CLAIMS

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

starting, by a blockchain node of a blockchain network after a smart contract has been deployed and consensus has been reached on the smart contract in the blockchain network, to compile a bytecode of the smart contract into a first machine code by using ahead-of-time, AoT, compilation and storing the first machine code; and

performing, by the blockchain node in execution of the smart contract and in response to the first machine code of the smart contract not being locally stored in the blockchain node, just-in-time, JIT, compilation on the bytecode of the smart contract to obtain a second machine code and storing the second machine code, and executing the second machine code the performing the JIT compilation on the bytecode of the smart contract comprising:

querying, by the blockchain node in executing the smart contract, a current AoT compilation status of the bytecode of the smart contract in response to the first machine code of the smart contract not being locally stored in the blockchain node; and

in response to the bytecode being in AoT compilation, performing JIT compilation on the bytecode of the smart contract; and

in response to the bytecode not being in AoT compilation, starting AoT compilation and performing JIT compilation on the bytecode of the smart contract,

the performing the JIT compilation on the bytecode of the smart contract further comprising:

in a process of at least one of interpreting or executing the bytecode of the smart contract, identifying a hotspot second machine code, and storing the hotspot second machine code for subsequent invoking; or

performing optimizing compilation to obtain the second machine code and storing the second machine code in the process of at least one of interpreting or executing the bytecode of the smart contract.

2. The method according to claim 1, wherein the starting, by the blockchain node after completing deployment and consensus of the smart contract, to compile the bytecode of the smart contract into the first machine code by using AoT compilation includes:

starting, by the blockchain node after the smart contract has been deployed, AoT compilation on the bytecode of the smart contract.

3. The method according to claim 1, wherein the performing, by the blockchain node, JIT compilation on the bytecode of the smart contract includes:

performing, by the blockchain node, JIT compilation on a bytecode corresponding to an invoked function in the smart contract.

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

in execution of the smart contract, executing, by the blockchain node, the first machine code of the smart contract in response to the first machine code being locally stored in the blockchain node.

5. The method according to claim 4, wherein the executing, by the blockchain node, the first machine code includes:

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

6. The method according to any one of claims 1, 4, or 5, wherein the storing is caching.

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

receiving, by the blockchain node, a request for deploying the smart contract in the blockchain network, the request including the bytecode of the smart contract.

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

a processor; and

a memory storing a program, wherein when the processor executes the program, the method according to any one of claims 1 to 7 is performed.