

METHOD AND SYSTEM FOR DETERMINING BLOCKCHAIN-BASED GRIDLOCK RESOLUTION

TECHNICAL FIELD

[1] The disclosure relates generally to systems and methods for blockchain-based gridlock resolution.

BACKGROUND

[2] Historically, inter-bank payments were settled via netting systems (e.g., by settling the nettable payments on a net basis), but as the volume and the value of transactions kept increasing, central banks became wary of the risks involved in deferred net settlement systems. Currently, central banks favor real-time gross settlement (RTGS) systems. In RTGS, payment instructions (e.g., payment transactions) are settled individually and immediately at their full amount. However, the benefit of immediate finality often incurs high liquidity costs on the banks. The liquidity demands in RTGS systems are enormous; in fact, the daily transfer volume in typical inter-bank payment systems could be as large as a substantial fraction of a country's annual GDP. Oftentimes, participating banks within the same network are not able to settle their payments individually due to insufficient liquidity, bringing the system to a halt known as gridlock. To resolve gridlocks, central bank often combines RTGS with liquidity saving mechanisms (LSM), of which intra-network gridlock resolution is the most effective one.

[3] However, resolving cross-network (or inter-network) gridlocks poses challenges as financial institutions are building different blockchain-based infrastructures to offer their services. Different blockchain networks may be built with different blockchain technologies (e.g., such as different encryption schemes), and due to privacy-preserving concerns, transaction information (e.g., payments, balance) of one blockchain network may be invisible to another blockchain network. In these scenarios, how to resolve cross-network gridlocks for blockchain-based payment systems has become an intriguing challenge.

SUMMARY

[4] Various embodiments of the present specification may include systems, methods, and non-transitory computer readable media for blockchain-based gridlock resolution.

[5] According to one aspect, the method for blockchain-based gridlock resolution may comprise, for each blockchain of a plurality of blockchains: obtaining, by a computing system associated with an entity from the blockchain, one or more first blockchain transactions respectively corresponding to one or more incoming value transfers associated with the entity and one or more second blockchain transactions respectively corresponding to one or more outgoing value transfers associated with the entity; conducting, by the computing system associated with the entity, one or more iterations of a transaction-selection process, wherein each of the one or more iterations of the transaction-selection process comprises: determining a subset of the one or more first blockchain transactions; selecting a subset of the one or more second blockchain transactions based on the determined subset of the one or more first blockchain transactions and a balance of a blockchain account of the entity on the blockchain; determining a delta value associated with the blockchain based on the subset of the one or more first blockchain transactions, the subset of the one or more second blockchain transactions, and the balance of the blockchain account on the blockchain, wherein the delta value corresponds to a value transfer between the blockchain account on the blockchain and one or more different blockchain accounts of the entity on one or more different blockchains of the plurality of blockchains; and adding to the blockchain a third blockchain transaction comprising one or more identifiers corresponding to the selected subset of the one or more second blockchain transactions and an encrypted version of the delta value; and terminating the one or more iterations of the transaction-selection process in response to obtaining a fourth blockchain transaction from the blockchain, the fourth blockchain transaction indicating convergence of transaction selection on each of the plurality of blockchains.

[6] In some embodiments, the method may further comprise, prior to conducting the one or more iterations of the transaction-selection process: obtaining, by the computing system, a blockchain transaction in each of the blockchains indicating a beginning of the transaction-selection process.

[7] In some embodiments, the adding to the blockchain a third blockchain transaction comprises: sending the third blockchain transaction to one or more blockchain nodes associated with blockchain to add to the blockchain, wherein the third blockchain transaction invokes a blockchain contract on the blockchain, and wherein the blockchain contract is executable to

aggregate the one or more identifiers in the third blockchain transaction with a plurality of other identifiers in a plurality of other blockchain transactions.

[8] In some embodiments, the determining a subset of the one or more first blockchain transactions comprises: for a first iteration among the one or more iterations of the transaction-selection process: determining the subset of the one or more first blockchain transactions as including each of the one or more first blockchain transactions; and for each of one or more iterations, other than the first iteration, among the one or more iterations of the transaction-selection process: obtaining, by the computing system from the blockchain, one or more blockchain transactions generated by the blockchain contract, the one or more blockchain transactions comprising a plurality of identifiers; and determining the subset of the one or more first blockchain transactions based on the plurality of identifiers.

[9] In some embodiments, the selecting a subset of the one or more second blockchain transactions comprises: determining a first total amount by aggregating a plurality of values associated with a global set of first blockchain transactions, wherein the global set of first blockchain transactions comprises the one or more first blockchain transactions for each of the plurality of blockchains; determining a second total amount by aggregating a global set of balances, wherein the global set of balances comprises the balance of the blockchain account of the entity on each of the plurality of blockchains; and selecting one or more second blockchain transactions from a global set of second blockchain transactions such that a sum of one or more values associated with the selected one or more second blockchain transactions is no greater than a sum of the first total amount and the second total amount, wherein the global set of second blockchain transactions comprises the obtained one or more second blockchain transactions for each of the plurality of blockchains.

[10] In some embodiments, the global set of second blockchain transactions are respectively associated with priority rankings; and the selecting the one or more second blockchain transactions from the global set of second blockchain transactions comprises selecting the one or more second blockchain transactions based on the priority rankings.

[11] In some embodiments, the priority rankings of the global set of second blockchain transactions are determined chronologically.

[12] In some embodiments, the determining a delta value associated with the blockchain comprises: obtaining a first total amount associated with the subset of the one or more first blockchain transactions and a second total amount associated with the subset of the one or more second blockchain transactions; determining the delta value by subtracting a sum of the first total amount and the balance of the blockchain account from the second total amount.

[13] In some embodiments, the encrypted version of the delta value is generated based on a homomorphic commitment scheme or a homomorphic encryption scheme.

[14] In some embodiments, each blockchain transaction of the one or more first blockchain transactions and the one or more second blockchain transactions comprises: an index corresponding to the blockchain transaction, a homomorphically encrypted version of an amount of corresponding value transfer, and an identifier associated with a recipient of the corresponding value transfer, wherein the homomorphically encrypted version of the value transfer amount based on a homomorphic commitment scheme or a homomorphic encryption scheme.

[15] In some embodiments, the third blockchain transaction further comprises a zero-knowledge range proof that a first sum is not less than a second sum, the first sum comprising the balance of the blockchain account on the blockchain and a first total amount associated with the subset of the one or more first blockchain transactions, the second sum comprising a second total amount associated with the subset of the one or more second blockchain transactions and the delta value.

[16] In some embodiments, the method may further comprise, for each blockchain of the plurality of blockchains: obtaining, from the blockchain, a proof that a sum of a global set of delta values associated with the entity equals zero, wherein the global set of delta values comprises the delta value associated with each of the plurality of blockchains.

[17] According to yet another aspect, a non-transitory computer-readable storage medium is configured with instructions executable by one or more processors to cause the one or more processors to perform the method of any of the preceding embodiments.

[18] According to still another aspect, an apparatus for blockchain-based gridlock resolution comprises a plurality of modules for performing the method of any of the preceding embodiments.

[19] According to another aspect, a system for blockchain-based gridlock resolution may comprise a plurality of sensors and a computer system that comprises a first computing device and a second computing device, the computer system comprising a processor and a non-transitory computer-readable storage medium storing instructions executable by the processor to cause the system to perform operations comprising, for each blockchain of a plurality of blockchains: obtaining, by a computing system associated with an entity from the blockchain, one or more first blockchain transactions respectively corresponding to one or more incoming value transfers associated with the entity and one or more second blockchain transactions respectively corresponding to one or more outgoing value transfers associated with the entity; conducting, by the computing system associated with the entity, one or more iterations of a transaction-selection process, wherein each of the one or more iterations of the transaction-selection process comprises: determining a subset of the one or more first blockchain transactions; selecting a subset of the one or more second blockchain transactions based on the determined subset of the one or more first blockchain transactions and a balance of a blockchain account of the entity on the blockchain; determining a delta value associated with the blockchain based on the subset of the one or more first blockchain transactions, the subset of the one or more second blockchain transactions, and the balance of the blockchain account on the blockchain, wherein the delta value corresponds to a value transfer between the blockchain account on the blockchain and one or more different blockchain accounts of the entity on one or more different blockchains of the plurality of blockchains; and adding to the blockchain a third blockchain transaction comprising one or more identifiers corresponding to the selected subset of the one or more second blockchain transactions and an encrypted version of the delta value; and terminating the one or more iterations of the transaction-selection process in response to obtaining a fourth blockchain transaction from the blockchain, the fourth blockchain transaction indicating convergence of transaction selection on each of the plurality of blockchains.

[20] According to yet another aspect, a non-transitory computer-readable storage medium for blockchain-based gridlock resolution may be configured with instructions executable by one or

more processors to cause the one or more processors to perform operations comprising, for each blockchain of a plurality of blockchains: obtaining, by a computing system associated with an entity from the blockchain, one or more first blockchain transactions respectively corresponding to one or more incoming value transfers associated with the entity and one or more second blockchain transactions respectively corresponding to one or more outgoing value transfers associated with the entity; conducting, by the computing system associated with the entity, one or more iterations of a transaction-selection process, wherein each of the one or more iterations of the transaction-selection process comprises: determining a subset of the one or more first blockchain transactions; selecting a subset of the one or more second blockchain transactions based on the determined subset of the one or more first blockchain transactions and a balance of a blockchain account of the entity on the blockchain; determining a delta value associated with the blockchain based on the subset of the one or more first blockchain transactions, the subset of the one or more second blockchain transactions, and the balance of the blockchain account on the blockchain, wherein the delta value corresponds to a value transfer between the blockchain account on the blockchain and one or more different blockchain accounts of the entity on one or more different blockchains of the plurality of blockchains; and adding to the blockchain a third blockchain transaction comprising one or more identifiers corresponding to the selected subset of the one or more second blockchain transactions and an encrypted version of the delta value; and terminating the one or more iterations of the transaction-selection process in response to obtaining a fourth blockchain transaction from the blockchain, the fourth blockchain transaction indicating convergence of transaction selection on each of the plurality of blockchains.

[21] According to yet other embodiments, an apparatus for executing a blockchain-based gridlock resolution may comprise an obtaining module, a transaction selection module, and a termination module. In some embodiments, the obtaining module may, for each blockchain of a plurality of blockchains obtain one or more first blockchain transactions respectively corresponding to one or more incoming value transfers associated with the entity and one or more second blockchain transactions respectively corresponding to one or more outgoing value transfers associated with the entity. In some embodiments, the transaction selection module may conduct one or more iterations of a transaction-selection process, wherein each of the one or more iterations of the transaction-selection process comprises: determining a subset of the one or more first blockchain transactions; selecting a subset of the one or more second blockchain

transactions based on the determined subset of the one or more first blockchain transactions and a balance of a blockchain account of the entity on the blockchain; determining a delta value associated with the blockchain based on the subset of the one or more first blockchain transactions, the subset of the one or more second blockchain transactions, and the balance of the blockchain account on the blockchain, wherein the delta value corresponds to a value transfer between the blockchain account on the blockchain and one or more different blockchain accounts of the entity on one or more different blockchains of the plurality of blockchains; and adding to the blockchain a third blockchain transaction comprising one or more identifiers corresponding to the selected subset of the one or more second blockchain transactions and an encrypted version of the delta value. In some embodiments, the termination module may terminate the one or more iterations of the transaction-selection process in response to obtaining a fourth blockchain transaction from the blockchain, the fourth blockchain transaction indicating convergence of transaction selection on each of the plurality of blockchains.

[22] Embodiments disclosed in the specification have one or more technical effects. In some embodiments, a blockchain-based method collects information of incoming and outgoing transactions for a plurality of entities, identifies a subset of the transactions that can be settled, and settles the identified transactions via a decentralized multi-step mechanism, in which all of the entities participate. The mechanism is enabled by automated operation of blockchain contracts. This provides effective, fair, reliable, and transparent resolution of the transactions and any gridlock that may occur. In other embodiments, the method arranges records of multiple entities' assets and liabilities across multiple blockchain networks and facilitates dynamic movement of each entity's assets across blockchain networks to satisfy the entity's transaction needs. This allows trusted exchange of information among different blockchain networks with different characteristics such as consensus protocols, access rules, and participating nodes. Such trusted exchange of information among blockchain networks improves effective use of each entity's liquidity and the chances for resolution of gridlock. In some embodiments, the transaction information within a blockchain network is kept only visible to related parties, and confidential information such as payment amounts are protected by commitments and proved by zero-knowledge range proofs. This protects confidential information of the participating entities. In some embodiments, the transactions to be settled are proposed based on a first-in-first-out protocol at a global scale across multiple participants and multiple networks and all participants

participate in each step of the gridlock resolution process. This ensures the fairness of the final resolution for all participants.

