Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 3033793 C 2021/06/15

(11)(21) 3 033 793

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2017/08/14
(87) Date publication PCT/PCT Publication Date: 2018/02/22
(45) Date de délivrance/Issue Date: 2021/06/15
(85) Entrée phase nationale/National Entry: 2019/02/13
(86) N° demande PCT/PCT Application No.: CA 2017/050962
(87) N° publication PCT/PCT Publication No.: 2018/032097
(30) Priorités/Priorities: 2016/08/14 (US62/374,907); 2017/03/30 (US15/474,785); 2017/08/11 (US15/675,041)

(51) Cl.Int./Int.Cl. *H04L 12/751* (2013.01), *G06Q 20/38* (2012.01), *G06Q 40/02* (2012.01), *H04L 12/725* (2013.01)

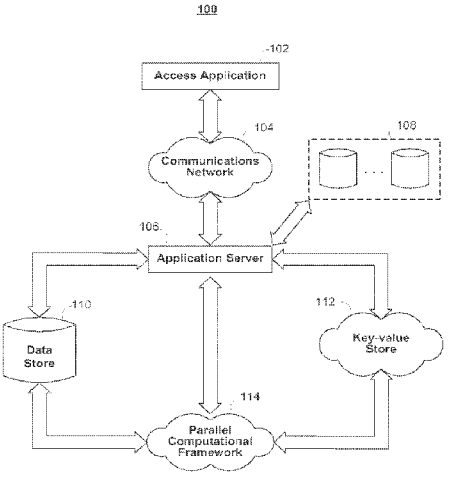
(72) Inventeurs/Inventors:
CHRAPKO, EVAN V., CA;
CHAN, LEO M., CA

(73) Propriétaire/Owner:
WWW.TRUSTSCIENCE.COM INC., CA

(74) Agent: SMART & BIGGAR LLP

(54) Titre : NOTATION DE LA FIABILITE, DE LA COMPETENCE ET/OU DE LA COMPATIBILITE D'UNE ENTITE QUELCONQUE POUR DES ACTIVITES COMPRENANT DES DECISIONS DE RECRUTEMENT OU D'EMBAUCHE, UN SUIVI DE SAUT, UNE SOUSCRIPTION D'ASSURANCE, DES DECISIONS DE CREDIT OU UN RACCOURCISSEMENT OU L'AMELIORATION DE CYCLES DE VENTE

(54) Title: SCORING TRUSTWORTHINESS, COMPETENCE, AND/OR COMPATIBILITY OF ANY ENTITY FOR ACTIVITIES INCLUDING RECRUITING OR HIRING DECISIONS, SKIP TRACING, INSURANCE UNDERWRITING, CREDIT DECISIONS, OR SHORTENING OR IMPROVING SALES CYCLES



(57) Abrégé/Abstract:
Systems and methods for recruiting, counter-terrorism/security, insurance underwriting, sales and marketing improvement, decisioning financial transactions and collections, and social scoring are provided. Machine learning can assign connectivity values to other community members, including individuals, companies, products, brands, cities or neighborhoods, etc. Connectivity values may be automatically harvested from or assigned by third parties or based on the frequency and/or type of interactions between community members. Connectivity values may represent such factors as alignment, reputation within the community, degree of trust, competence at one or more skills, or compatibility with others. The degree and type of connectivity between two entities may be assessed by computing a connectivity value based upon connections between entities and relative or absolute trust, competence and/or compatibility features of the connections. Connectivity values identify best prospects (customers, hires, dates), find off-grid people, underwrite insurance, 'decision' loans & collections, shorten sales cycles, etc.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 February 2018 (22.02.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/032097 A1

(51) International Patent Classification:
H04L 12/751 (2013.01) *G06Q 40/02* (2012.01)
G06Q 20/38 (2012.01) *H04L 12/725* (2013.01)

(21) International Application Number:
PCT/CA2017/050962

(22) International Filing Date:
14 August 2017 (14.08.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

62/374,907	14 August 2016 (14.08.2016)	US
15/474,785	30 March 2017 (30.03.2017)	US
15/675,041	11 August 2017 (11.08.2017)	US

(71) Applicant: **WWW.TRUSTSCIENCE.COM INC.** [CA/CA]; #128, Bldg. 14, 9977-178 Street, Edmonton, Alberta T5T 6J6 (CA).

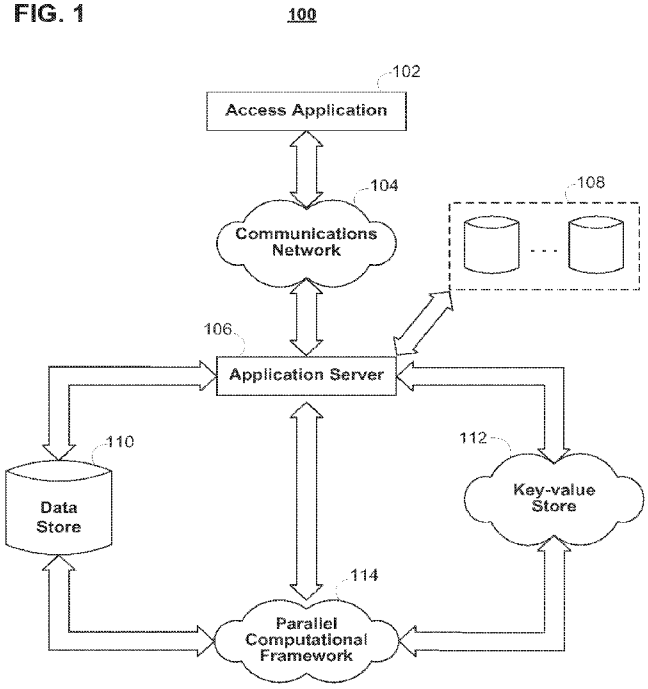
(72) Inventors: **CHRAPKO, Evan V**; #128, Bldg. 14, 9977-178 Street, Edmonton, Alberta T5T 6J6 (CA). **CHAN, Leo M.**; #30, 4755 Terwillegar Common, Edmonton, Alberta T6R 3V6 (CA).

