CLAIMS

What is claimed is:

1. A method for 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;

compiling, by the blockchain node, all of the bytecode of the smart contract into first machine code by using a global compilation method comprising Ahead of Time, AoT, compilation after deploying the smart contract, and storing the first machine code, wherein the compiling, by the blockchain node, the bytecode of the smart contract into first machine code by using a global compilation method after deploying the smart contract comprises globally compiling all of the bytecode of the smart contract in an idle time period of the blockchain node after creating the smart contract; and

locally compiling without waiting for completion of the global compilation method, by the blockchain node when executing the smart contract, the bytecode of the smart contract by using a local compilation method comprising Just-In-Time, JIT, compilation to obtain second machine code if the first machine code of the smart contract is not locally stored and the bytecode of the smart contract is being globally compiled, storing the second machine code, and executing the second machine code.

2. The method according to claim 1, wherein the compiling the bytecode of the smart contract into first machine code by using a global compilation method comprises:

performing optimized compilation in a process of globally compiling the bytecode of the smart contract to obtain the first machine code.

3. The method according to claim 1 or 2, wherein the locally compiling, by the blockchain node when executing the smart contract, the bytecode of the smart contract if the first machine code of the smart contract is not locally stored specifically comprises:

when the blockchain node executes the smart contract, if the first machine code of the smart contract is not locally stored, further querying, by the blockchain node, a current global compilation status of the bytecode of the smart contract, and locally

compiling the bytecode of the smart contract if the bytecode of the smart contract is being globally compiled; or

when the blockchain node executes the smart contract, if the first machine code of the smart contract is not locally stored, further querying, by the blockchain node, a current global compilation status of the bytecode of the smart contract, and if global compilation has not been started, starting global compilation, and locally compiling the bytecode of the smart contract.

4. The method according to claim 1 or 2, wherein the locally compiling, by the blockchain node, the bytecode of the smart contract comprises:

locally compiling, by the blockchain node, bytecode corresponding to a function to be invoked in the smart contract.

5. The method according to claim 1 or 3, wherein the locally compiling the bytecode of the smart contract to obtain second machine code, and storing the second machine code comprises:

identifying second machine code that serves as a hotspot in a process of translating and/or executing the bytecode of the smart contract, and storing the second machine code that serves as a hotspot for subsequent invocation; and/or

performing optimized compilation to obtain the second machine code in the process of translating and/or executing the bytecode of the smart contract, and storing the second machine code.

6. The method according to claim 1, wherein when executing the smart contract, the blockchain node executes the first machine code if the first machine code of the smart contract is locally stored.

7. The method according to claim 6, wherein the executing, by the blockchain node, the first machine code comprises:

executing, by the blockchain node, stored first machine code corresponding to a function to be invoked in the smart contract.

8. The method according to any one of the preceding claims, further comprising:

interpreting and executing, by the blockchain node when executing the smart contract, the bytecode of the smart contract if the first machine code of the smart contract is not locally stored.

9. The method according to claim 8, wherein the interpreting and executing, by the blockchain node when executing the smart contract, the bytecode of the smart contract if the first machine code of the smart contract is not locally stored specifically comprises:

when the blockchain node executes the smart contract, if the first machine code of the smart contract is not locally stored, further querying, by the blockchain node, a current global compilation status of the bytecode of the smart contract, and interpreting and executing the bytecode of the smart contract if the bytecode of the smart contract is being globally compiled; or

when the blockchain node executes the smart contract, if the first machine code of the smart contract is not locally stored, further querying, by the blockchain node, a current global compilation status of the bytecode of the smart contract, and if global compilation has not been started, starting global compilation, and interpreting and executing the bytecode of the smart contract.

10. The method according to any one of the preceding claims, further comprising: executing, by the blockchain node when executing the smart contract, the second machine code if the second machine code of the smart contract is locally stored.

11. The method according to any one of the preceding claims, further comprising: interpreting and executing, by the blockchain node when executing the smart contract, the bytecode of the smart contract if the second machine code of the smart contract is not locally stored.

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

a processor, and

a memory, wherein the memory stores a program, and when the processor executes the program, the method according to any of the preceding claims is performed.