[23] These and other features of the systems, methods, and non-transitory computer readable media disclosed herein, as well as the methods of operation and functions of the related elements of structure and the combination of parts and economies of manufacture, will become more apparent upon consideration of the following description and the appended claims with reference to the accompanying drawings, all of which form a part of this specification, wherein like reference numerals designate corresponding parts in the various figures. It is to be expressly understood, however, that the drawings are for purposes of illustration and description only and are not intended as a definition of the limits of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

[24] **FIG. 1** illustrates a network environment associated with a blockchain in accordance with some embodiments.

[25] **FIG. 2** illustrates a framework for implementing blockchain transactions in accordance with some embodiments.

[26] **FIG. 3** illustrates an example network environment for implementing blockchain-base gridlock resolution in accordance with some embodiments.

[27] **FIG. 4** illustrates an example diagram of a computing system for implementing blockchain-base gridlock resolution in accordance with some embodiments.

[28] **FIG. 5** illustrates an example blockchain contract for implementing blockchain-base gridlock resolution in accordance with some embodiments.

[29] **FIG. 6** illustrates an example diagram of a timing service for implementing blockchain-base gridlock resolution in accordance with some embodiments.

[30] **FIG. 7** illustrates an example workflow for implementing blockchain-base gridlock resolution in accordance with some embodiments.

[31] **FIG. 8** illustrates an example application of blockchain-base gridlock resolution in accordance with some embodiments.

[32] **FIG. 9** illustrates an example method for implementing blockchain-base gridlock resolution in accordance with some embodiments.

[33] **FIG. 10** illustrates an example block diagram of a computing system for implementing blockchain-base gridlock resolution in accordance with some embodiments.

[34] **FIG. 11** illustrates an example block diagram of a computer system in which any of the embodiments described herein may be implemented.

DETAILED DESCRIPTION

[35] Embodiments described herein provide methods, systems, and apparatus associated with a framework for blockchain-based gridlock resolution. Gridlocks may occur when entities exchanging assets are unable to settle their outgoing transactions or liabilities individually due to insufficient liquidity. The platform integrates various components, such as blockchain networks, off-chain communication channels, cloud applications, client applications, application programming interfaces, and other suitable components to enable various functionalities related to resolving cross-network gridlocks. Various parties, such as asset owners or service providers (e.g., banks), regulatory agencies, information service providers, resolution coordinators, may participate in the gridlock resolution framework. The parties may participate in the gridlock resolution process by interacting with one or more blockchain networks for recording transactions that are to be settled. A party may interact with a blockchain network by controlling a node associated with the blockchain network, participating in a consensus protocol associated with the blockchain network via the node, and performing operations associated with gridlock resolution using the node. A party may alternatively interact with the blockchain network through one or more blockchain nodes associated with one or more other entities. A party may also interact with the blockchain network using one or more interfaces provided by a platform offering various blockchain-related services.

[36] To illustrate the application scenario of various embodiments disclosed herein, here is a simple example of gridlock resolution in a single payment network. For example, Bank 1, Bank

2, and Bank 3 in a same network (e.g., a blockchain network) are involved in a three-way asset-transfer scenario: Bank 1 needs to transfer \$120K to Bank 2, Bank 2 needs to transfer \$150K to Bank 3, and Bank 3 needs to transfer \$80K to Bank 1. Assuming Bank 1, Bank 2, and Bank 3 each has a current balance of \$50K, none of them can directly settle its outgoing transaction due to insufficient current balance. However, the banks may settle the payments on a net basis. The assets transfer transactions may be aggregated by a central computing system within the network to determine whether a gridlock resolution is available. For example, Bank 1 may take into account the incoming transaction \$80K from Bank 3 when settling its outgoing transaction \$120K to Bank 2; Bank 2 may take into account the incoming transaction \$120K from Bank 1 when settling its outgoing transaction \$150K to Bank 3; and Bank 3 may take into account the incoming transaction \$150K from Bank 2 when settling its outgoing transaction \$80K to Bank 1. As a result, Bank 1 only needs to transfer \$40K to Bank 3, and Bank 2 only needs to transfer \$30K to Bank 3. After applying the gridlock resolution, Bank 1's new balance will be \$10K, Bank 2's new balance will be \$20K, and Bank 3's new balance will be \$120K.

[37] Gridlock resolution based on direct-netting may be impractical when transactions are distributed across a plurality of blockchain networks. For example, the transactions in one network may be invisible to another network and/or encrypted in a different way than that in other networks, and there may not be a central computing system that is able to aggregate all asset-transfer transactions across the networks. According to some embodiments provided in this specification, a resolution coordinator (e.g., a computing system associated with a timing service provider) may manage the workflow of a blockchain-based gridlock resolution, such as starting the resolution process, timing out/terminating an iteration of the workflow, and terminating the entire resolution. The resolution may comprise one or more iterations of transaction-selection processes on each of a plurality of blockchain networks. During each iteration, each participant (e.g., a computing system associated with a bank or an asset exchange institution) may propose a list of outgoing transactions to settle during the current iteration by considering various factors, such as priorities of its outgoing transactions, other participants' proposals, current balances of its blockchain accounts on the plurality of blockchain networks, other suitable factors, or any combination thereof. The resolution coordinator may monitor the proposals submitted by each participant across the plurality of blockchain networks. When the proposals are converged at a global scale (e.g., implying that all participants have agreed the proposals), the resolution

coordinator may determine a gridlock solution has been found (e.g., a list of outgoing transactions may be settled among the participants across the plurality of blockchain networks). Subsequently, an information service provider (such as a global notary service or an oracle) offering cross-network interoperability may sign on the gridlock solution and notify the assets exchange institutions to settle the corresponding transactions.

[38] FIG. 1 illustrates a network environment associated with a blockchain in accordance with some embodiments. As shown, in the environment 100, a client 111 may couple to a server end 118, and the server end 118 and a Node B may couple to a blockchain system 112 through various communication networks. Similarly, the server end 118 may optionally couple to additional blockchain systems similar to the blockchain system 112 such as blockchain system 113, blockchain system 114, etc. Each blockchain system may maintain one or more blockchains.

[39] In some embodiments, the client 111 may comprise one or more servers (e.g., Node C) and one or more other computing devices (e.g., Node A1, Node A2, Node A3). Node A1, Node A2, and Node A3 may couple to Node C. In some embodiments, Node C may be implemented by an entity (e.g., website, mobile phone Application, organization, company, enterprise), which has various local accounts (e.g., local accounts accessed from Node A1, Node A2, Node A3). For example, a mobile phone application may have millions of end-users accessing the application's server from respective user accounts. The application's server may correspondingly store millions of user accounts. The components of the client 111 and their arrangement may have many other configurations.

[40] In some embodiments, Node B may include a lightweight node. A lightweight node may not download the complete blockchain but may instead just download the block headers to validate the authenticity of the blockchain transactions. Lightweight nodes may be served by and effectively dependent on full nodes (e.g., blockchain nodes in the blockchain system 112) to access more functions of the blockchain. The lightweight nodes may be implemented in electronic devices such as laptops, mobile phones, and the like by installing an appropriate software.

[41] In some embodiments, there may be many more clients coupled to the server end 118 similar to client 111. The server end 118 may provide Blockchain-as-a-Service (BaaS) and be referred to as a BaaS cloud. In one embodiment, BaaS is a cloud service model in which clients or developers outsource behind-the-scenes aspects of a web or mobile application. BaaS may provide pre-written software for activities that take place on blockchains, such as user authentication, database management, and remote updating. The BaaS cloud may be implemented in a server, server cluster, or other devices. In one embodiment, the BaaS cloud provides an enterprise-level platform service based on blockchain technologies. This service may help clients to build a secure and stable blockchain environment as well as manage the deployment, operation, maintenance, and development of blockchain easily. Based on the abundant security strategies and multi-tenant isolation of cloud, the BaaS cloud can provide advanced security protection using chip encryption technologies. Based on highly reliable data storage, this service may provide end-to-end and highly available services that can scale up quickly without interruption. The BaaS cloud can provide native support for standard blockchain applications and data.

[42] In some embodiments, the blockchain system 112 may comprise a plurality of blockchain nodes (e.g., Blockchain Node 1, Blockchain Node 2, Blockchain Node 3, Blockchain Node 4, Blockchain Node i, etc.) that maintain one or more blockchains (e.g., public blockchain, private blockchain, consortium blockchain). Other blockchain systems (e.g., blockchain system 113, blockchain system 114) may comprise similar arrangements of blockchain nodes maintaining other blockchains. Each blockchain node may be found in one or more blockchain systems. The blockchain nodes of each blockchain system may maintain one or more blockchains. The blockchain nodes may include full nodes. Full nodes may download every block and blockchain transaction and check them against the blockchain's consensus rules. The blockchain nodes may form a network (e.g., peer-to-peer network) with one blockchain node communicating with another. The order and the number of the blockchain nodes as shown are merely examples for illustration. The blockchain nodes may be implemented in servers, computers, etc. For example, each blockchain node may be implemented in a server or a cluster of servers. The cluster of servers may employ load balancing. Each blockchain node may correspond to one or more physical hardware devices or virtual devices coupled together via

various types of communication methods such as TCP/IP. Depending on the classifications, the blockchain nodes may also be referred to as full nodes, Geth nodes, consensus nodes, etc.

[43] In the environment 100, each of the nodes and devices may be installed with appropriate software (e.g., application programming interface) and/or hardware (e.g., wires, wireless connections) to access other devices of the environment 100. In general, the nodes and devices may be able to communicate with one another through one or more wired or wireless networks (e.g., the Internet) through which data can be communicated. Each of the nodes and devices may include one or more processors and one or more memories coupled to the one or more processors. The memories may be non-transitory and computer-readable and configured with instructions executable by one or more processors to cause the one or more processors to perform operations described herein. The instructions may be stored in the memories or downloaded over a communications network without necessarily being stored in the memories. Although the nodes and devices are shown as separate components in this figure, it will be appreciated that these nodes and devices can be implemented as single devices or multiple devices coupled together. For example, Node B may be alternatively integrated into Blockchain Node 2.

[44] The devices such as Node A1, Node A2, Node A3, Node B, and Node C may be installed with an appropriate blockchain software to create blockchain accounts, and initiate, forward, or access blockchain transactions. The term “blockchain transaction” may refer to a unit of task executed in a blockchain system and recorded in the blockchain. For example, Node A1 may access the blockchain through communications with Node C, the server end 118, and Blockchain Node 1, and Node B may access the blockchain through communications with Blockchain Node 2. In some embodiments, Node A1 may submit a blockchain account creation request to Node C. Node C may forward the request and other similar requests to the server end 118. The server end 118 may accordingly create blockchain accounts.

[45] In some embodiments, after receiving a blockchain transaction request of an unconfirmed blockchain transaction, a recipient blockchain node may perform some preliminary verification of the blockchain transaction. For example, Blockchain Node 1 may perform the preliminary verification after receiving a blockchain transaction from Node C. Once verified, the

blockchain transaction may be stored in a database of the recipient blockchain node (e.g., Blockchain Node 1), which may also forward the blockchain transaction to one or more other blockchain nodes (e.g., Blockchain Node 3, Blockchain Node 4). As each blockchain node may comprise or couple to a memory, the database may be respectively stored in the memories of the blockchain nodes. The database may store a pool of blockchain transactions submitted by the one or more client devices. After receiving the blockchain transaction, the one or more other blockchain nodes may repeat the process done by the recipient blockchain node.

[46] Each blockchain node may select some of the blockchain transactions from the pool according to its preference and form them into a proposed new block for the blockchain. The blockchain node may perform “mining” of the proposed new block by devoting computing power to solve complex mathematical problems. If the blockchain transaction involves a blockchain contract, the blockchain nodes may execute the blockchain contract locally in respective virtual machines (VMs). The blockchain contract may comprise instructions, code, or programs that are automatically executable by a blockchain system when one or more preset triggering conditions are met. To handle the blockchain contracts, each blockchain node of the blockchain network may run a corresponding VM and executes the same instructions in the blockchain contract. A VM is a software emulation of a computer system based on computer architectures and provides functionality of a physical computer. VM in the blockchain context can be understood as a system designed to operate as a runtime environment for blockchain contracts.