(74) Agent: **SMART & BIGGAR**; P.O. Box 2999, Station D, 900-55 Metcalfe Street, Ottawa, Ontario K1P 5Y6 (CA).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME,

(54) Title: SCORING TRUSTWORTHINESS, COMPETENCE, AND/OR COMPATIBILITY OF ANY ENTITY FOR ACTIVITIES INCLUDING RECRUITING OR HIRING DECISIONS, SKIP TRACING, INSURANCE UNDERWRITING, CREDIT DECISIONS, OR SHORTENING OR IMPROVING SALES CYCLES

FIG. 1



(57) Abstract: Systems and methods for recruiting, counter-terrorism/security, insurance underwriting, sales and marketing improvement, decisioning financial transactions and collections, and social scoring are provided. Machine learning can assign connectivity values to other community members, including individuals, companies, products, brands, cities or neighborhoods, etc. Connectivity values may be automatically harvested from or assigned by third parties or based on the frequency and/or type of interactions between community members. Connectivity values may represent such factors as alignment, reputation within the community, degree of trust, competence at one or more skills, or compatibility with others. The degree and type of connectivity between two entities may be assessed by computing a connectivity value based upon connections between entities and relative or absolute trust, competence and/or

[Continued on next page]

WO 2018/032097 A1

WO 2018/032097 A1

MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

compatibility features of the connections. Connectivity values identify best prospects (customers, hires, dates), find off-grid people, underwrite insurance, 'decision' loans & collections, shorten sales cycles, etc.

TITLE: SCORING TRUSTWORTHINESS, COMPETENCE, AND/OR COMPATIBILITY OF ANY ENTITY FOR ACTIVITIES INCLUDING RECRUITING OR HIRING DECISIONS, SKIP TRACING, INSURANCE UNDERWRITING, CREDIT DECISIONS, OR SHORTENING OR IMPROVING SALES CYCLES

Background of the Invention

This invention relates generally to networks of individuals and/or entities and network communities and, more particularly, to systems and methods for determining trust scores or connectivity within or between individuals and/or entities or networks of individuals and/or entities and using these scores to facilitate financial transactions.

The connectivity, or relationships, of an individual or entity within a network community may be used to infer attributes of that individual or entity. For

Date Reçue/Date Received 2020-08-05

example, an individual or entity's connectivity within a network community may be used to determine the identity of the individual or entity (e.g., used to make decisions about identity claims and authentication), the trustworthiness or reputation of the individual, or the membership, status, and/or influence of that individual in a particular community or subset of a particular community.

An individual or entity's connectivity within a network community, however, is difficult to quantify. For example, network communities may include hundreds, thousands, millions, billions or more members. Each member may possess varying degrees of connectivity information about itself and possibly about other members of the community. Some of this information may be highly credible or objective, while other information may be less credible and subjective. In addition, connectivity information from community members may come in various forms and on various scales, making it difficult to meaningfully compare one member's "trustworthiness" or "competence" and connectivity information with another member's "trustworthiness" or "competence" and connectivity information. Also, many individuals may belong to multiple communities, further complicating the determination of a quantifiable representation of trust and connectivity within a network community. Similarly, a particular individual may be associated with duplicate entries in one or more communities, due to, for example, errors in personal information such as name/information misspellings and/or outdated personal information. Even if a quantifiable representation of an individual's connectivity is determined, it is often difficult to use this representation in a meaningful way to make real-world decisions about the individual (e.g., whether or not to trust the individual).

Further, it may be useful for these real-world decisions to be made prospectively (i.e., in advance of an anticipated event). Such prospective analysis may be difficult as an individual or entity's connectivity within a network community may change rapidly as the connections between the individual or entity and others in the network community may change quantitatively or qualitatively. This analysis becomes increasingly complex as if applied across multiple communities.

Summary of the Invention

In view of the foregoing, systems and methods are provided for determining the connectivity between nodes within a network community and inferring attributes, such as trustworthiness, compatibility, or competence, from the connectivity.

Connectivity may be determined, at least in part, using various graph traversal and normalization techniques described in more detail below and in U.S. Provisional Patent Application No. 61/247,343, filed September 30, 2009, U.S. Provisional Patent Application No. 61/254,313, filed October 23, 2009, International Patent Application No. CA2010001531, filed September 30, 2010, and International Patent Application No. CA2010001658, filed October 22, 2010.

In an embodiment, a path counting approach may be used where processing circuitry is configured to count the number of paths between a first node n_1 and a second node n_2 within a network community. A connectivity rating R_{n1n2} may then be assigned to the nodes. The assigned connectivity rating may be proportional to the number of subpaths, or relationships, connecting the two nodes, among other possible measures. Using the number of subpaths as a measure, a path with one or more intermediate nodes between the first node n_1 and the second node n_2 may be scaled by an appropriate number (e.g., the number of intermediate nodes) and this scaled number may be used to calculate the connectivity rating.

In some embodiments, weighted links are used in addition or as an alternative to the subpath counting approach. Processing circuitry may be configured to assign a relative user weight to each path connecting a first node n_1 and a second node n_2 within a network community. A user connectivity value may be assigned to each link. For example, a user or entity associated with node n_1 may assign user connectivity values for all outgoing paths from node n_1. In some embodiments, the connectivity values assigned by the user or entity may be indicative of that user or entity's trust in the user or entity associated with node n_2. The link values assigned by a particular user or entity may then be compared to each other to determine a relative user weight for each link.

The relative user weight for each link may be determined by first computing the average of all the user connectivity values assigned by that user or node (i.e., the out-link values). If t_i is the user connectivity value assigned to link i, then the relative user weight, w_i, assigned to that link may be given in accordance with:

$$w_i = 1 + (t_i - \bar{t}_i)^2 \quad (1)$$

In some embodiments, an alternative relative user weight, w_i', may be used based on the number of standard deviations, σ, the user connectivity value differs from the average

Date Reçue/Date Received 2020-08-05

value assigned by that user or node. For example, the alternative relative user weight may be given in accordance with:

$$w_i' = 1 - \frac{1}{2+k^2} \text{ where } k = \left\{ \frac{0, if\ \sigma = 0}{\frac{t_i - \bar{t}_i}{\sigma}, otherwise} \right\} \quad (2)$$

To determine the overall weight of a path, in some embodiments, the weights of all the links along the path may be multiplied together. The overall path weight may then be given in accordance with:

$$w_{path} = \Pi(w_i) \quad (3)$$

or

$$w_{path} = \Pi(w_i') \quad (4)$$

The connectivity value for the path may then be defined as the minimum user connectivity value of all the links in the path multiplied by the overall path weight in accordance with:

$$t_{path} = w_{path} \times t_{\min} \quad (5)$$

In some embodiments, only "qualified" paths are used to determine connectivity values. A qualified path may be a path whose path weight is greater than or equal to some threshold value. As described in more detail below, any suitable threshold function may be used to define threshold values. The threshold function may be based, at least in some embodiments, on empirical data, desired path keep percentages, or both. In some embodiments, threshold values may depend on the length, l, of the path. For example, an illustrative threshold function specifying the minimum path weight for path p may be given in accordance with:

$$threshold(p) = \begin{cases} 0.5, if\ l = 1 \\ 0.428, if\ l = 2 \\ 0.289, if\ l = 3 \\ 0.220, if\ l = 4 \\ 0.216, if\ l = 5 \\ 0.192, if\ l = 6 \end{cases} \quad (6)$$

To determine path connectivity values, in some embodiments, a parallel computational framework or distributed computational framework (or both) may be used. For example, in one embodiment, a number of core processors implement an Apache Hadoop or Google MapReduce cluster. This cluster may perform some or all of the distributed computations in connection with determining new path link values and path weights.

