predetermined number, interpreting and executing, by the blockchain node, the bytecode of the smart contract.

18. The method according to claim 17, wherein interpreting and executing, by the blockchain node, the bytecode of the smart contract, comprises:

interpreting and executing, by the blockchain node, a bytecode corresponding to an invoked function in the smart contract.

19. The method according to claim 17, wherein when the blockchain node is to execute the smart contract deployed, no machine code of the smart contract is stored locally, interpreting and executing, by the blockchain node, the bytecode of the smart contract, comprises:

when the blockchain node is to execute the smart contract deployed, no machine code of the smart contract is stored locally; further querying, by the blockchain node, current status of JIT compilation for the bytecode of the smart contract, and in a case that the status is under JIT compilation, interpreting and executing, by the blockchain node, the bytecode of the smart contract;

or,

when the blockchain node is to execute the smart contract deployed, no machine code of the smart contract is stored locally; further querying, by the blockchain node, current status of JIT compilation for the bytecode of the smart contract, and in a case that the JIT compilation is not started, starting the JIT compilation, and interpreting and executing the bytecode of the smart contract.

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

reporting, by the blockchain node when the blockchain node is to reach a consensus on a transaction invoking the smart contract deployed, the compilation progress of the local JIT compilation against the smart contract deployed to other blockchain nodes in the blockchain network, and obtaining, by the blockchain node, the compilation progress reported by the other blockchain nodes; and

determining, by the blockchain node, whether all blockchain nodes in the blockchain network store the machine code corresponding to the bytecode of the smart contract according to the compilation progress of the blockchain node and the compilation progress reported by the other blockchain nodes.

21. The method according to claim 20, wherein reporting, by the blockchain node when the blockchain node is to reach a consensus on the transaction invoking the smart contract

deployed, the compilation progress of the local JIT compilation against the smart contract deployed to the other blockchain nodes in the blockchain network, comprises:

if a consensus mechanism based on a practical Byzantine fault-tolerant PBFT algorithm is adopted, and the blockchain node is a master node, sending, by the blockchain node, a pre-prepare message to the other blockchain nodes in a pre-prepare phase, wherein the pre-prepare message contains the compilation progress of the blockchain node; and if a consensus mechanism based on the PBFT algorithm is adopted, and the blockchain node is a slave node, sending, by the blockchain node, a prepare message to the other blockchain nodes in a prepare phase, wherein the prepare message contains the compilation progress of the blockchain node; and/or,

if a consensus mechanism based on the PBFT algorithm is adopted, sending, by the blockchain node, a commit message to the other blockchain nodes in a commit phase, wherein the commit message contains the compilation progress of the blockchain node.

22. The method according to any one of claims 12 to 21, wherein the storing is caching.

23. A blockchain node for executing a smart contract, configured to execute the method according to any one of claims 1 to 11, or, 12 to 22.

24. A blockchain node for executing a smart contract, comprising:

a processor, and

a memory with a program stored therein, wherein the method according to any one of claims 1 to 11, or, 12 to 22 is executed when the processor executes the program.

25. A storage medium for storing a program, wherein when the program is executed, the operation according to any one of claims 1 to 11, or 12 to 22 is performed.