[47] A certain blockchain node that successfully mines the proposed new block of blockchain transactions in accordance with consensus rules may pack the new block into its local copy of the blockchain and multicast the results to other blockchain nodes. The certain blockchain node may be a blockchain node that has first successfully completed the verification, that has obtained a verification privilege, or that has been chosen based on another consensus rule, etc. Then, the other blockchain nodes may follow the same order of execution performed by the certain blockchain node to locally execute the blockchain transactions in the new block, verify the execution results with one another (e.g., by performing hash calculations), and synchronize their copies of the blockchain with that of the certain blockchain node. By updating their local copies of the blockchain, the other blockchain nodes may similarly write such information in the

blockchain transaction into respective local memories. As such, the blockchain contract can be deployed on the blockchain. If the verification fails at some point, the blockchain transaction is rejected.

[48] The deployed blockchain contract may have an address, according to which the deployed contract can be accessed. A blockchain node may invoke the deployed blockchain contract by inputting certain parameters to the blockchain contract. In one embodiment, Node C or Node B may request to invoke the deployed blockchain contract to perform various operations. For example, data stored in the deployed blockchain contract may be retrieved. For another example, data may be added to the deployed blockchain contract. For yet another example, a financial transaction specified in the deployed blockchain contract may be executed. Notwithstanding the above, other types of blockchain systems and associated consensus rules may be applied to the disclosed blockchain system.

[49] **FIG. 2** illustrates a framework for implementing blockchain transactions in accordance with some embodiments. In some embodiments, the client 111 may transmit information (e.g., a request with relevant information for creating a blockchain account) to the server end 118 for the server end 118 to create a blockchain account. To this end, the server end 118 may generate cryptographic keys, compile the request with other account creation requests, and/or perform other operations. Then, the server end 118 may transmit a blockchain transaction (e.g., blockchain transaction A) including the compiled account creation requests to one or more of blockchain nodes for execution.

[50] In some embodiments, Node B may construct a signed blockchain transaction and transmit it to one or more blockchain nodes for execution. In one embodiment, Node B may construct a blockchain transaction B. The blockchain transaction B may comprise a blockchain contract B for deployment or invoke a deployed blockchain contract. For example, the blockchain transaction B may comprise a blockchain contract that creates a blockchain account or invokes a deployed blockchain contract A. The blockchain contract B may be programmed in source code at a user-end application 221. For example, a user or machine may program the blockchain contract B. Node B may compile the source code using a corresponding compiler, which converts the source code into bytecode. The blockchain transaction B may comprise

information such as nonce (e.g., transaction serial number), from (e.g., a blockchain address of Node B or another blockchain address), to (e.g., empty if deploying a blockchain contract), transaction fee, value (e.g., transaction amount), signature (e.g., signature of Node B), data (e.g., message to a contract account), etc. The Node B may send the blockchain transaction B to one or more blockchain nodes through a remote procedure call (RPC) interface 223 for execution. RPC is a protocol that a first program (e.g., user-end application) can use to request a service from a second program located in another computer on a network (e.g., blockchain node) without having to understand the network's details. When the first program causes a procedure to execute in a different address space, it is as if a normal (local) procedure call, without the programmer explicitly coding the details for the remote interaction.

[51] In some embodiments, on receiving the blockchain transaction (e.g., blockchain transaction A or B), the recipient blockchain may verify if the blockchain transaction is valid. For example, the signature and other formats may be verified. If the verification succeeds, the recipient blockchain node may broadcast the received blockchain transaction (e.g., blockchain transaction A or B) to the blockchain network including various other blockchain nodes. Some blockchain nodes may participate in the mining process of the blockchain transactions. The blockchain transaction may be picked by a certain blockchain node for consensus verification to pack into a new block. If the blockchain transaction involves a blockchain contract, the certain blockchain node may create a contract account for a blockchain contract in association with a contract account address. If the blockchain transaction involves invoking a deployed blockchain contract, the certain blockchain node may trigger its local VM to execute the received blockchain transaction, thereby invoking the deployed blockchain contract from its local copy of the blockchain and updating the account states in the blockchain. If the certain blockchain node succeeds in mining a new block, the certain blockchain node may broadcast the new block to other blockchain nodes. The other blockchain nodes may verify the new block as mined by the certain blockchain node. If consensus is reached, the blockchain transaction B is respectively packed to the local copies of the blockchain maintained by the blockchain nodes. The blockchain nodes may similarly trigger their local VMs to execute the blockchain transaction B, thus invoking the blockchain contract A deployed on the local copies of the blockchain and making corresponding updates.

[52] Upon receiving the new block, the other blockchain nodes may perform verifications. If a consensus is reached that the new block is valid, the new block is respectively packed to the local copies of the blockchain maintained by the blockchain nodes. The blockchain nodes may similarly trigger their local VMs (e.g., local VM 1, local VM i, local VM 2) to execute the blockchain transactions in the new block, thus invoking local copies of the blockchain (e.g., local blockchain copy 1, local blockchain copy i, local blockchain copy 2) and making corresponding updates. The hardware machine of each blockchain node may have access to one or more virtual machines, which may be a part of or couple to the corresponding blockchain node. Each time, a corresponding local VM may be triggered to execute the blockchain transaction. Likewise, all other blockchain transactions in the new block will be executed. Lightweight nodes may also synchronize to the updated blockchain.

[53] **FIG. 3** illustrates an example network environment for implementing a blockchain-based gridlock resolution in accordance with some embodiments. Each of the letters A, B, C, D, E in FIG. 3 represents a participant involved in asset or other transactions (e.g., an assets transfer institution such as a bank). In the example network environment, A participates in three blockchain networks 310, 320, and 330; B participates in two blockchain networks 310 and 320; C participates in two blockchain networks 310 and 330; D and E participate in blockchain network 320 and 330 respectively. Each of the blockchain networks 310, 320, 330 may be implemented as one or more of the blockchain systems 112-114 in FIGs. 1 and 2 or components thereof. In some embodiments, a participant may participate in a blockchain network by providing a blockchain node (e.g., in the form of a computing device, or a virtual machine) to the blockchain network. A participant may participate in multiple blockchain networks through multiple separate computing devices, or multiple virtual machines provided by one or more physical computing devices. In other embodiments, a participant may participate in a blockchain network via one or more blockchain nodes associated with one or more other entities or via one or more interfaces provided by a platform offering various blockchain based services. The platform may be implemented as the server end 118 in FIGs. 1-2 or components thereof. Computing systems associated with the participants may each be implemented as the client 111 in FIGs. 1-2 or components thereof. Within each blockchain network that a participant is engaged in, a blockchain account may be created and associated with the participant. The blockchain account may comprise an account balance indicating a current balance that the

participant possesses in the corresponding blockchain network. In some embodiments, the blockchain networks may support creation and transaction of blockchain based assets (e.g., tokens, cryptocurrencies). One or more participants of a common blockchain network may execute one or more asset transactions to each other by adding one or more blockchain transactions to a blockchain associated with the blockchain network. By way of the blockchain transactions, blockchain-based assets may be transferred among blockchain accounts of the participants. In other embodiments, the blockchain networks may be used to record off-blockchain asset ownership and transactions. One or more participants may engage in asset transactions independent of a blockchain network and record the details of the asset transactions on the blockchain associated with the blockchain network. Alternatively, one or more participants may commit to asset transactions by adding one or more blockchain transactions representing the asset transactions to the blockchain. The one or more participants may later clear the committed asset transactions via one or more off-blockchain channels. Furthermore, a participant may have access to all its blockchain accounts on all the blockchain networks that it is engaged in. That is, the participant may transfer a certain amount of balance from one blockchain account in one blockchain network to another blockchain account on another blockchain network (e.g., redistributing its cross-network assets). This may be achieved by internal record updates or transactions via off-blockchain channels. The participant can add one or more blockchain transactions to each blockchain to update the respective account balances.

[54] In some embodiments, the blockchain-base gridlock resolution may require a resolution coordinator to coordinate the operations among the multiple blockchain networks. Such coordinator may be implemented as a timing service 360 that notifies the participants (e.g., assets transfer institutions) the commencement of a gridlock resolution process, notifies each blockchain network that a certain phase of the resolution process (e.g., one of a plurality of iterations in the gridlock resolution process) is finished, monitors each blockchain network to detect convergences, performing another suitable task, or any combination thereof. In some embodiments, the timing service 360 may comprise a blockchain node associated with each of the blockchain networks (e.g., 310, 320, and 330) in order to interact with the other blockchain nodes in the each blockchain network, including the nodes associated with the participants. For example, the timing service 360 may submit a blockchain transaction to the blockchain networks 310, 320 and 330 to kick off a gridlock resolution process (e.g., each participant may start

proposing outgoing transactions to settle once it observes the blockchain transaction) or to notify the participants that a resolution process will start at a specific time. As another example, the timing service 360 may, through the blockchain nodes associated with the timing service 360, monitor the blockchain networks 310, 320 and 330 to detect blockchain transactions indicating a solution to the gridlock has been found. An embodiment of the timing service 360 is described in detail in FIG. 6.

[55] FIG. 4 illustrates an example diagram of a computing system for implementing blockchain-based gridlock resolution in accordance with some embodiments. The computing system 400 in FIG. 4 may refer to a computing device, a virtual machine, or another suitable terminal device that is associated with a participant (e.g., an assets transfer institution) involved in a gridlock resolution process. The computing system 400 may be implemented as one or more of the components in FIGs. 1-3, such as the client 111, the server end 118, the user-end application 221. The computing system 400 may enable the participant to participate in a blockchain network 440. The blockchain network 440 may be implemented as one or more of the blockchain systems 112-114 in FIGs. 1 and 2 or components thereof, or one or more of the blockchain systems 310-330 in FIG. 3. In some embodiments, a participant may be associated with a plurality of such computing systems 400 to participate in a plurality of blockchain networks, respectively.

[56] In some embodiments, the computing system 400 may include a plurality of modules providing various functionalities. The modules shown in FIG. 4 are exemplary. Depending on the implementation, the computing system 400 may have more, fewer, alternative modules. Some of the modules may be merged or split. In the embodiment shown in FIG. 4, the computing system 400 comprises a transaction collection module 421, a transaction selection module 422, a delta determination module 423, and a blockchain interaction module 425.

[57] In some embodiments, the transaction collection module 421 of the computing device 400 may collect all outgoing transactions (e.g., outgoing payments) and all incoming transactions (e.g., incoming payments) associated with the participant from the blockchain in the blockchain network 440. These transactions may be blockchain transactions retrieved from a blockchain associated with the blockchain network 440. An outgoing transaction may comprise or represent

a transfer of value from a blockchain account associated with the participant (e.g., transferor) to a blockchain account associated with another participant (e.g., transferee). The transferor and transferee in the outgoing transaction may be represented by their respective unique identifiers, such as blockchain account addresses. The outgoing transaction may also comprise the amount to be transferred out. For privacy protection purposes, the amount may be represented as a ciphertext or a commitment before sending to the blockchain network 440. In some embodiments, the amounts may be homomorphically encrypted (e.g., by applying a homomorphic encryption or a homomorphic commitment scheme).

[58] In some embodiments, the transaction selection module 422 of the computing device 400 may select a subset of the outgoing transactions collected by the transaction collection module 421 to settle. The selected outgoing transactions may be determined based on various factors, such as the current balance of the participant in the current blockchain network 440, the total amount of incoming transactions that the participant presumes it may receive (e.g., based on outgoing transactions proposed by other participants in the blockchain network 440), and/or an amount that the participant may transfer in (e.g., assets reallocation) from its other blockchain accounts in other blockchain networks.

[59] In some embodiments, if the factors indicate that the currently selected outgoing transactions may not be settled (e.g., not enough assets), the transaction selection module 422 may remove one or more outgoing transactions with a lowest priority from the subset. In some embodiments, each outgoing transaction may be associated with a certain priority ranking determined chronologically (e.g., an outgoing transaction with an older timestamp has a higher priority). In some embodiments, the outgoing transaction's priority ranking may be determined by other factors, such as the recipient's priority (e.g., if the outgoing payment is being paid to a preferred client/receiver, it may have a higher priority), a deadline of the outgoing transaction (e.g., different recipients may set different payment deadlines), another suitable factors, or any combination thereof.