The processing circuitry may identify a changed node within a network community. For example, a new outgoing link may be added, a link may be removed, or a user connectivity value may have been changed. In response to identifying a changed node, in some embodiments, the processing circuitry may re-compute link, path, and weight values associated with some or all nodes in the implicated network community or communities.

In some embodiments, only values associated with affected nodes in the network community are recomputed after a changed node is identified. If there exists at least one changed node in the network community, the changed node or nodes may first undergo a prepare process. The prepare process may include a "map" phase and "reduce" phase. In the map phase of the prepare process, the prepare process may be divided into smaller sub-processes which are then distributed to a core in the parallel computational framework cluster. For example, each node or link change (e.g., tail to out-link change and head to in-link change) may be mapped to a different core for parallel computation. In the reduce phase of the prepare process, each out-link's weight may be determined in accordance with equation (1). Each of the out-link weights may then be normalized by the sum of the out-link weights (or any other suitable value). The node table may then be updated for each changed node, its in-links, and its out-links.

After the changed nodes have been prepared, the paths originating from each changed node may be calculated. Once again, a "map" and "reduce" phase of this process may be defined. During this process, in some embodiments, a depth-first search may be performed of the node digraph or node tree. All affected ancestor nodes may then be identified and their paths recalculated.

In some embodiments, to improve performance, paths may be grouped by the last node in the path. For example, all paths ending with node n_1 may be grouped together, all paths ending with node n_2 may be grouped together, and so on. These path groups may then be stored separately (e.g., in different columns of a single database table). In some embodiments, the path groups may be stored in columns of a key-value

store implementing an HBase cluster (or any other compressed, high performance database system, such as BigTable).

In some embodiments, one or more threshold functions may be defined. The threshold function or functions may be used to determine the maximum number of links in a path that will be analyzed in a connectivity determination or connectivity computation. Threshold factors may also be defined for minimum link weights, path weights, or both. Weights falling below a user-defined or system-defined threshold may be ignored in a connectivity determination or connectivity computation, while only weights of sufficient magnitude may be considered.

In some embodiments, a user connectivity value may represent the degree of trust between a first node and a second node. In one embodiment, node n_1 may assign a user connectivity value of l_1 to a link between it and node n_2. Node n_2 may also assign a user connectivity value of l_2 to a reverse link between it and node n_1. The values of l_1 and l_2 may be at least partially subjective indications of the trustworthiness, competence or compatibility of the individual or entity associated with the node connected by the link. For example, one or more of the individual's or entity's reputation within the network community (or some other community), the individual's or entity's alignment with the trusting party (e.g., political, social, religious, educational, or employment alignment), past dealings with the individual or entity, and the individual's or entity's character and integrity (or any other relevant considerations) may be used to determine a partially subjective user connectivity value indicative of trust, competence, or compatibility. A user (or other individual authorized by the node) may then assign this value to an outgoing link connecting the node to the individual or entity. Objective measures (e.g., data from third-party ratings agencies or credit bureaus) may also be used, in some embodiments, to form composite user connectivity values indicative of trust, competence or compatibility. The subjective, objective, or both types of measures may be automatically harvested or manually inputted for analysis.

In some embodiments, a decision-making algorithm may access the connectivity values in order to make automatic decisions (e.g., automatic network-based decisions, such as authentication or identity requests) on behalf of a user. Connectivity values may additionally or alternatively be outputted to external systems and processes located at third-parties. The external systems and processes may be configured to automatically initiate a transaction (or take some particular course of action) based, at least in part, on received connectivity values. For example, electronic or online

advertising, recruiting pitches, credit applications, insurance offers, sales offers, etc. may be targeted to subgroups of members of a network community based, at least in part, on network connectivity values.

As another example, the decision-making algorithm may take the form of a financial application, hiring application, sales application, etc., such as a loan, lending, or donation application. Connectivity values may be used by financial institutions to make automatic credit-granting, insurance underwriting, or security decisions. In some embodiments, connectivity values may be used in conjunction with third-party ratings agency information (e.g., credit bureau ratings information) in order to make credit-granting, insurance underwriting, sales, or hiring decisions. Connectivity values may also be used to advertise, promote, publish, or solicit information about charitable gifts, donations, or loans to other parties in a social networking environment or other network-based community. Decisions regarding loan amounts, interests rates, and/or loan repayment schedules may be automatically generated after a loan is approved and accepted by the financial application, the lender, or both the lender and financial application. Decisions regarding insurance amounts, rates, deductibles and/or other actuarial data may be automatically generated after an insurance policy is approved and accepted by the financial application, the underwriter, or both the underwriter and financial application. Decisions regarding recruiting, hiring, salary, benefits, and/or team assignments or recommendations may be automatically generated after a resume or job application is screened and accepted by a hiring application, a human resources professional, or both a professional and a hiring application. Decisions regarding sales, terms of sale, credit to extend, payment terms, shipping responsibility and/or other related information may be automatically generated after a customer, potential customer, purchase offer, or the like is screened and accepted by a sales application, a sales professional or both a professional and a sales application.

In some embodiments, a decision-making algorithm may access connectivity values to make decisions prospectively (e.g., before an anticipated event like a request for credit, request for insurance, request for reassignment within a company, request for sales terms, etc.). Such decisions may be made at the request of a user, or as part of an automated process (e.g., a credit bureau's periodic automated analysis of a database of customer information). This prospective analysis may allow for the initiation of a transaction (or taking of some particular action) in a fluid and/or dynamic manner.

