

CLAIMS

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

—receiving (702) a number of M blockchain transactions to be executed by the blockchain network node, wherein the blockchain network node is associated with a storage system that stores blockchain data in a blockchain, wherein each of the M blockchain transactions is categorized as either a CPU-bound blockchain transaction or an I/O bound blockchain transaction;

executing (704) the M blockchain transactions by the blockchain network node, wherein executing the M blockchain transactions comprises:

i) identifying (706) a number of N blockchain transactions out of the M blockchain transactions ~~to be executed asynchronously~~, wherein the number of N blockchain transactions are selected for parallel processing, wherein N is smaller than or equal to M;

ii) distributing (708) the N blockchain transactions to be executed by a first thread pool (310) of K threads in parallel, wherein a respective number of N_K blockchain transactions out of the N blockchain transactions is distributed to a respective K^{th} thread; and

iii) executing (710), in parallel by the K threads, the N blockchain transactions by the K threads ~~in parallel~~, wherein: the K threads are defined for processing CPU-bound blockchain transactions in parallel, and

—for a number of N_1 blockchain transactions out of the N blockchain transactions distributed to a first thread in the first thread pool,

iii. a) identifying (721) a number of P_1 coroutines to be executed asynchronously concurrently based on the N_1 blockchain transactions, wherein P_1 is larger than or equal to N_1 ; wherein the number of P_1 coroutines are scheduled to concurrently process I/O bound operations of the N_1 blockchain transactions to define a two-dimensional computing architecture for executions of the blockchain transactions by the blockchain network node, wherein the two dimensions are K and P_1 , and

—iii. b) executing (731) the N_1 blockchain transactions by executing the P_1 coroutines asynchronously concurrently, wherein the execution of the N_1 blockchain transactions concurrently includes

~~_____executing of a transaction before receiving an execution result of a previous transaction, and~~

~~accessing blockchain data stored in the storage system using a second thread pool (312), wherein the second thread pool is different from the first thread pool, wherein the P₁ coroutines are configured to reuse one or more threads in the first thread pool or one or more additional threads in the second thread pool;~~

after executing the M blockchain transactions, generating (712) a blockchain block comprising the M blockchain transactions; and

~~_____adding (714) the blockchain block to the blockchain in the storage system associated with the blockchain network node.~~

2. The computer-implemented method of claim 1, wherein ~~executing the P₁ coroutines asynchronously comprises accessing blockchain data stored in the storage system using a second thread pool, wherein the second thread pool is different from the first thread pool~~the second thread pool (312) is dedicated to the storage system.

3. The computer-implemented method of claim 2, ~~wherein the second thread pool is dedicated to the storage system.~~

4. ~~The computer-implemented method of claim 2~~, wherein accessing blockchain data stored

~~_____~~ in the storage system comprises one or more of:

~~_____~~ reading data stored in the blockchain stored in the storage system; or

~~_____~~ writing new data into the blockchain, and

wherein the blockchain data comprises one or more of block data, state data, index data, or stream data.

~~54.~~ The computer-implemented method of any preceding claim, wherein one or more of the M blockchain transactions are executed synchronously with the N blockchain transactions.

~~65.~~ The computer-implemented method of any preceding claim, wherein identifying a number of P_1 coroutines to be executed ~~asynchronouslyconcurrently~~ based on the N_1 blockchain transactions comprises identifying one or more coroutines to be executed ~~asynchronouslyconcurrently~~ for each of the N_1 blockchain transactions.—

~~76.~~ The computer-implemented method of any preceding claim, wherein the N_1 blockchain transactions comprise a series of operations, wherein the series of operations comprise a first operation and a second ~~operationsoperation~~ subsequent to the first ~~operationsoperation~~, wherein the first operation comprises an input/output (I/O) operation accessing the storage system.

~~8.—The computer implemented method of claim 7.—The computer-implemented method of claim 6,~~ wherein the I/O operation comprises at least one of ~~an-asynchronousa concurrent~~ GET request or ~~an-asynchronousa concurrent~~ PUT request.

~~98.~~ The computer-implemented method of claim ~~76~~, wherein the P_1 coroutines ~~comprisescomprise~~ a first coroutine and a second coroutine, and wherein executing the N_1 blockchain transactions by executing the P_1 coroutines ~~asynchronouslyconcurrently~~ comprises:

— executing the first operation using the first coroutine; and
executing the second operation ~~asynchronouslyconcurrently~~ with the first operation, wherein executing the second operation ~~asynchronouslyconcurrently~~ with the first operation comprises executing the second operation using the second coroutine before receiving an execution result of the first operation.

~~109.~~ The computer-implemented method of claim ~~98~~, wherein executing the second operation using a second coroutine before receiving an execution result of the first operation comprises:

saving a first operation context of the first operation, wherein the first operation context comprises at least one of register data, stack pointer, or program counter of the first operation; and

performing a context switch from a first operation context of the first operation to a second operation context of the second operation.

~~11~~10. The computer-implemented method of claim 98, wherein the first coroutine comprises a callback identifier corresponding to a callback function of the first operation, and the method further comprises:

receiving a value returned by the callback function of the first operation;
saving a second operation context of the second operation;
restoring the first operation context of first operation; and
resuming execution of the first operation based on the value returned by the callback function of the first operation.

11. The computer implemented method of claim 8, wherein at least two coroutines are generated for two operations of a single blockchain transaction, wherein the two operations do not depend on each others execution.

12. 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 11.

13. 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 11.