[60] In some embodiments, the delta determination module 423 of the computing device 400 may determine a delta value corresponding to the participant's blockchain account in the blockchain network 440, where the delta value may be a variable indicating an amount to be

transferred into or out of the participant's blockchain account in the blockchain network 440. For example, the delta value may be determined based on the participant's current balance in the blockchain account in the blockchain network 440, all the incoming transactions in the blockchain network 440 that are to be received by the participant, and the outgoing transactions selected by the transaction selection module 422 (e.g., the outgoing transactions that are proposed to settle). If a total amount of the outgoing transactions is greater than a sum of the current balance and a total amount of the incoming transactions, the participant may need to transfer in some assets from its other blockchain accounts, which may be represented by a positive delta value. If the total amount of the outgoing transactions is smaller than the sum of the current balance and the total amount of the incoming transactions, the participant may be able to transfer out a certain amount to help out its other blockchain accounts to settle outgoing transactions where the certain amount may be represented by a negative delta value).

[61] If the participant has a plurality of computing systems 400, each of the computing systems 400 may determine (e.g., through its delta determination module 423) a delta value such that the total sum of the plurality of delta values is 0. That is, during this internal redistribution of assets (e.g., assets redistribution among the participant's plurality of blockchain accounts), no liquidity may be created or diminished. In some embodiments, in order to guarantee that the sum of the plurality of delta values is 0, the participant's plurality of computing systems 400 may communicate and determine their delta values collectively, or one of the computing systems 400 may aggregate information from others and determine the plurality of delta values for the computing systems 400.

[62] In some embodiments, the blockchain interaction module 425 of the computing device 400 may submit blockchain transactions (including blockchain contracts) to the blockchain network 440, monitor the blockchain associated with the blockchain network 440. For example, when the sum of all the delta values (e.g., respectively corresponding to the computing systems 400 of the participant) is 0, the blockchain interaction module 425 may submit a blockchain transaction comprising the outgoing transactions selected by the transactions selection module 422 and the delta value (e.g., in an encrypted format) determined by delta determination module 423 to the blockchain network 440. The blockchain transaction may represent a proposal from the computing system 400 that the included outgoing transactions of the participant that may be

settled. In some embodiments, the blockchain transaction may also comprise a range proof that a post-balance of the participant's current blockchain account is non-negative after the proposed outgoing transactions are settled and the balance redistribution (e.g., with an amount to be transferred in or out represented by the delta value) is performed.

[63] FIG. 5 illustrates an example blockchain contract for implementing blockchain-base gridlock resolution in accordance with some embodiments. The blockchain contract 520 may be created in a blockchain associated with a blockchain network 440 that comprises a plurality of blockchain nodes. The blockchain nodes may comprise computing systems such as 420A, 420B, 420C that are associated with a plurality of participants. As illustrated by the embodiments of FIG.4, each of the computing systems may submit a blockchain transaction comprising selected outgoing transactions that are proposed to be settled by each participant. These blockchain transactions (e.g., proposals) may be aggregated by the blockchain contract 520 (e.g., a smart contract). For example, the blockchain transactions may specify the blockchain contract 520's address and trigger certain methods of the blockchain contract 520 when submitting the corresponding proposals.

[64] In some embodiments, the blockchain contract 520 may comprise a plurality of modules, such as an aggregation module 522, a verification module 523, and a reporting module 524. Each module may comprise computer-readable code representing programs, algorithms, software methods, other suitable functional components, or any combination thereof. Depending on the implementation, the blockchain contract 520 may comprise fewer, more, or alternative modules.

[65] The aggregation module 522 may be configured to receive and aggregate the proposals submitted from the computing systems such as 420A, 420B, and 420C. In some embodiments, the aggregation module 522 may be implemented as a method in the blockchain contract 520 that extracts the proposed outgoing transactions from the submitted blockchain transactions.

[66] The verification module 523 may be configured to verify that a post-balance of a participant is non-negative after the participant's proposed outgoing transactions are settled and the balance redistribution according to the delta value is performed. In some embodiments, the transaction amounts in the blockchain transactions and the delta value in the proposal may be homomorphically encrypted (e.g., in the form of a ciphertext or a commitment), and the

calculation of the post-balance may be adjusted according to the used homomorphic encryption or commitment scheme. In some embodiments, the submitted blockchain transaction may already comprise a range proof provided by the corresponding participant. In this case, the verification module 523 may need to verify the provided proof. In some embodiments, if the verification of a proposal fails, the blockchain contract may notify the corresponding blockchain node/computing system (e.g., by submitting another blockchain transaction) to re-submit another proposal, or exclude the blockchain node (e.g., the corresponding participant) from participating in the current gridlock resolution process.

[67] The reporting module 524 may be configured to submit to the blockchain network 440 a blockchain transaction comprising the verified outgoing transactions that the plurality of participants in the blockchain network 440 propose to settle. This blockchain transaction may be monitored by the computing systems such as 420A, 420B, 420C as an indication that the next iteration may start. Since one participant's outgoing transactions may be another participant's incoming transactions, for a specific participant, these verified outgoing transactions may reveal all the incoming transactions that it may receive from other participants in the blockchain network 440. As a result, the blockchain transaction submitted by the reporting module 524 may be the basis for each participant to identify all incoming transactions that it may receive and adjust its proposal during the next iteration.

[68] **FIG. 6** illustrates an example diagram of a timing service for implementing blockchain-base gridlock resolution in accordance with some embodiments. In some embodiments, the timing service 360 may be configured to coordinate the gridlock resolution process among a plurality of blockchain networks. For example, the timing service 360 may notify each of the plurality of blockchain networks the commencement of the gridlock resolution process, notify each blockchain network that a certain iteration of the process (e.g., one of a plurality of iterations in the resolution process) is finished, monitors each blockchain network to detect convergences (e.g., a local convergence on a blockchain network, and a global convergence across all blockchain networks), performing another suitable task, or any combination thereof. In some embodiments, the timing service 360 may comprise a plurality of blockchain nodes for the blockchain networks. The timing service 360 may perform the aforementioned operations by submitting one or more blockchain transactions to each of the blockchain networks.

[69] The exemplary timing service 360 illustrated in FIG. 6 comprises a plurality of modules, including a triggering module 362, a timeout module 364, and a monitoring module 366. Depending on the implementation, the timing service 360 may have more, fewer, alternative modules. Some of the modules may be merged or split.

[70] The triggering module 362 may be configured to periodically (e.g., every hour, every 6 hours, every day, every 10 minutes) submit blockchain transactions to the blockchain networks to kick off an iteration (e.g., a gridlock resolution process may comprise a plurality of iterations) by notifying the participants therein to make proposals (e.g., notifying each participant to submit a blockchain transaction comprising a list of outgoing transactions that the participant proposes to settle). Here, the “notifying” may be understood as submitting blockchain transactions by the timing service 360 to the blockchain networks, such that participants that monitor the blockchains associated with the blockchain networks obtain the blockchain transactions.

[71] The timeout module 364 may be configured to periodically submit timeout blockchain transactions to the blockchain networks to terminate a current iteration during which the participants are allowed to submit their proposals. In some embodiments, if one or more participants fail to submit proposals before timeout, they may be excluded from participating in the current gridlock resolution process. In other embodiments, these participants may be allowed to participate in the next iteration of the same gridlock resolution process.

[72] The monitoring module 366 may be configured to monitor the gridlock resolution process, and determine whether a valid solution has been found, or whether no valid solution was found within a predetermined time window. For example, a valid solution to a gridlock may be found when a global convergence occurs. The global convergence occurs when each of the blockchain networks reaches a local convergence. A local convergence occurs when all proposals in the corresponding blockchain network remain the same for two or more iterations.. In order to observe the convergences, the monitoring module 366 in some embodiments may monitor the blockchain transactions submitted by the blockchain contracts 520 shown in FIG. 5 (through the reporting modules 524) during a current iteration and from a last iteration, and compare the outgoing transactions included therein. Once the global convergence is observed, the timing service 360 may submit a blockchain transaction to each of the blockchain networks

to announce the termination of the gridlock resolution process. In some embodiments, each blockchain contract may first determine whether a local convergence is reached, and if so, submit a corresponding blockchain transaction to notify the monitoring module 366 (e.g., when the corresponding blockchain transaction is observed by the monitoring module 366). The global convergence may be reached when the monitoring module 366 observes the local convergences on all of the plurality of blockchain networks simultaneously.

[73] **FIG. 7** illustrates an example workflow for implementing blockchain-base gridlock resolution in accordance with some embodiments. The example workflow in FIG. 7 involves a timing service 710, a plurality of blockchain networks (e.g., 720A, 720B), a plurality of computing systems (e.g., 730A, 730B) associated with a plurality of participants (e.g., the computing systems may refer to blockchain nodes that enable the participants to interact with the corresponding blockchain networks), and an information service provider 770 (e.g., an oracle or notary service for importing external information, fact verification, or other suitable information into a blockchain network).

[74] As shown, the timing service 710 may first notify the participants of the plurality of blockchain networks a commencement of the gridlock resolution process by submitting blockchain transactions to the plurality of blockchains (such as 720A and 720B) at steps 712A and 712B. The participants may observe (e.g., through corresponding computing systems 730A and 730B) such blockchain transactions and start the process of proposing settleable outgoing transactions.

[75] In some embodiments, the process may start with each participant first performing an initialization step to obtain, by a computing system associated with the participant from the blockchain, one or more first blockchain transactions respectively corresponding to one or more incoming value transfers associated with the participant and one or more second blockchain transactions respectively corresponding to one or more outgoing value transfers associated with the participant.

[76] Referring back to FIG. 7, the initialization step may include steps 722AA and 722AB. For example, the computing system 730A may look up the blockchain 720A hosted by the corresponding blockchain network and retrieve the corresponding participant's all incoming

transactions (e.g., transactions listing the corresponding participant's blockchain account address as the payment recipient) and all outgoing transactions (e.g., transactions listing the participant's blockchain account address as the payment transferor) on the blockchain 720A at step 722AA. Similarly, the computing system 730B may also look up the blockchain 720A to collect its corresponding incoming transactions and outgoing transactions at step 722AB. Similarly, each of the participants such as 730A and 730B may perform the "look up" operations on all the blockchain networks (such as blockchain 720B) that host its blockchain accounts to obtain the participant's all incoming transactions and outgoing transactions on each of the blockchain networks.

[77] To simplify following descriptions in the specification, the incoming transactions collected by participant i (e.g., through participant i 's computing system) during iteration t are denoted as $RecIn_i^t$, and the outgoing transactions collected by participant i during iteration t are denoted as $PayOut_i^t$. It may be appreciated that iteration 0 (e.g., $t=0$) represents an initialization step (e.g., including 722AA and 722AB in FIG. 7). In some embodiments, the $PayOut_i^t$ of participant i may be formulated as a list of blockchain transactions, each comprising an index corresponding to the blockchain transaction, a homomorphically encrypted version of an amount of a corresponding value transfer, and an identifier associated with a recipient of the corresponding value transfer, where the homomorphically encrypted version of the transaction (e.g., payment) amount may be generated with a homomorphic commitment scheme or a homomorphic encryption scheme. For example the list of transactions may be represented as following:

$$[(Ind_{i,1}, Amt_{i,1}, Rec_{i,1}), \dots, (Ind_{i,m_i}, Amt_{i,m_i}, Rec_{i,m_i})]$$

where $Ind_{i,k}$ ($1 \leq k \leq m_i$) is a unique transaction index that uniquely identifies a transaction within the blockchain network, $Amt_{i,k}$ ($1 \leq k \leq m_i$) is the transaction amount, $Rec_{i,k}$ ($1 \leq k \leq m_i$) is the transferee (e.g., receiver) of this transaction, and m_i is a total number of transactions (i.e., outgoing transactions) in the $PayOut_i^t$. In some embodiments, each $Amt_{i,k}$ in the $PayOut_i^t$ may be protected by a homomorphic encryption or a homomorphic commitment scheme that supports computation on ciphertexts. For example, each $Amt_{i,k}$ may be protected by Pedersen Commitment.

[78] After the initialization step is performed, each of the participants may perform an iterative method comprising the following steps during each iteration.

[79] Step 1: proposing, by each computing system associated with the corresponding blockchain network (e.g., 730A and 730B associated with blockchain 720A), a list of outgoing transactions to settle during a current iteration t to the corresponding blockchain network. The proposal may be submitted as a blockchain transaction. For example, computing systems 730A and 730B may each submit to blockchain 720A a blockchain transaction at step 732A and 732B.