In some embodiments, connectivity values may be used to present information to the user. This information may include, but is not limited to, static and/or interactive visualizations of connectivity values within a user's associated network community or communities. In some embodiments, this information may allow the user to explore or interact with an associated network community or communities, and encourage and/or discourage particular interactions within a user's associated network community or communities. In some embodiments, this information may explicitly present the user with the connectivity values. For example, a percentage may indicate how trustworthy another individual and/or entity is to a user. In some embodiments, the information may implicitly present the user with a representation of the connectivity values. For example, an avatar representing another individual and/or entity may change in appearance based on how trustworthy that individual and/or entity is to a user.

According to one aspect of the present invention, there is provided a method for facilitating financial transactions comprising: transmitting, to a first user and potential members of a publication group, software for communications relating to trust-based transactions; receiving with a server, a request from the first user's software to initiate a first trust-based transaction; automatically determining a publication group to publish at least some information relating to the transaction, wherein determining the publication group comprises: accessing, with at least one server, information in a datastore relating to a network community; identifying, with the at least one server, paths in the network community from the first user to at least one potential member of the publication group; and identifying sub-processes required to calculate a connectivity value between the first user and the at least one potential member, wherein identifying the sub-processes comprises: identifying a plurality of links in the identified paths; for one or more identified links, accessing a data structure to identify nodes connected to the identified link; and for each identified node, creating an indication of a sub-process, wherein the sub-process comprises calculating an out-link weight for one or more out-links of the identified node; distributing the indications of the sub-processes to a plurality of processors arranged in a parallel computational framework; receiving, from the plurality of processors, calculated out-link weights for the identified nodes; calculating the connectivity value based on the calculated out-link weights; and adding

Date Reçue/Date Received 2020-08-05

the at least one potential member to the publication group based on the calculated connectivity value; publishing the information relating to the transaction to the publication group; and transmitting, to the publication group's software, information related to the transaction with which information a member of the publication group may initiate a second trust-based transaction.

According to another aspect of the present invention, there is provided a system for facilitating financial transactions comprising processing circuitry configured to: receive with a server, a request from the first user's software to initiate a first trust-based transaction; automatically determine a publication group to publish at least some information relating to the transaction, wherein determining the publication group comprises: accessing, with at least one server, information in a datastore relating to a network community; identifying, with the at least one server, paths in the network community from the first user to at least one potential member of the publication group; and identifying sub-processes required to calculate a connectivity value between the first user and the at least one potential member, wherein identifying the sub-processes comprises: identifying a plurality of links in the identified paths; for one or more identified links, accessing a data structure to identify nodes connected to the identified link; and for each identified node, creating an indication of a sub-process, wherein the sub-process comprises calculating an out-link weight for one or more out-links of the identified node; distributing the indications of the sub-processes to a plurality of processors arranged in a parallel computational framework; receiving, from the plurality of processors, calculated out-link weights for the identified nodes; calculating the connectivity value based on the calculated out-link weights; and adding the at least one potential member to the publication group based on the calculated connectivity value; publish the information relating to the transaction to the publication group; and transmit, to the publication group's software, information related to the transaction with which information a member of the publication group may initiate a second trust-based transaction

According to another aspect of the present invention, there is provided a method for facilitating financial transactions comprising: transmitting, to a first user and potential members of a publication group, software for communications relating to trust-based transactions; receiving with a server, a request from the first user's software to initiate a first

Date Reçue/Date Received 2020-08-05

trust-based transaction; determining whether the first transaction is public; if the first transaction is public, automatically determining a publication group to publish at least some information relating to the transaction, wherein determining the publication group comprises: accessing, with at least one server, information in a datastore relating to a network community; identifying, with the at least one server, paths in the network community from the first user to at least one potential member of the publication group; and identifying sub-processes required to calculate a connectivity value between the first user and the at least one potential member, wherein identifying the sub-processes comprises: identifying a plurality of links in the identified paths; for one or more identified links, accessing a data structure to identify nodes connected to the identified link; and for each identified node, creating an indication of a sub-process, wherein the sub-process comprises calculating an out-link weight for one or more out-links of the identified node; distributing the indications of the sub-processes to a plurality of processors arranged in a parallel computational framework; receiving, from the plurality of processors, calculated out-link weights for the identified nodes; calculating the connectivity value based on the calculated out-link weights; and adding the at least one potential member to the publication group based on the calculated connectivity value.

Brief Description of the Drawings

The above and other features of the present invention, its nature and various advantages will be more apparent upon consideration of the following detailed description, taken in conjunction with the accompanying drawings, and in which:

FIG. 1 is an illustrative block diagram of a network architecture used to support connectivity within a network community in accordance with one embodiment of the invention;

FIG. 2 is another illustrative block diagram of a network architecture used to support connectivity within a network community in accordance with one embodiment of the invention;

Date Reçue/Date Received 2020-08-05

FIGS. 3A, 3B, and 3C show illustrative data tables for supporting connectivity determinations within a network community in accordance with one embodiment of the invention;

FIGS. 4A-4H show illustrative processes for supporting connectivity determinations within a network community in accordance with one embodiment of the invention;

FIG. 5 shows an illustrative process for querying all paths to a target node and computing a network connectivity value in accordance with one embodiment of the invention;

FIG. 6 shows an illustrative process for supporting user sign-in profiles in accordance with one embodiment of the invention; and

Date Reçue/Date Received 2020-08-05

FIG. 7 shows an illustrative process for facilitating financial transactions in accordance with one embodiment of the invention.

Detailed Description

Systems and methods for determining the connectivity between nodes in a network community are provided. As defined herein, a "node" may include any user terminal, network device, computer, mobile device, access point, robot, or any other electronic device capable of being uniquely identified within a network community. For example, nodes may include robots (or other machines) assigned unique serial numbers or network devices assigned unique network addresses. In some embodiments, a node may also represent an individual human being, entity (e.g., a legal entity, such as a public or private company, corporation, limited liability company (LLC), partnership, sole proprietorship, or charitable organization), concept (e.g., a social networking group, brand, advertising campaign, subgroup within a larger group), animal, city/town/village, parcel of land (which may be identified by land descriptions), or inanimate object (e.g., a car, aircraft, tool, other product, webpage, website, document, etc.). As also defined herein, a "network community" may include a collection of nodes and may represent any group of devices, individuals, or entities.

