

CLAIMS

1. A computer-implemented method for processing blockchain transactions by a blockchain node of a blockchain network, the method comprising:

receiving a plurality of blockchain transactions to be executed by the blockchain node, wherein the blockchain node is associated with a storage system that stores blockchain data in a blockchain;

determining whether the plurality of blockchain transactions are CPU-bound or I/O-bound by the blockchain node, wherein I/O-bound blockchain transactions of the plurality of blockchain transactions comprise storage I/O-bound blockchain transactions having one or more I/O operations that access the storage system;

allocating, based on whether the plurality of blockchain transactions are CPU-bound or I/O-bound, one or more threads of a first thread pool of the blockchain node and one or more coroutines of the blockchain node for processing the plurality of blockchain transactions, wherein the blockchain node comprises a second thread pool dedicated to the storage system, the second thread pool being different from the first thread pool;

executing the plurality of blockchain transactions by the blockchain node using the allocated one or more threads and one or more coroutines, wherein executing the plurality of blockchain transactions comprises executing the one or more I/O operations asynchronously using the one or more coroutines, and wherein executing the one or more I/O operations comprises accessing blockchain data stored in the storage system using the second thread pool;

after executing the plurality of blockchain transactions, generating a blockchain block comprising the plurality of blockchain transactions; and

adding the blockchain block to the blockchain in the storage system associated with the blockchain node.

2. The computer-implemented method of claim 1, wherein allocating, based on whether the plurality of blockchain transactions are CPU-bound or I/O-bound, one or more threads of the first thread pool and one or more coroutines of the blockchain node for processing the plurality of blockchain transactions comprises:

selecting a preconfigured first number of threads for processing the plurality of blockchain transactions in parallel in response to determining that the plurality of blockchain transactions are I/O-bound, the first number of threads being preconfigured before receiving the plurality of blockchain transactions; or

determining a second number of threads for processing the plurality of blockchain transactions in parallel in response to determining that the plurality of blockchain transactions are CPU-bound,

wherein the first number of threads is equal to or less than the second number of threads.

3. The computer-implemented method of any preceding claim, wherein allocating, based on whether the plurality of blockchain transactions are CPU-bound or I/O-bound, one or more threads of the first thread pool and one or more coroutines of the blockchain node for processing the plurality of blockchain transactions comprises:

determining a third number of coroutines to be allocated for processing the plurality of blockchain transactions asynchronously in response to determining that the plurality of blockchain transactions are I/O-bound; or

determining a fourth number of coroutines to be allocated for processing the plurality of blockchain transactions asynchronously in response to determining that the plurality of blockchain transactions are CPU-bound,

wherein the fourth number of coroutines is less than the third number of coroutines.

4. The computer-implemented method of any preceding claim, wherein determining the plurality of blockchain transactions are CPU-bound or I/O-bound comprises determining the plurality of blockchain transactions are CPU-bound or I/O-bound based on transaction types of the plurality of blockchain transactions.

5. The computer-implemented method of claim 4, wherein the transaction types comprise one or more of an asset transfer transaction, an account inquiry transaction, a verification and commit transaction, a certificate storage transaction, a native smart contract transaction, an

encryption transaction, a TEE (Trusted Execution Environment) transaction, a WASM (WebAssembly) transaction, or an EVM (Ethereum Virtual Machine) transaction.

6. The computer-implemented method of claim 4, wherein determining the plurality of blockchain transactions are CPU-bound or I/O-bound based on the transaction types of the plurality of blockchain transactions comprises:

determining whether one or more of the plurality of blockchain transactions are CPU-bound or I/O-bound based on a statistical computational resource utilization record of previous blockchain transactions of the corresponding transaction types.

7. The computer-implemented method of claim 6, wherein determining the plurality of blockchain transactions are CPU-bound or I/O-bound based on the transaction types of the plurality of blockchain transactions comprises:

determining, by the blockchain node, the plurality of blockchain transactions are I/O-bound in response to determining that a total number of I/O-bound transactions, as determined based on the transaction types, in the plurality of blockchain transactions exceeds a first threshold.

8. The computer-implemented method of claim 6, wherein determining, by the blockchain node, the plurality of blockchain transactions are CPU-bound or I/O-bound based on the transaction types of the plurality of blockchain transactions comprises:

determining, by the blockchain node, the plurality of blockchain transactions are CPU-bound in response to determining that a total number of CPU-bound transactions, as determined based on the transaction types, in the plurality of blockchain transactions exceeds a second threshold.

9. The computer-implemented method of any preceding claim, wherein the blockchain node comprises a total number of threads, a subset of the total number of threads is dedicated to the storage system, and wherein the allocated one or more threads of the first thread pool are not included in the subset of the total number of threads.

10. The computer-implemented method of any preceding claim, further comprising:
 - obtaining utilization data of computational resources of the blockchain node for executing the plurality of blockchain transactions;
 - analyzing the utilization data; and
 - updating a criterion for determining whether a next plurality of blockchain transactions are CPU-bound or I/O-bound based on the analyzing.
11. A system for data processing, comprising:
 - one or more processors; and
 - one or more computer-readable memories coupled to the one or more processors and having instructions stored thereon that are executable by the one or more processors to perform the computer-implemented method of any one of claims 1 to 10.
12. An apparatus for processing blockchain transactions, the apparatus comprising a plurality of modules for performing the computer-implemented method of any one of claims 1 to 10.