[80] In other words, Step 1 involves a selecting step to determine the list of outgoing transactions to propose to the blockchain, which may be a subset of all the participant's outgoing transactions. In some embodiments, this selecting step may include following sub-steps. Each of the sub-steps may be performed by each participant on each of the blockchain networks that host the participant's blockchain accounts.

[81] Sub-step 1: for participant i on blockchain network l during iteration t , calculating a minimum transfer amount $\Delta_i(l)$ that participant i needs to settle all outgoing transactions in $PayOut_i^{t-1}(l)$ by taking into account all incoming transactions in $RecIn_i^{t-1}(l)$ and the current balance of participant i 's blockchain account in the blockchain network l .

[82] In some embodiments, the $\Delta_i(l)$ may be calculated by obtaining a first total amount associated with the subset of the one or more first blockchain transactions (e.g., a total amount of incoming transactions in $RecIn_i^{t-1}(l)$) and a second total amount associated with the subset of the one or more second blockchain transactions (e.g., a total amount of outgoing transactions in $PayOut_i^{t-1}(l)$); determining the delta value $\Delta_i(l)$ by subtracting a sum of the first total amount and the balance of the blockchain account from the second total amount.

[83] For example, if participant i needs transfer-in from its other blockchain accounts on other blockchain networks in order to settle the proposed outgoing transactions in $PayOut_i^{t-1}(l)$, $\Delta_i(l)$ may have a positive value representing the amount to be transferred in; if participant i may transfer out a certain amount (e.g., surplus) after settling all proposed outgoing transactions in $PayOut_i^{t-1}(l)$, $\Delta_i(l)$ may have a negative value representing the amount

offered to transfer out. This exemplary configuration may be reversed or changed depending on the implementation.

[84] Sub-step 2: for participant i on blockchain network l during iteration t , assigning $PayOut_i^t(l) = PayOut_i^{t-1}(l)$, where $PayOut_i^t(l)$ may be updated by the following sub-steps.

[85] Sub-step 3: for participant i during iteration t , calculate the sum of delta values for all the networks which it holds an account, if $\sum_l \Delta_i(l) > 0$ (e.g., the sum of all $\Delta_i(l)$ of participant i from all the blockchain networks that participant i participates is positive), removing the lowest priority transaction from $\cup_l PayOut_i^t(l)$ (e.g., a union of all outgoing transactions of participant i from all the blockchain networks that participant i participates); and if $\sum_l \Delta_i(l) \leq 0$, jumping to sub-step 5. In some embodiments, the transaction with the lowest priority may refer to the transaction with the newest submission time. In this sub-step, the “removing the lowest priority transaction from $\cup_l PayOut_i^t(l)$ ” may be understood as the transaction with the lowest priority will not be considered again during the current gridlock resolution process (even though it may be considered in the future processes). This removed transaction may be associated with one of the blockchain networks that the participant engaged in, thus the “removing the lowest priority transaction from $\cup_l PayOut_i^t(l)$ ” may update (e.g., by removing a transaction from) one of the $PayOut_i^t$ lists.

[86] Sub-step 4: for the participant i on the blockchain network l during iteration t , recalculating $\Delta_i(l)$ based on $RecIn_i^{t-1}(l)$ and $PayOut_i^t(l)$ for all blockchain networks, and looping back to sub-step 3.

[87] Sub-step 5: for the participant i during iteration t , adjusting $\Delta_i(l)$ for each blockchain network l so that $\sum_l \Delta_i(l) = 0$. It may be noted that sub-step 5 may be reached only when $\sum_l \Delta_i(l) \leq 0$, which means some participant i 's blockchain accounts may offer excessive amounts to other blockchain accounts (e.g., more than the other blockchain accounts need to settle their proposed outgoing transactions). In this case, certain participant i 's blockchain accounts may reduce its proposed transfer-out amount to make $\sum_l \Delta_i(l) = 0$.

[88] Sub-step 6: for the participant i on the blockchain network l during iteration t , adding to the blockchain a third blockchain transaction comprising one or more identifiers corresponding to the proposed subset of the one or more second blockchain transactions and an encrypted version of the delta value. The third blockchain transaction may refer to a proposal from participant i , the proposal including a list of outgoing transactions to each of the blockchain networks comprising $\{PayOut_i^t(l), COM(\Delta_i(l)), \pi_l\}$, where $PayOut_i^t(l)$ is the proposed subset of the one or more second blockchain transactions (e.g., after skipping low priority transactions), $COM(\Delta_i(l))$ is $\Delta_i(l)$ protected by the Pedersen Commitment (e.g., for simplicity, the random number required by the Pedersen Commitment is omitted), and π_l is a zero-knowledge range proof of a post-balance of participant i 's blockchain account on that network is not negative. The post-balance may be determined based on a difference between a first sum and a second sum, the first sum comprising the balance of the blockchain account on the blockchain and a first total amount associated with the subset of the one or more first blockchain transactions, the second sum comprising a second total amount associated with the subset of the one or more second blockchain transactions and the delta value. For example, the post-balance may be determined by the following formula: the current balance participant i 's blockchain account on that network minus the total outgoing amount in $PayOut_i^t(l)$, plus the total incoming amount in $RecIn_i(l)$ and $\Delta_i(l)$. In some embodiments, π_l may be optional. In some embodiments, the proposal may also include a second range proof that a sum of a global set of delta values associated with the entity equals zero (e.g., a proof that $\sum_l \Delta_i(l) = 0$).

[89] As the above sub-steps described, the purpose of Step 1 is to propose a list of outgoing transactions of participant i across all the blockchain networks to settle during the current iteration. In order to settle the list of proposed outgoing transactions, participant i may need to redistribute its blockchain account balances on the blockchain networks without creating or losing any asset (e.g., guaranteed by $\sum_l \Delta_i(l) = 0$). The proposal on blockchain network l may be based on a presumption that the incoming transactions in the $RecIn_i(l)$ are all available to participant i (e.g., participant i presumes it will receive all of the incoming transactions in $RecIn_i(l)$). It may be appreciated that $RecIn_i(l)$ may change after all participants submit their proposals. For example, participant i may realize some of its incoming transactions are unavailable if the corresponding transferors (e.g., other participants) do not propose the

corresponding outgoing transactions (e.g., the transactions with participant i as the recipient) to settle.

[90] In some embodiments, instead of cyclically determining $\Delta_i(l)$ for each blockchain network by removing (e.g., skipping) one lowest priority outgoing transaction at a time, participant i in Step 1 may directly determine a global set of outgoing transactions that participant i may settle by considering its global current balance and a global set of incoming transactions (e.g., $\cup_l RecIn_i(l)$). The determination may skip a minimum number of outgoing transactions with lower priorities so that a total sum of the global current balance and the global set of incoming transactions is no less than a total sum of the remaining outgoing transactions. The remaining outgoing transactions may be referred to as the global set of outgoing transactions. After determining the global set of outgoing transactions, participant i may submit to each of the blockchain networks a blockchain transaction (e.g., making a proposal) comprising a subset of the global set of outgoing transactions that are associated with the each blockchain network. The selection of the global set of outgoing transactions may be performed by one of participant i 's computing systems by exchanging information (e.g., current balances, incoming and outgoing transactions) with participant i 's other computing systems.

[91] Under this embodiment, the Step 1 may be formulated as: determining a first total amount by aggregating a plurality of values associated with a global set of first blockchain transactions available to the participant or entity (the available first blockchain transactions may be determined based on one or more blockchain transactions (e.g., third blockchain transactions as described under sub-step 6 above) sent by the transferors of the first blockchain transactions identifying proposed outgoing transactions of the transferors), wherein the global set of first blockchain transactions comprises the one or more first blockchain transactions for each of the plurality of blockchains; determining a second total amount by aggregating a global set of balances, wherein the global set of balances comprises the balance of the blockchain account of the entity on each of the plurality of blockchains selecting one or more second blockchain transactions from a global set of second blockchain transactions such that a sum of one or more values associated with the selected one or more second blockchain transactions is no greater than a sum of the first total amount and the second total amount, wherein the global set of second

blockchain transactions comprises the obtained one or more second blockchain transactions for each of the plurality of blockchains.

[92] Step 2: on each of the blockchain networks, aggregating, through a blockchain contract, the proposals submitted from all the participants in the blockchain network. In some embodiments, the blockchain contract may submit a blockchain transaction to the blockchain network comprising a list of aggregated outgoing transactions that the participants proposed to settle at 724A. The list of aggregated outgoing transactions may be represented as a list of the transaction indexes of the aggregated outgoing transactions.

[93] In some embodiments, the list of aggregated outgoing transactions received during the current iteration may be compared against the list of aggregated outgoing transactions received during the previous iteration in order to detect convergence (e.g., convergence occurs when the two lists are the same). The solution/result of the gridlock resolution process is reached when the blockchain networks are all converged. In some embodiments, this comparison may be performed by the timing service 710.

[94] Step 3: obtaining, by the computing system from each of the blockchain networks, one or more blockchain transactions generated by the blockchain contract, the one or more blockchain transactions comprising a plurality of identifiers; and determining the subset of the one or more first blockchain transactions (e.g., incoming transactions) based on the plurality of identifiers, repeating Steps 1-3 until a global convergence is observed. The one or more blockchain transactions may refer to the blockchain transaction submitted by the blockchain contract after aggregating the outgoing transactions proposed during the most recent iteration by the participants in the blockchain. For example, these aggregated outgoing transactions may allow each participant to determine whether the incoming transactions in the $RecIn_i^t(l)$ (for the participant i on the blockchain network l during iteration t) from the previous iteration t are available, and adjust $RecIn_i^{t+1}(l)$ for the next iteration $t + 1$ by skipping the unavailable incoming transactions (e.g., the transactions that are not proposed to settle by the transferor). With $RecIn_i^{t+1}(l)$ being adjusted, participant i may accordingly adjust its proposal.

[95] As shown in FIG. 7, the computing system 730A and the computing system 730B may observe and retrieve the blockchain transaction submitted by the blockchain contract at 726AA

and 726AB, respectively, where the blockchain transaction comprises the outgoing transactions proposed by all the participants of the blockchain 720A during the previous iteration.

[96] Step 4: once a global convergence is observed, terminating the gridlock resolution process. In some embodiments, the blockchain contract on each of the blockchains (e.g., 720A, 720B) may determine whether there is a local convergence (e.g., whether the proposals received from the participant in the corresponding blockchain remain the same in the previous two or more consecutive iterations). It may be noted that, a blockchain that reaches a local convergence at one iteration may diverge in the next iteration. Therefore, the global convergence may be observed when each of the blockchains reaches its local convergence during the same iteration.

[97] In some embodiments, each local convergence may result in a blockchain event being logged. The event is a dispatched signal that a blockchain contract may fire, and may be listened or monitored by a party connected to the blockchain network (e.g., decentralized applications connected to Ethereum JSON-RPC API). The timing service 710 may monitor each of the blockchains to observe the blockchain events indicating local convergences during the same iteration (e.g., at steps 714A and 714B in FIG. 7). In some embodiments, if a global convergence is observed, the timing service 710 may submit, through its blockchain nodes, a blockchain transaction to each of the blockchains (e.g., 720A and 720B) to announce a result of the gridlock resolution has been reached. In some embodiments, the blockchain transaction submitted to a blockchain may comprise the locally converged proposals received from the participants of the blockchain. Each proposal may include a participant's list of outgoing transactions to be settled, and a delta value indicating an amount to be transferred into or out of the participant's blockchain account on the blockchain. In some embodiments, the delta value in the each proposal may be represented as a homomorphic commitment, or a homomorphically encrypted value based on a public key shared by the participants (e.g., a regulator's public key).

[98] Step 5: settling, by each participant on each blockchain network, the proposed outgoing transactions. This step may be triggered in various ways. In some embodiments, each participant may monitor (e.g., through its computing systems) the blockchains for the global convergence blockchain transaction submitted by the timing service 710. Once such blockchain transaction is observed, the participant may proceed to settle the outgoing transactions it proposed during the

last iteration. Since each proposal comprises the delta value indicating assets redistribution among the participant's blockchain accounts, the participant may perform the redistribution first, followed by settling the proposed outgoing transactions. These operations (e.g., assets redistribution and settling outgoing transactions) may include multiple steps such as updating account balances and actual assets transfers. These steps may be carried out in a synchronous manner (e.g., the assets are transferred and the account balances are updated at the same time) or an asynchronous manner (e.g., the account balances are updated in a timely manner to reflect the assets transfers and the cleared transactions, but the actual assets transfers may occur at a later time). For example, the blockchain networks may support creation and transaction of blockchain based assets (e.g., tokens, cryptocurrencies). One or more participants of a common blockchain network may execute one or more asset transactions to each other by adding one or more blockchain transactions to a blockchain associated with the blockchain network. By way of the blockchain transactions, blockchain-based assets may be transferred among blockchain accounts of the participants and the balances of the blockchain accounts are updated accordingly. In another example, the blockchain networks may be used to record off-blockchain asset ownership and transactions. One or more participants may engage in asset transactions independent of a blockchain network and record the details of the asset transactions on the blockchain associated with the blockchain network. Alternatively, one or more participants may commit to asset transactions by adding one or more blockchain transactions representing the asset transactions to the blockchain. The one or more participants may later clear the committed asset transactions via one or more off-blockchain channels.