For example, all or some subset of the users of a social networking website or social networking service (or any other type of website or service, such as an online gaming community) may make up a single network community. Each user may be represented by a node in the network community. As another example, all the subscribers to a particular newsgroup or distribution list may make up a single network community, where each individual subscriber may be represented by a node in the network community. Any particular node may belong in zero, one, or more than one network community, or a node may be banned from all, or a subset of, the community. To facilitate network community additions, deletions, and link changes, in some embodiments a network community may be represented by a directed graph, or digraph, weighted digraph, tree, or any other suitable data structure.

FIG. 1 shows illustrative network architecture 100 used to support the connectivity determinations within a network community. A user may utilize access application 102 to access application server 106 over communications network 104. For example, access application 102 may include a standard web browser, application server 106 may include a web server, and communication network 106 may include the Internet. Access application 102 may also include proprietary applications specifically developed

for one or more platforms or devices. For example, access application 102 may include one or more instances of an Apple iOS, Android, or WebOS application or any suitable application for use in accessing application server 106 over communications network 104. Multiple users may access application server 106 via one or more instances of access application 102. For example, a plurality of mobile devices may each have an instance of access application 102 running locally on the devices. One or more users may use an instance of access application 102 to interact with application server 106.

Communication network 104 may include any wired or wireless network, such as the Internet, WiMax, wide area cellular, or local area wireless network. Communication network 104 may also include personal area networks, such as Bluetooth and infrared networks. Communications on communications network 104 may be encrypted or otherwise secured using any suitable security or encryption protocol.

Application server 106, which may include any network server or virtual server, such as a file or web server, may access data sources 108 locally or over any suitable network connection. Application server 106 may also include processing circuitry (e.g., one or more microprocessors), memory (e.g., RAM, ROM, and hybrid types of memory), storage devices (e.g., hard drives, optical drives, and tape drives). The processing circuitry included in application server 106 may execute a server process for supporting the network connectivity determinations of the present invention, while access application 102 executes a corresponding client process. The processing circuitry included in application server 106 may also perform any of the calculations and computations described herein in connection with determining network connectivity. In some embodiments, a computer-readable medium with computer program logic recorded thereon is included within application server 106. The computer program logic may determine the connectivity between two or more nodes in a network community and it may or may not output such connectivity to a display screen or data store.

For example, application server 106 may access data sources 108 over the Internet, a secured private LAN, or any other communications network. Data sources 108 may include one or more third-party data sources, such as data from third-party social networking services, third-party ratings bureaus, document issuers (e.g., driver's license and license plate issuers, such as the Department of Motor Vehicles), government records, privately compiled records, insurance databases, etc. For example, data sources 108 may include user and relationship data (e.g., "friend" or “follower” data) from one or more of Facebook, MySpace, openSocial, Friendster, Bebo, hi5, Orkut, PerfSpot, Yahoo!

360, Gmail, Yahoo! Mail, Hotmail, other email-based services and accounts, LinkedIn, Twitter, Snapchat, Instagram, Flickr, Monster, Upwork, Freelancer, Google Buzz, Really Simply Syndication readers, or any other social networking website or information service. Data sources 108 may also include data stores and databases local to application server 106 containing relationship information about users accessing application server 106 via access application 102 (e.g., databases of addresses, legal records, transportation passenger lists, gambling patterns, political affiliations, vehicle license plate or identification numbers, universal product codes, news articles, business listings, hospital affiliations, university affiliations, purchase histories, employer affiliations, insurance claims, credit requests, professional organization affiliations, or other organizational affiliations).

Application server 106 may be in communication with one or more of data store 110, key-value store 112, and parallel computational framework 114. Data store 110, which may include any relational database management system (RDBMS), file server, or storage system, may store information relating to one or more network communities. For example, one or more of data tables 300 (FIG. 3A) may be stored on data store 110. Data store 110 may store identity information about users and entities in the network community, an identification of the nodes in the network community, user link and path weights, user configuration settings, system configuration settings, and/or any other suitable information. There may be one instance of data store 110 per network community, or data store 110 may store information relating to a plural number of network communities. For example, data store 110 may include one database per network community, or one database may store information about all available network communities (e.g., information about one network community per database table).

Parallel computational framework 114, which may include any parallel or distributed computational framework or cluster, may be configured to divide computational jobs into smaller jobs to be performed simultaneously, in a distributed fashion, or both. For example, parallel computational framework 114 may support data-intensive distributed applications by implementing a map/reduce computational paradigm where the applications may be divided into a plurality of small fragments of work, each of which may be executed or re-executed on any core processor in a cluster of cores. A suitable example of parallel computational framework 114 includes an Apache Hadoop cluster.

Parallel computational framework 114 may interface with key-value store 112, which also may take the form of a cluster of cores. Key-value store 112 may hold sets of key-value pairs for use with the map/reduce computational paradigm implemented by parallel computational framework 114. For example, parallel computational framework 114 may express a large distributed computation as a sequence of distributed operations on data sets of key-value pairs. User-defined map/reduce jobs may be executed across a plurality of nodes in the cluster. The processing and computations described herein may be performed, at least in part, by any type of processor or combination of processors. For example, various types of quantum processors (e.g., solid-state quantum processors and light-based quantum processors), artificial neural networks, and the like may be used to perform massively parallel computing and processing.

In some embodiments, parallel computational framework 114 may support two distinct phases, a "map" phase and a "reduce" phase. The input to the computation may include a data set of key-value pairs stored at key-value store 112. In the map phase, parallel computational framework 114 may split, or divide, the input data set into a large number of fragments and assign each fragment to a map task. Parallel computational framework 114 may also distribute the map tasks across the cluster of nodes on which it operates. Each map task may consume key-value pairs from its assigned fragment and produce a set of intermediate key-value pairs. For each input key-value pair, the map task may invoke a user defined map function that transmutes the input into a different key-value pair. Following the map phase, parallel computational framework 114 may sort the intermediate data set by key and produce a collection of tuples so that all the values associated with a particular key appear together. Parallel computational framework 114 may also partition the collection of tuples into a number of fragments equal to the number of reduce tasks.

In the reduce phase, each reduce task may consume the fragment of tuples assigned to it. For each such tuple, the reduce task may invoke a user-defined reduce function that transmutes the tuple into an output key-value pair. Parallel computational framework 114 may then distribute the many reduce tasks across the cluster of nodes and provide the appropriate fragment of intermediate data to each reduce task.

Tasks in each phase may be executed in a fault-tolerant manner, so that if one or more nodes fail during a computation the tasks assigned to such failed nodes may

be redistributed across the remaining nodes. This behavior may allow for load balancing and for failed tasks to be re-executed with low runtime overhead.