[99] In some embodiments, an information service provider such as a notary service provider or an oracle 770 may be adopted to observe the global convergence, verify that each participant does not create or destroy any asset or liquidity, sign on the gridlock result, and submit blockchain transactions with its signature to the blockchains (e.g., at steps 772A and 772B) to enable each participant to perform the assets redistribution and outgoing transaction settlement. For example, since the final proposals (e.g., the proposals shown in the *(b) Cross-Network Gridlock Resolution* section of FIG. 8) submitted by each participant on each blockchain network not only include the transaction identifiers, but also the amount to be transferred in or out of the participant's blockchain account on the blockchain network (i.e., $\Delta_i(l)$ for participant i 's

blockchain account on blockchain j), the information service provider (e.g., the oracle 770) may verify that for each participant i, its aggregated transferring amount should be equal to 0, i.e., $\sum_l \Delta_i(l) = 0$, where $l \in \{\text{all the blockchain networks}\}$. The $\Delta_i(l)$ in some embodiments may be protected by Peterson Commitment (or another suitable homomorphic encryption/commitment scheme) as $COM(\Delta_i(l))$. In this case, the $\sum_l \Delta_i(l) = 0$ may be verified by $\prod_l COM(\Delta_i(l)) = 1$, where $\prod$ refers to a product operation. Once the proposals are verified, the information service provider may sign the proposals with its privacy key and submit the signatures back to the corresponding blockchain networks in the form of blockchain transactions. Subsequently, the smart contract on each blockchain network may verify the received signatures based on the information service provider's public key. Once the verifications are successful, the smart contract may settle all the payment messages in the aggregated set and update each participant's account balance according to the corresponding delta value and aggregation results.

[100] FIG. 8 illustrates an example application of blockchain-base gridlock resolution in accordance with some embodiments. The (a) *Before Netting* section of FIG. 8 illustrates an initial setup including two blockchain networks X and Y and four participants A, B, C, and D (e.g., banks). As shown, A, B, C participate in network X and A, B, D participate in network Y. A's blockchain account in network X has one outgoing transaction obligation represented as:

$$(Ind(2), Amt(Com(12,*)), Rec(C))$$

where $Ind(2)$ is a unique identifier of this outgoing transaction within the network X, $Amt(Com(12,*))$ is a Peterson Commitment protected transaction amount (in this case, 12), and $Rec(C)$ specifies the recipient of the transaction (in this case, C). It may be noted that the transaction amount may be protected by another suitable homomorphic encryption scheme or homomorphic commitment scheme, which is not limited in this specification. For simplicity, the required random number for a Peterson Commitment representation is represented as *.

[101] Similarly, C's blockchain account in network X has one outgoing transaction obligation $(Ind(1), Amt(Com(5,*)), Rec(B))$, representing a payment from C to B in network X for an amount of 5. B's blockchain account in network Y has one outgoing transaction $(Ind(1), Amt(Com(20,*)), Rec(D))$, representing a payment from B to D in network Y for an

amount of 20. D's blockchain account in network Y has one outgoing transaction $(Ind(2), Amt(Com(25,*)), Rec(A))$, representing a payment from D to A in network Y for an amount of 25.

[102] Furthermore, A's blockchain account in network X has a current balance 5, which is also protected by Peterson Commitment in the form of $Com(5,*)$ as shown in FIG. 8. Similarly, B and C each has a current balance of $Com(5,*)$ in network X, A, B, and D each has a current balance of $Com(5,*)$ in network Y.

[103] As described in FIGs. 1-7, a gridlock resolution process may start when the participants receives a signal (e.g., by observing a blockchain transaction submitted by a timing service 360 in FIG. 6, or by following a pre-negotiated schedule). The gridlock resolution process may comprise a plurality of iterations. During each iteration, each participant proposes a list of its outgoing transactions to settle.

[104] Referring to the *(b) Cross-Network Gridlock Resolution* section of FIG. 8, during the first iteration (e.g., round 1 proposal), A first collects all its incoming transactions, all its outgoing transactions, and all its current balances. In this case, A has one incoming transaction from D in network Y: $(Ind(2), Amt(Com(25,*)), Rec(A))$, and one outgoing transactions: $(Ind(2), Amt(Com(12,*)), Rec(C))$ on network X. A has two current balances corresponding to its two blockchain accounts: 5 for each. Then A presumes that all its incoming transactions will be received (e.g., paid by the transferor), and determines the outgoing transactions it may settle. In this case, the total incoming transactions to A may yield an amount of 25 (from D), and the total current balance of A is 10 (5 from network X and 5 from network Y), therefore A's total assets available is 35. Since A only has one outgoing transaction of 12 to C, it has sufficient assets to settle $(Ind(2), Amt(Com(12,*)), Rec(C))$. However, A's current balance in network X is only 5, and there is no incoming transaction to A in network X. Thus, in order to settle the outgoing transaction, A may determine a delta value (e.g., $\Delta_A(X)$) as the amount to be transferred from A's another blockchain account. In this case, A may need 7 (e.g., $12-5=7$) to be redistributed from its blockchain account in network Y to its blockchain account in network X. As a result, A (e.g., through A's computing system associated with network X) may submit a blockchain transaction to network X comprising the following proposal:

$$\{Payout = [Ind(2)], Delta = COM(7,*), \pi_A\}$$

where *Payout* refers to a list of A's outgoing transactions to settle in network X, *Delta* specifies an amount to be transferred into or out of A's blockchain account in network X (e.g., a positive 7 means A needs to transfer in 7), and π_A refers to a range proof that A's post-balance after settling the outgoing transactions in *Payout* is not negative. In some embodiments, π_A may be optional. In this case, *Payout* has the outgoing transaction index *Ind*(2) (referring to the outgoing transaction from A to C in network X).

[105] Similarly, A may also submit a blockchain transaction to network Y comprising the following proposal:

$$\{Payout = [], Delta = COM(-7,*), \pi_A\}$$

where *Payout* here is empty because A does not have outgoing transaction to settle in network Y, *Delta* has a negative 7 indicating that A proposes to transfer out 7 from its blockchain account in network Y, and π_A is the optional range proof.

[106] Following the similar process, participants B, C, D may also make their proposals during the first iteration as shown in FIG. 8. For C and D, since they only participate in one network, their *Delta* are 0 (indicating they will not transfer in or out any amount).

[107] These proposals (e.g., blockchain transactions) may be submitted to a blockchain contract in each of the networks. This blockchain contract may aggregate all the outgoing transactions proposed by all the participants in the corresponding network. For example, after the first iteration, the blockchain contract in network X may aggregate a list of proposed outgoing transactions, including $\{Ind(2), Ind(1)\}$, where *Ind*(2) is proposed by A, and *Ind*(1) is proposed by C. Similarly, the blockchain contract in network Y may aggregate a list including $\{Ind(1), Ind(2)\}$, where *Ind*(1) is proposed by B, and *Ind*(2) is proposed by D.

[108] This aggregated lists of outgoing transactions may be used by each of the participants in the next iteration to determine whether its presumed incoming transactions are actually available. For example, in iteration 1, A presumes that its incoming transaction from D in network Y: (*Ind*(2), *Amt*(*Com*(25,*)), *Rec*(A)) is available. After iteration 1, the aggregated lists show that

the transaction $Ind(1)$ in network Y is proposed by D, which means this incoming transaction is still available during the next iteration. As a result, A may again make the same proposal as it did last iteration (as the incoming transactions, current balances, and outgoing transactions have not changed) in both network X and network Y.

[109] Following the similar process, the incoming transactions for B in network X and Y also don't change in iteration 2, and thus B may make the same proposals as it did in iteration 1. Similarly, C and D may also make the same proposals in iteration 2 as they did in iteration 1.

[110] In some embodiments, each of the blockchain contracts on the networks may determine whether the proposals received are converged (e.g., whether the proposals received during this iteration are the same as the ones received during the last iteration). If convergence is observed, the each blockchain contract may log a blockchain event to announce a local convergence corresponding to the network. The timing service or another suitable cross-network coordinator may observe such local convergences events and determine whether a global convergence has been reached. In this case, both networks have observed convergences, and thus the global convergence has been reached, and a result of the gridlock resolution has been found. For example, after a notary service confirms that the sum of deltas for each participant across network X and Y is zero, a netting process may be triggered by a signature from the notary to settle all the participant's proposals and updates their post-balance simultaneously within each network. As shown in the (c) *After Netting* section of FIG. 8, A may transfer an amount of 7 from its blockchain account in network Y to its blockchain account on network X; B may transfer an amount of 15 from its blockchain account in network X to its blockchain account on network Y. The netting may be required to actually settle (e.g., pay) the proposed outgoing transactions.

[111] **FIG. 9** illustrates an example method for implementing a blockchain-base gridlock resolution in accordance with some embodiments. The method 900 may be performed by a device, apparatus, or system for optimizing resource allocation. The method 900 may be performed by one or more modules/components of the environment or system illustrated by FIGs. 1-8, such as the computing system 400 in FIG. 4. The operations of the method 900 presented

below are intended to be illustrative. Depending on the implementation, the method 900 may include additional, fewer, or alternative steps performed in various orders or in parallel.

[112] Block 910 includes for each blockchain of a plurality of blockchains: obtaining, by a computing system associated with an entity from the blockchain, one or more first blockchain transactions respectively corresponding to one or more incoming value transfers associated with the entity and one or more second blockchain transactions respectively corresponding to one or more outgoing value transfers associated with the entity.

[113] Block 920 includes conducting, by the computing system associated with the entity, one or more iterations of a transaction-selection process. In some embodiments, each of the one or more iterations of the transaction-selection process comprises: determining a subset of the one or more first blockchain transactions; selecting a subset of the one or more second blockchain transactions based on the determined subset of the one or more first blockchain transactions and a balance of a blockchain account of the entity on the blockchain; determining a delta value associated with the blockchain based on the subset of the one or more first blockchain transactions, the subset of the one or more second blockchain transactions, and the balance of the blockchain account on the blockchain, wherein the delta value corresponds to a value transfer between the blockchain account on the blockchain and one or more different blockchain accounts of the entity on one or more different blockchains of the plurality of blockchains; and adding to the blockchain a third blockchain transaction comprising one or more identifiers corresponding to the selected subset of the one or more second blockchain transactions and an encrypted version of the delta value.

[114] In some embodiments, the adding to the blockchain a third blockchain transaction comprises: sending the third blockchain transaction to one or more blockchain nodes associated with blockchain to add to the blockchain, wherein the third blockchain transaction invokes a blockchain contract on the blockchain, and wherein the blockchain contract is executable to aggregate the one or more identifiers in the third blockchain transaction with a plurality of other identifiers in a plurality of other blockchain transactions.

[115] In some embodiments, the determining a subset of the one or more first blockchain transactions comprises: for a first iteration among the one or more iterations of the transaction-

selection process: determining the subset of the one or more first blockchain transactions as including each of the one or more first blockchain transactions; and for each of one or more iterations, other than the first iteration, among the one or more iterations of the transaction-selection process: obtaining, by the computing system from the blockchain, one or more blockchain transactions generated by the blockchain contract, the one or more blockchain transactions comprising a plurality of identifiers; and determining the subset of the one or more first blockchain transactions based on the plurality of identifiers.

[116] In some embodiments, the selecting a subset of the one or more second blockchain transactions comprises: determining a first total amount by aggregating a plurality of values associated with a global set of first blockchain transactions, wherein the global set of first blockchain transactions comprises the one or more first blockchain transactions for each of the plurality of blockchains; determining a second total amount by aggregating a global set of balances, wherein the global set of balances comprises the balance of the blockchain account of the entity on each of the plurality of blockchains; and selecting one or more second blockchain transactions from a global set of second blockchain transactions such that a sum of one or more values associated with the selected one or more second blockchain transactions is no greater than a sum of the first total amount and the second total amount, wherein the global set of second blockchain transactions comprises the obtained one or more second blockchain transactions for each of the plurality of blockchains.