Key-value store 112 may implement any distributed file system capable of storing large files reliably. For example key-value store 112 may implement Hadoop's own distributed file system (DFS) or a more scalable column-oriented distributed database, such as HBase. Such file systems or databases may include BigTable-like capabilities, such as support for an arbitrary number of table columns.

Although FIG. 1, in order to not over-complicate the drawing, only shows a single instance of access application 102, communications network 104, application server 106, data source 108, data store 110, key-value store 112, and parallel computational framework 114, in practice network architecture 100 may include multiple instances of one or more of the foregoing components. In addition, key-value store 112 and parallel computational framework 114 may also be removed, in some embodiments. As shown in network architecture 200 of FIG. 2, the parallel or distributed computations carried out by key-value store 112 and/or parallel computational framework 114 may be additionally or alternatively performed by a cluster of mobile devices 202 instead of stationary cores. In some embodiments, cluster of mobile devices 202, key-value store 112, and parallel computational framework 114 are all present in the network architecture. Certain application processes and computations may be performed by cluster of mobile devices 202 and certain other application processes and computations may be performed by key-value store 112 and parallel computational framework 114. In addition, in some embodiments, communication network 104 itself may perform some or all of the application processes and computations. For example, specially-configured routers or satellites may include processing circuitry adapted to carry out some or all of the application processes and computations described herein.

Cluster of mobile devices 202 may include one or more mobile devices, such as PDAs, cellular telephones, mobile computers, or any other mobile computing device. Cluster of mobile devices 202 may also include any appliance (e.g., audio/video systems, microwaves, refrigerators, food processors) containing a microprocessor (e.g., with spare processing time), storage, or both. Application server 106 may instruct devices within cluster of mobile devices 202 to perform computation, storage, or both in a similar fashion as would have been distributed to multiple fixed cores by parallel computational framework 114 and the map/reduce computational paradigm. Each device in cluster of mobile devices 202 may perform a discrete computational job, storage job, or

both. Application server 106 may combine the results of each distributed job and return a final result of the computation.

FIG. 3A shows illustrative data tables 300 used to support the connectivity determinations of the present invention. One or more of tables 300 may be stored in, for example, a relational database in data store 110 (FIG. 1). Table 302 may store an identification of all the nodes registered in the network community. A unique identifier may be assigned to each node and stored in table 302. In addition, a string name may be associated with each node and stored in table 302. As described above, in some embodiments, nodes may represent individuals or entities, in which case the string name may include the individual or person's first and/or last name, nickname, handle, or entity name.

Table 304 may store user connectivity values. User connectivity values may be positive, indicating some degree of trust between two or more parties, or may be negative, indicating some degree of distrust between two or more parties. In some embodiments, user connectivity values may be assigned automatically by the system (e.g., by application server 106 (FIG. 1)). For example, application server 106 (FIG. 1) may monitor all electronic interaction (e.g., electronic communication, electronic transactions, or both) between members of a network community. In some embodiments, a default user connectivity value (e.g., the link value 1) may be assigned initially to all links in the network community. After electronic interaction is identified between two or more nodes in the network community, user connectivity values may be adjusted upwards or downwards depending on the type of interaction between the nodes, the content of the interaction, and/or the result of the interaction. For example, each simple email exchange between two nodes may automatically increase or decrease the user connectivity values connecting those two nodes by a fixed amount. In some embodiments, the content of the emails in the email exchange may be processed by, for example, application server 106 (FIG. 1) to determine the direction of the user connectivity value change as well as its magnitude. For example, an email exchange regarding a transaction executed in a timely fashion may increase the user connectivity value, whereas an email exchange regarding a missed deadline may decrease the user connectivity value. The content of the email exchange or other interaction may be processed by using heuristic and/or data/text mining techniques to parse the content of the interaction. For example, a language parser may be used to identify keywords in the email exchange. In some embodiments, individual emails and/or the email exchange may be processed to identify keywords that are

associated with successful/favorable transactions and/or keywords that are associated with unsuccessful/unfavorable transactions, and the difference between the frequency/type of the keywords may affect the user connectivity value. In certain embodiments, natural language parsers may be used to extract semantic meaning from structured text in addition to keyword detection.

More complicated interactions (e.g., product or service sales or inquires) between two nodes may increase or decrease the user connectivity values connecting those two nodes by some larger fixed amount. In some embodiments, user connectivity values between two nodes may always be increased unless a user or node indicates that the interaction was unfavorable, not successfully completed, or otherwise adverse. For example, a transaction may not have been timely executed or an email exchange may have been particularly displeasing. Adverse interactions may automatically decrease user connectivity values while all other interactions may increase user connectivity values (or have no effect). In some embodiments, the magnitude of the user connectivity value change may be based on the content of the interactions. For example, a failed transaction involving a small monetary value may cause the user connectivity value to decrease less than a failed transaction involving a larger monetary value. In addition, user connectivity values may be automatically harvested using outside sources. For example, third-party data sources (such as ratings agencies and credit bureaus) may be automatically queried for connectivity information. This connectivity information may include completely objective information, completely subjective information, composite information that is partially objective and partially subjective, any other suitable connectivity information, or any combination of the foregoing.

In some embodiments, user connectivity values may be manually assigned by members of the network community. These values may represent, for example, the degree or level of trust, competence, or compatibility between two users or nodes or one node's assessment of another node's competence in some endeavor. As described above, user connectivity values may include a subjective component and an objective component in some embodiments. The subjective component may include a trustworthiness (or competence or compatibility) "score" indicative of how trustworthy (or competent or compatible) a first user or node finds a second user, node, community, or subcommunity. This score or value may be entirely subjective and based on interactions between the two users, nodes, or communities. A composite user connectivity value including subjective and objective components may also be used. For example, third-party information may

be consulted to form an objective component based on, for example, the number of consumer complaints, credit score, insurance claims, job applications, employment complaints or reprimands, defaults, product returns, awards or honors, socio-economic factors (e.g., age, income, political, religious or other affiliations, and criminal history), or number of citations/hits in the media or in search engine searches. Third-party information may be accessed using communications network 104 (FIG. 1). For example, a third-party credit bureau's database may be polled or a personal biography and background information, including criminal history information, may be accessed from a third-party database or data source (e.g., as part of data sources 108 (FIG. 1) or a separate data source) or input directly by a node, user, or system administrator. In some embodiments, the third-party data source(s) or system(s) may also include third-party user connectivity values and transaction histories, related to user interactions with the third-party system(s). In these embodiments, the user connectivity value or composite user connectivity value may also include one or more components based on the third-party user connectivity values and transaction histories.