[117] In some embodiments, the global set of second blockchain transactions are respectively associated with priority rankings; and the selecting the one or more second blockchain transactions from the global set of second blockchain transactions comprises selecting the one or more second blockchain transactions based on the priority rankings.

[118] In some embodiments, the priority rankings of the global set of second blockchain transactions are determined chronologically.

[119] In some embodiments, the determining a delta value associated with the blockchain comprises: obtaining a first total amount associated with the subset of the one or more first blockchain transactions and a second total amount associated with the subset of the one or more

second blockchain transactions; determining the delta value by subtracting a sum of the first total amount and the balance of the blockchain account from the second total amount.

[120] In some embodiments, the encrypted version of the delta value is generated based on a homomorphic commitment scheme or a homomorphic encryption scheme.

[121] In some embodiments, each blockchain transaction of the one or more first blockchain transactions and the one or more second blockchain transactions comprises: an index corresponding to the blockchain transaction, a homomorphically encrypted version of an amount of corresponding value transfer, and an identifier associated with a recipient of the corresponding value transfer, wherein the homomorphically encrypted version of the value transfer amount based on a homomorphic commitment scheme or a homomorphic encryption scheme.

[122] In some embodiments, the third blockchain transaction further comprises a zero-knowledge range proof that a first sum is not less than a second sum, the first sum comprising the balance of the blockchain account on the blockchain and a first total amount associated with the subset of the one or more first blockchain transactions, the second sum comprising a second total amount associated with the subset of the one or more second blockchain transactions and the delta value.

[123] Block 930 includes terminating the one or more iterations of the transaction-selection process in response to obtaining a fourth blockchain transaction from the blockchain, the fourth blockchain transaction indicating convergence of transaction selection on each of the plurality of blockchains.

[124] In some embodiments, the method 900 may further comprise, prior to conducting the one or more iterations of the transaction-selection process: obtaining, by the computing system, a blockchain transaction in each of the blockchains indicating a beginning of the transaction-selection process.

[125] In some embodiments, the method 900 may further comprise, for each blockchain of the plurality of blockchains: obtaining, from the blockchain, a fifth blockchain transaction comprising a proof that a sum of a global set of delta values associated with the entity equals

zero, wherein the global set of delta values comprises the delta value associated with each of the plurality of blockchains.

[126] FIG. 10 illustrates an example block diagram of a computing system for implementing a blockchain-base gridlock resolution in accordance with some embodiments. The components of the computer system 1000 presented below are intended to be illustrative. Depending on the implementation, the computer system 1000 may include additional, fewer, or alternative components.

[127] The computer system 1000 may be an example of an implementation of one or more components of the computing system 400. The flows and methods illustrated in FIGs. 1-9 may be implemented by the computer system 1000. The computer system 1000 may comprise one or more processors and one or more non-transitory computer-readable storage media (e.g., one or more memories) coupled to the one or more processors and configured with instructions executable by the one or more processors to cause the system or device (e.g., the processor) to perform the above-described method, e.g., the method 900. The computer system 1000 may comprise various units/modules corresponding to the instructions (e.g., software instructions).

[128] In some embodiments, the computer system 1000 may be referred to as an apparatus for executing a blockchain-based gridlock resolution. The apparatus may comprise an obtaining module 1010, a transaction selection module 1020, and a termination module 1030. In some embodiments, the obtaining module 1010 may, for each blockchain of a plurality of blockchains obtain one or more first blockchain transactions respectively corresponding to one or more incoming value transfers associated with the entity and one or more second blockchain transactions respectively corresponding to one or more outgoing value transfers associated with the entity.

[129] In some embodiments, the transaction selection module 1020 may conduct one or more iterations of a transaction-selection process, wherein each of the one or more iterations of the transaction-selection process comprises: determining a subset of the one or more first blockchain transactions; selecting a subset of the one or more second blockchain transactions based on the determined subset of the one or more first blockchain transactions and a balance of a blockchain account of the entity on the blockchain; determining a delta value associated with the blockchain

based on the subset of the one or more first blockchain transactions, the subset of the one or more second blockchain transactions, and the balance of the blockchain account on the blockchain, wherein the delta value corresponds to a value transfer between the blockchain account on the blockchain and one or more different blockchain accounts of the entity on one or more different blockchains of the plurality of blockchains; and adding to the blockchain a third blockchain transaction comprising one or more identifiers corresponding to the selected subset of the one or more second blockchain transactions and an encrypted version of the delta value.

[130] In some embodiments, the termination module 1030 may terminate the one or more iterations of the transaction-selection process in response to obtaining a fourth blockchain transaction from the blockchain, the fourth blockchain transaction indicating convergence of transaction selection on each of the plurality of blockchains.

[131] The techniques described herein may be implemented by one or more special-purpose computing devices. The special-purpose computing devices may be desktop computer systems, server computer systems, portable computer systems, handheld devices, networking devices or any other device or combination of devices that incorporate hard-wired and/or program logic to implement the techniques. The special-purpose computing devices may be implemented as personal computers, laptops, cellular phones, camera phones, smart phones, personal digital assistants, media players, navigation devices, email devices, game consoles, tablet computers, wearable devices, or a combination thereof. Computing device(s) may be generally controlled and coordinated by operating system software. Conventional operating systems control and schedule computer processes for execution, perform memory management, provide file system, networking, I/O services, and provide a user interface functionality, such as a graphical user interface (“GUI”), among other things. The various systems, apparatuses, storage media, modules, and units described herein may be implemented in the special-purpose computing devices, or one or more computing chips of the one or more special-purpose computing devices. In some embodiments, the instructions described herein may be implemented in a virtual machine on the special-purpose computing device. When executed, the instructions may cause the special-purpose computing device to perform various methods described herein. The virtual machine may include a software, hardware, or a combination thereof.

[132] FIG. 11 illustrates an example block diagram of a computer system in which any of the embodiments described herein may be implemented. The computing device may be used to implement one or more components of the systems and the methods shown in FIGs. 1-10 The computing device 1100 may comprise a bus 1102 or other communication mechanism for communicating information and one or more hardware processors 1104 coupled with bus 1102 for processing information. Hardware processor(s) 1104 may be, for example, one or more general purpose microprocessors.

[133] The computing device 1100 may also include a main memory 1107, such as a random-access memory (RAM), cache and/or other dynamic storage devices, coupled to bus 1102 for storing information and instructions to be executed by processor(s) 1104. Main memory 1107 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor(s) 1104. Such instructions, when stored in storage media accessible to processor(s) 1104, may render computing device 1100 into a special-purpose machine that is customized to perform the operations specified in the instructions. Main memory 1107 may include non-volatile media and/or volatile media. Non-volatile media may include, for example, optical or magnetic disks. Volatile media may include dynamic memory. Common forms of media may include, for example, a floppy disk, a flexible disk, hard disk, solid state drive, magnetic tape, or any other magnetic data storage medium, a CD-ROM, any other optical data storage medium, any physical medium with patterns of holes, a RAM, a DRAM, a PROM, and EPROM, a FLASH-EPROM, NVRAM, any other memory chip or cartridge, or networked versions of the same.

[134] The computing device 1100 may implement the techniques described herein using customized hard-wired logic, one or more ASICs or FPGAs, firmware and/or program logic which in combination with the computing device may cause or program computing device 1100 to be a special-purpose machine. According to one embodiment, the techniques herein are performed by computing device 1100 in response to processor(s) 1104 executing one or more sequences of one or more instructions contained in main memory 1107. Such instructions may be read into main memory 1107 from another storage medium, such as storage device 1109. Execution of the sequences of instructions contained in main memory 1107 may cause processor(s) 1104 to perform the process steps described herein. For example, the

processes/methods disclosed herein may be implemented by computer program instructions stored in main memory 1107. When these instructions are executed by processor(s) 1104, they may perform the steps as shown in corresponding figures and described above. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions.

[135] The computing device 1100 also includes a communication interface 1110 coupled to bus 1102. Communication interface 1110 may provide a two-way data communication coupling to one or more network links that are connected to one or more networks. As another example, communication interface 1110 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN (or WAN component to communicate with a WAN). Wireless links may also be implemented.

[136] The performance of certain of the operations may be distributed among the processors, not only residing within a single machine, but deployed across a number of machines. In some example embodiments, the processors or processor-implemented engines may be located in a single geographic location (e.g., within a home environment, an office environment, or a server farm). In other example embodiments, the processors or processor-implemented engines may be distributed across a number of geographic locations.

[137] Each of the processes, methods, and algorithms described in the preceding sections may be embodied in, and fully or partially automated by, code modules executed by one or more computer systems or computer processors comprising computer hardware. The processes and algorithms may be implemented partially or wholly in application-specific circuitry.

[138] When the functions disclosed herein are implemented in the form of software functional units and sold or used as independent products, they can be stored in a processor executable non-volatile computer readable storage medium. Particular technical solutions disclosed herein (in whole or in part) or aspects that contributes to current technologies may be embodied in the form of a software product. The software product may be stored in a storage medium, comprising a number of instructions to cause a computing device (which may be a personal computer, a server, a network device, and the like) to execute all or some steps of the methods of the embodiments of the present application. The storage medium may comprise a flash drive, a portable hard drive,

ROM, RAM, a magnetic disk, an optical disc, another medium operable to store program code, or any combination thereof.

[139] Particular embodiments further provide a system comprising a processor and a non-transitory computer-readable storage medium storing instructions executable by the processor to cause the system to perform operations corresponding to steps in any method of the embodiments disclosed above. Particular embodiments further provide a non-transitory computer-readable storage medium configured with instructions executable by one or more processors to cause the one or more processors to perform operations corresponding to steps in any method of the embodiments disclosed above.

[140] Embodiments disclosed herein may be implemented through a cloud platform, a server or a server group (hereinafter collectively the “service system”) that interacts with a client. The client may be a terminal device, or a client registered by a user at a platform, wherein the terminal device may be a mobile terminal, a personal computer (PC), and any device that may be installed with a platform application program.

[141] The various features and processes described above may be used independently of one another or may be combined in various ways. All possible combinations and sub-combinations are intended to fall within the scope of this disclosure. In addition, certain method or process blocks may be omitted in some implementations. The methods and processes described herein are also not limited to any particular sequence, and the blocks or states relating thereto can be performed in other sequences that are appropriate. For example, described blocks or states may be performed in an order other than that specifically disclosed, or multiple blocks or states may be combined in a single block or state. The example blocks or states may be performed in serial, in parallel, or in some other manner. Blocks or states may be added to or removed from the disclosed example embodiments. The exemplary systems and components described herein may be configured differently than described. For example, elements may be added to, removed from, or rearranged compared to the disclosed example embodiments.

[142] The various operations of exemplary methods described herein may be performed, at least partially, by an algorithm. The algorithm may be comprised in program codes or instructions stored in a memory (e.g., a non-transitory computer-readable storage medium

described above). Such algorithm may comprise a machine learning algorithm. In some embodiments, a machine learning algorithm may not explicitly program computers to perform a function but can learn from training data to make a prediction model that performs the function.

[143] The various operations of exemplary methods described herein may be performed, at least partially, by one or more processors that are temporarily configured (e.g., by software) or permanently configured to perform the relevant operations. Whether temporarily or permanently configured, such processors may constitute processor-implemented engines that operate to perform one or more operations or functions described herein.

[144] Similarly, the methods described herein may be at least partially processor-implemented, with a particular processor or processors being an example of hardware. For example, at least some of the operations of a method may be performed by one or more processors or processor-implemented engines. Moreover, the one or more processors may also operate to support performance of the relevant operations in a “cloud computing” environment or as a “software as a service” (SaaS). For example, at least some of the operations may be performed by a group of computers (as examples of machines including processors), with these operations being accessible via a network (e.g., the Internet) and via one or more appropriate interfaces (e.g., an Application Program Interface (API)).

[145] The performance of certain of the operations may be distributed among the processors, not only residing within a single machine, but deployed across a number of machines. In some example embodiments, the processors or processor-implemented engines may be located in a single geographic location (e.g., within a home environment, an office environment, or a server farm). In other example embodiments, the processors or processor-implemented engines may be distributed across a number of geographic locations.