Table 304 may store an identification of a link head, link tail, and user connectivity value for the link. Links may or may not be bidirectional. For example, a user connectivity value from node n_1 to node n_2 may be different (and completely separate) than a link from node n_2 to node n_1. Especially in the trust context described above, each user can assign his or her own user connectivity value to a link (i.e., two users need not trust each other an equal amount in some embodiments).

Table 306 may store an audit log of table 304. Table 306 may be analyzed to determine which nodes or links have changed in the network community. In some embodiments, a database trigger is used to automatically insert an audit record into table 306 whenever a change of the data in table 304 is detected. For example, a new link may be created, a link may be removed, and/or a user connectivity value may be changed. This audit log may allow for decisions related to connectivity values to be made prospectively (i.e., before an anticipated event). Such decisions may be made at the request of a user, or as part of an automated process, such as the processes described below with respect to FIG. 5. This prospective analysis may allow for the initiation of a transaction (or taking of some particular action) in a fluid and/or dynamic manner. After such a change is detected, the trigger may automatically create a new row in table 306. Table 306 may store an identification of the changed node, identification of the changed link head, changed link tail, and/or the user connectivity value to be assigned to the

changed link. Table 306 may also store a timestamp indicative of the time of the change and/or an operation code. In some embodiments, operation codes may include "insert," "update," and/or "delete" operations, corresponding to whether a link was inserted, a user connectivity value was changed, or a link was deleted, respectively. Other operation codes may be used in other embodiments.

FIG. 3B shows illustrative data structure 310 used to support the connectivity determinations of the present invention. In some embodiments, data structure 310 may be stored using key-value store 112 (FIG. 1), while tables 300 are stored in data store 110 (FIG. 1). As described above, key-value store 112 (FIG. 1) may implement an HBase storage system and include BigTable support. Like a traditional relational database management system, the data shown in FIG. 3B may be stored in tables. However, the BigTable support may allow for an arbitrary number of columns in each table, whereas traditional relational database management systems may require a fixed number of columns.

Data structure 310 may include node table 312. In the example shown in FIG. 3B, node table 312 includes several columns. Node table 312 may include row identifier column 314, which may store 64-bit, 128-bit, 256-bit, 512-bit, or 1024-bit integers and may be used to uniquely identify each row (e.g., each node) in node table 312. Column 316 may include a list of all the incoming links for the current node. Column 318 may include a list of all the outgoing links for the current node. Node table 312 may also include one or more "bucket" columns 320 and 322. These columns may store a list of paths that connect, for example, a source node to the current node, the current node to a target node, or both. As described above, grouping paths by the last node in the path (e.g., the target node), the first node in the path (e.g., the source node), or both, may facilitate connectivity computations. As shown in FIG. 3B, in some embodiments, to facilitate scanning, bucket column names may include the target node identifier appended to the end of the "bucket:" column name.

FIG. 3C shows illustrative database schema 330 used to facilitate financial transactions. Table 332 includes information related to users' sign-in profiles. For example, a user may have accounts for multiple email, social networking services, other online or network services, or any combination of the foregoing. Each of these accounts may be included in a separate sign-in profile associated with the user. As such, a single user may be associated with one or more sign-in profiles. In some embodiments, instead of including a distinct sign-in system specific to the connectivity system, a user may sign

in to one of these existing accounts or services identified in a sign-in profile, and then the connectivity system may ask the existing service to vouch for or verify the identity of the user. Table 332 may include a string identification of the service or provider associated with the profile, a unique identifier associated with the profile, an email or username field, and a nickname, handle, or real name field.

For example, a user may wish to log into the connectivity system (or some loan, insurance application, credit transaction, sales evaluation, or financial transaction system that uses the connectivity system) using access application 102 (FIG. 1). Application server 106 (FIG. 1) may then ask the user which service (of a list of available external services) to use for authentication. Application server 106 (FIG. 1) may then redirect the user to the external service's sign-in mechanism. The external service may then redirect the user back to the connectivity system (for example, a web page hosted by application server 106 (FIG. 1)). Application server 106 (FIG. 1) may then lookup the sign-in profile (e.g., in table 332) in order to identify the user.

Table 334 may include an indication of a person or node in the network community. For example, the person associated with table 334 may be an officer in a financial institution, a lender, a borrower, a donor, an insured, an underwriter, a buyer, or a seller. Officer table 336 may include a unique identifier representing the financial institution associated with the officer and identified in organization table 338. Donation table 340 and loan table 342 may include any suitable information related to donations or loans, respectively, available on the network. Donation table 340 may include such information as a unique identifier associated with a donation, a unique identifier associated with the donor, a unique identifier associated with the financial application, whether or not a tax receipt is needed, whether or not a tax receipt has been issued, the tax receipt number, the tax receipt date, and a status indicator. The status indicator may include "0" if the donation is still waiting for a check as a source of funding for the donation, a "1" if the donation is still waiting for an external payment system as a source of funding for the donation, "2" if the donation has been canceled by the user, the financial application, the officer, or financial institution, "3" if the donation is currently active, "4" is the donation has been completed, "5" if the donor has defaulted, "6" is the donation is associated with a refund amount.

Similarly, loan table 342 may include a unique identifier associated with a loan, a unique identifier associated with the financial application, a unique identifier associated with the lender, the principal of the loan, the balance of the loan (e.g., the

remaining principal on the loan), and a status indication. The status indicator may be the same as the status indicators described above with respect to the donation table. Financial application table 344 may identify the loans, donations, or other types of financial applications available in the network. Financial application table 344 may include a unique identifier for the application, a string description associated with the application (which may also include attribute flags and other metadata associated with the financial application and used in determining publication groups, as described in more detail with regard to FIG. 7 below), a unique borrower identifier, a currency type indication, the principal requested or available, the principal raised, the interest rate associated with the loan or donation, the payment period, the number of payment periods per year, and the number of compounding periods per year. Some fields in financial application table 344 may only apply to loan type applications or donation type applications.