[146] Throughout this specification, plural instances may implement components, operations, or structures described as a single instance. Although individual operations of one or more methods are illustrated and described as separate operations, one or more of the individual operations may be performed concurrently, and nothing requires that the operations be performed in the order illustrated. Structures and functionality presented as separate components in example configurations may be implemented as a combined structure or component. Similarly,

structures and functionality presented as a single component may be implemented as separate components. These and other variations, modifications, additions, and improvements fall within the scope of the subject matter herein.

[147] Although an overview of the subject matter has been described with reference to specific example embodiments, various modifications and changes may be made to these embodiments without departing from the broader scope of embodiments of the present disclosure. Such embodiments of the subject matter may be referred to herein, individually or collectively, by the term “invention” merely for convenience and without intending to voluntarily limit the scope of this application to any single disclosure or concept if more than one is, in fact, disclosed.

[148] The embodiments illustrated herein are described in sufficient detail to enable those skilled in the art to practice the teachings disclosed. Other embodiments may be used and derived therefrom, such that structural and logical substitutions and changes may be made without departing from the scope of this disclosure. The Detailed Description, therefore, is not to be taken in a limiting sense, and the scope of various embodiments is defined only by the appended claims, along with the full range of equivalents to which such claims are entitled.

[149] Any process descriptions, elements, or blocks in the flow diagrams described herein and/or depicted in the attached figures should be understood as potentially representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or steps in the process. Alternate implementations are included within the scope of the embodiments described herein in which elements or functions may be deleted, executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those skilled in the art.

[150] As used herein, “or” is inclusive and not exclusive, unless expressly indicated otherwise or indicated otherwise by context. Therefore, herein, “A, B, or C” means “A, B, A and B, A and C, B and C, or A, B, and C,” unless expressly indicated otherwise or indicated otherwise by context. Moreover, “and” is both joint and several, unless expressly indicated otherwise or indicated otherwise by context. Therefore, herein, “A and B” means “A and B, jointly or severally,” unless expressly indicated otherwise or indicated otherwise by context. Moreover,

plural instances may be provided for resources, operations, or structures described herein as a single instance. Additionally, boundaries between various resources, operations, engines, and data stores are somewhat arbitrary, and particular operations are illustrated in a context of specific illustrative configurations. Other allocations of functionality are envisioned and may fall within a scope of various embodiments of the present disclosure. In general, structures and functionality presented as separate resources in the example configurations may be implemented as a combined structure or resource. Similarly, structures and functionality presented as a single resource may be implemented as separate resources. These and other variations, modifications, additions, and improvements fall within a scope of embodiments of the present disclosure as represented by the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

[151] The term “include” or “comprise” is used to indicate the existence of the subsequently declared features, but it does not exclude the addition of other features. Conditional language, such as, among others, “can,” “could,” “might,” or “may,” unless specifically stated otherwise, or otherwise understood within the context as used, is generally intended to convey that certain embodiments include, while other embodiments do not include, certain features, elements and/or steps. Thus, such conditional language is not generally intended to imply that features, elements and/or steps are in any way required for one or more embodiments or that one or more embodiments necessarily include logic for deciding, with or without user input or prompting, whether these features, elements and/or steps are included or are to be performed in any particular embodiment.

CLAIMS:

What is claimed is:

1. A computer-implemented method for blockchain-based gridlock resolution, comprising, for each blockchain of a plurality of blockchains:

obtaining, by a computing system associated with an entity from the blockchain, one or more first blockchain transactions respectively corresponding to one or more incoming value transfers associated with the entity and one or more second blockchain transactions respectively corresponding to one or more outgoing value transfers associated with the entity;

conducting, by the computing system associated with the entity, one or more iterations of a transaction-selection process, wherein each of the one or more iterations of the transaction-selection process comprises:

determining a subset of the one or more first blockchain transactions;

selecting a subset of the one or more second blockchain transactions based on the determined subset of the one or more first blockchain transactions and a balance of a blockchain account of the entity on the blockchain;

determining a delta value associated with the blockchain based on the subset of the one or more first blockchain transactions, the subset of the one or more second blockchain transactions, and the balance of the blockchain account on the blockchain, wherein the delta value corresponds to a value transfer between the blockchain account on the blockchain and one or more different blockchain accounts of the entity on one or more different blockchains of the plurality of blockchains; and

adding to the blockchain a third blockchain transaction comprising one or more identifiers corresponding to the selected subset of the one or more second blockchain transactions and an encrypted version of the delta value; and

terminating the one or more iterations of the transaction-selection process in response to obtaining a fourth blockchain transaction from the blockchain, the fourth blockchain transaction indicating convergence of transaction selection on each of the plurality of blockchains.

2. The method of claim 1, further comprising, prior to conducting the one or more iterations of the transaction-selection process:

obtaining, by the computing system, a blockchain transaction in each of the blockchains indicating a beginning of the transaction-selection process.

3. The method of any of the claims 1-2, wherein the adding to the blockchain a third blockchain transaction comprises:

 sending the third blockchain transaction to one or more blockchain nodes associated with the blockchain to add to the blockchain, wherein the third blockchain transaction invokes a blockchain contract on the blockchain, and wherein the blockchain contract is executable to aggregate the one or more identifiers in the third blockchain transaction with a plurality of other identifiers in a plurality of other blockchain transactions.

4. The method of claim 3, wherein the determining a subset of the one or more first blockchain transactions comprises:

 for a first iteration among the one or more iterations of the transaction-selection process:
 determining the subset of the one or more first blockchain transactions as including each of the one or more first blockchain transactions; and

 for each of one or more iterations, other than the first iteration, among the one or more iterations of the transaction-selection process:

 obtaining, by the computing system from the blockchain, one or more blockchain transactions generated by the blockchain contract, the one or more blockchain transactions comprising a plurality of identifiers; and

 determining the subset of the one or more first blockchain transactions based on the plurality of identifiers.

5. The method of claim 4, wherein the selecting a subset of the one or more second blockchain transactions comprises:

 determining a first total amount by aggregating a plurality of values associated with a global set of first blockchain transactions, wherein the global set of first blockchain transactions comprises the one or more first blockchain transactions for each of the plurality of blockchains;

determining a second total amount by aggregating a global set of balances, wherein the global set of balances comprises the balance of the blockchain account of the entity on each of the plurality of blockchains; and

selecting one or more second blockchain transactions from a global set of second blockchain transactions such that a sum of one or more values associated with the selected one or more second blockchain transactions is no greater than a sum of the first total amount and the second total amount, wherein the global set of second blockchain transactions comprises the obtained one or more second blockchain transactions for each of the plurality of blockchains.

6. The method of claim 5, wherein:

the global set of second blockchain transactions are respectively associated with priority rankings; and

the selecting the one or more second blockchain transactions from the global set of second blockchain transactions comprises selecting the one or more second blockchain transactions based on the priority rankings.

7. The method of claim 6, wherein the priority rankings of the global set of second blockchain transactions are determined chronologically.

8. The method of any of the claims 1-4, wherein the determining a delta value associated with the blockchain comprises:

obtaining a first total amount associated with the subset of the one or more first blockchain transactions and a second total amount associated with the subset of the one or more second blockchain transactions;

determining the delta value by subtracting a sum of the first total amount and the balance of the blockchain account from the second total amount.

9. The method of any of the claims 1-8, wherein the encrypted version of the delta value is generated based on a homomorphic commitment scheme or a homomorphic encryption scheme.

10. The method of any of the claims 1-9, wherein each blockchain transaction of the one or more first blockchain transactions and the one or more second blockchain transactions comprises:

- an index corresponding to the blockchain transaction,
- a homomorphically encrypted version of an amount of corresponding value transfer, and
- an identifier associated with a recipient of the corresponding value transfer,

wherein the homomorphically encrypted version of the value transfer amount is based on a homomorphic commitment scheme or a homomorphic encryption scheme.

11. The method of claim 1, wherein the third blockchain transaction further comprises a zero-knowledge range proof that a first sum is not less than a second sum, the first sum comprising the balance of the blockchain account on the blockchain and a first total amount associated with the subset of the one or more first blockchain transactions, the second sum comprising a second total amount associated with the subset of the one or more second blockchain transactions and the delta value.

12. The method of any of the claims 1-4, further comprising, for each blockchain of the plurality of blockchains:

- obtaining, from the blockchain, a proof that a sum of a global set of delta values associated with the entity equals zero, wherein the global set of delta values comprises the delta value associated with each of the plurality of blockchains.

13. A non-transitory computer-readable storage medium configured with instructions executable by one or more processors to cause the one or more processors to perform the method of any one of the claims 1-12.

14. An apparatus for blockchain-based gridlock resolution comprising a plurality of modules for performing the method of any one of the claims 1-12.

15. A system for blockchain-based gridlock resolution, comprising one or more processors and one or more non-transitory computer-readable memories coupled to the one or

more processors and configured with instructions executable by the one or more processors to cause the system to perform operations comprising, for each blockchain of a plurality of blockchains:

- obtaining, by a computing system associated with an entity from the blockchain, one or more first blockchain transactions respectively corresponding to one or more incoming value transfers associated with the entity and one or more second blockchain transactions respectively corresponding to one or more outgoing value transfers associated with the entity;

- conducting, by the computing system associated with the entity, one or more iterations of a transaction-selection process, wherein each of the one or more iterations of the transaction-selection process comprises:

 - determining a subset of the one or more first blockchain transactions;

 - selecting a subset of the one or more second blockchain transactions based on the determined subset of the one or more first blockchain transactions and a balance of a blockchain account of the entity on the blockchain;

 - determining a delta value associated with the blockchain based on the subset of the one or more first blockchain transactions, the subset of the one or more second blockchain transactions, and the balance of the blockchain account on the blockchain, wherein the delta value corresponds to a value transfer between the blockchain account on the blockchain and one or more different blockchain accounts of the entity on one or more different blockchains of the plurality of blockchains; and

 - adding to the blockchain a third blockchain transaction comprising one or more identifiers corresponding to the selected subset of the one or more second blockchain transactions and an encrypted version of the delta value; and

- terminating the one or more iterations of the transaction-selection process in response to obtaining a fourth blockchain transaction from the blockchain, the fourth blockchain transaction indicating convergence of transaction selection on each of the plurality of blockchains.

16. The system of claim 15, wherein the operations further comprise, prior to conducting the one or more iterations of the transaction-selection process:

- obtaining, by the computing system, a blockchain transaction in each of the blockchains indicating a beginning of the transaction-selection process.

17. The system of any of the claims 15-16, wherein the adding to the blockchain a third blockchain transaction comprises:

 sending the third blockchain transaction to one or more blockchain nodes associated with blockchain to add to the blockchain, wherein the third blockchain transaction invokes a blockchain contract on the blockchain, and wherein the blockchain contract is executable to aggregate the one or more identifiers in the third blockchain transaction with a plurality of other identifiers in a plurality of other blockchain transactions.

18. The system of claim 17, wherein the determining a subset of the one or more first blockchain transactions comprises:

 for a first iteration among the one or more iterations of the transaction-selection process:
 determining the subset of the one or more first blockchain transactions as including each of the one or more first blockchain transactions; and

 for each of one or more iterations, other than the first iteration, among the one or more iterations of the transaction-selection process:

 obtaining, by the computing system from the blockchain, one or more blockchain transactions generated by the blockchain contract, the one or more blockchain transactions comprising a plurality of identifiers; and

 determining the subset of the one or more first blockchain transactions based on the plurality of identifiers.

19. The system of claim 18, wherein the selecting a subset of the one or more second blockchain transactions comprises:

 determining a first total amount by aggregating a plurality of values associated with a global set of first blockchain transactions, wherein the global set of first blockchain transactions comprises the one or more first blockchain transactions for each of the plurality of blockchains;

 determining a second total amount by aggregating a global set of balances, wherein the global set of balances comprises the balance of the blockchain account of the entity on each of the plurality of blockchains; and

selecting one or more second blockchain transactions from a global set of second blockchain transactions such that a sum of one or more values associated with the selected one or more second blockchain transactions is no greater than a sum of the first total amount and the second total amount, wherein the global set of second blockchain transactions comprises the obtained one or more second blockchain transactions for each of the plurality of blockchains.

20. The system of any of the claims 15-18, wherein the determining a delta value associated with the blockchain comprises:

obtaining a first total amount associated with the subset of the one or more first blockchain transactions and a second total amount associated with the subset of the one or more second blockchain transactions;

determining the delta value by subtracting a sum of the first total amount and the balance of the blockchain account from the second total amount.