In some embodiments, the description field in financial application table 344 may include "LIKE" and "DISLIKE" flags identifying affinity groups, blogs, newsgroups, and other information used to determine what nodes or users may be interested or not interested in a particular financial application. These flags may be used in determining publication groups, as described in more detail below. For example, a mortgage type financial application may include a "LIKE" flag for users or nodes interested in securing real property (e.g., users or nodes belonging to a real estate affinity group or real estate blog or newsgroup). As another example, a donation type financial application to support same-sex marriage may include a "LIKE" flag for users or nodes subscribed to the Human Rights Campaign or American Civil Liberties Union affinity group and a "DISLIKE" flag for users or nodes belonging to "Yes on Prop 8" or defense of marriage affinity group. Other attribute flags may also be defined in financial application table 344. These flags may be created by the sponsor or creator of the financial application and may be customized by users initiating financial transactions, in some embodiments.

Repayment schedule table 346 may be associated with each loan in loan table 342. Repayment schedule table 346 may include a unique identifier associated with the loan to which the repayment schedule relates, the current payment number, the due date for the net payment, the total amount due, and the total amount paid. Repayment schedule table 346 may be automatically generated, in some embodiments, whenever a new loan is created or initiated by a user and approved.

In a typical usage scenario, a user may be notified when certain users in the user's network have initiated a new financial transaction using a financial application identified in financial application table 344. For example, in some embodiments, users are notified whenever any other user initiates a financial transaction. In other embodiments, users are only notified about financial transactions made by other users meeting some threshold path weight or threshold user connectivity value with the to-be-notified user. For example, a message may be sent to second user that a first user has loaned $10,000 to "Save the Pandas" and that the specific financial application is the "Wildlife Sanctuary Project." This message may appear in email, as a pop-up message, or displayed as a link on the user's homepage, profile page, or initial log-in page.

The notified user may also decide to initiate a financial transaction using the same financial application. The user may then decide whether to fund the transaction using a check or using an external payment system (such as PayPal). Before the funding is received, the transaction may be marked as "waiting" for either a check or external payment system. For example, the status indicators in donation table 340 or loan table 342 may be set to "0" or "1". A repayment schedule may then be generated. For example, repayment schedule table 346 may be populated.

After funding has been received, the transaction may be marked as "active" and repayments may begin (depending on the transaction type). Repayments may be made, in some embodiments, by mailing a check, direct deposit, using an external payment system, or using any other suitable mechanism.

Although FIG. 3C shows one illustrative arrangement for schema 330, any other suitable schema may also be used. For example, more or fewer tables than those shown in FIG. 3C may be defined, each including more or fewer fields. In addition, although a relational database management system may be used in some embodiments to save and access information in accordance with schema 330, any other storage or access mechanism may be used in other embodiments.

FIGS. 4A-4H show illustrative processes for determining the connectivity of nodes within a network community. FIG. 4A shows process 400 for updating a connectivity graph (or any other suitable data structure) associated with a network community. As described above, in some embodiments, each network community is associated with its own connectivity graph, digraph, tree, or other suitable data structure. In other embodiments, a plurality of network communities may share one or more connectivity graphs (or other data structure).

In some embodiments, the processes described with respect to FIG. 4A-4H may be executed to make decisions prospectively (i.e., before an anticipated event). Such decisions may be made at the request of a user, or as part of an automated process, such as the processes described below with respect to FIG. 5. This prospective analysis may allow for the initiation of a transaction (or taking of some particular action) in a fluid and/or dynamic manner.

In some embodiments, the processes described with respect to FIG. 4A-4H may be executed to provide information to a user. Such presentations may be made at the request of a user, or as part of an automated presentation. This information may include, but is not limited to, static and/or interactive visualizations of connectivity values within a user's associated network community or communities. In some embodiments, this information may be integrated into explorations of or interactions within a user's associated network community or communities. Providing this information to a user may allow the user to better understand what other individuals and/or entities they may trust within a network community, and/or may encourage and/or discourage particular interactions within a user's associated network community or communities.

At step 402, a determination is made whether at least one node has changed in the network community. As described above, an audit record may be inserted into table 306 (FIG. 3) after a node has changed. By analyzing table 306 (FIG. 3), a determination may be made (e.g., by application server 106 of FIG. 1) that a new link has been added, an existing link has been removed, or a user connectivity value has changed. If, at step 404, it is determined that a node has changed, then process 400 may continue to step 410 (shown in FIG. 4B) to process the changed links, step 412 (shown in FIG. 4C) to save the nodes with changed links, step 414 (shown in FIG. 4D) to create path set input files, step 416 (shown in FIG. 4E) to remove paths with changed nodes, one or more iterations of step 418 (shown in FIG. 4F) to grow paths by one link at a time, step 420 (shown in FIG. 4G) to save the paths that have grown by one or more links, and step 422 (shown in FIG. 4H) to join paths that go through changed nodes. It should be noted that more than one step or task shown in FIGS. 4B, 4C, 4D, 4E, 4F, 4G, and 4H may be performed in parallel using, for example, a cluster of cores. For example, multiple steps or tasks shown in FIG. 4B may be executed in parallel or in a distributed fashion, then multiple steps or tasks shown in FIG. 4C may be executed in parallel or in a distributed fashion, then multiple steps or tasks shown in FIG. 4D may be executed in parallel or in a distributed fashion, then multiple steps or tasks shown in FIG. 4E may be executed in

parallel or in a distributed fashion, and so on. In this way, overall latency associated with process 400 may be reduced.

As described above, step 418 may be executed one or more times. This step may be operative to grow paths by a single link. Each iteration of step 418 may take as input the results of a previous iteration of step 418 so that paths may grow by more than one link, if desired. In the example of FIG. 4A, three iterations of step 418 are shown. Thus, process 400 may generate paths with lengths less than or equal to three. In other embodiments, more or fewer iterations of step 418 may allow process 400 to generate paths with more or fewer links.

If a node change is not detected at step 404, then process 400 enters a sleep mode at step 406. For example, in some embodiments, an application thread or process may continuously check to determine if at least one node or link has changed in the network community. In other embodiments, the application thread or process may periodically check for changed links and nodes every n seconds, where n is any positive number. After the paths are calculated that go through a changed node at step 416 or after a period of sleep at step 406, process 400 may determine whether or not to loop at step 408. For example, if all changed nodes have been updated, then process 400 may stop at step 418. If, however, there are more changed nodes or links to process, then process 400 may loop at step 408 and return to step 404.

In practice, one or more steps shown in process 400 may be combined with other steps, performed in any suitable order, performed in parallel (e.g., simultaneously or substantially simultaneously), or removed.

FIGS. 4B-4H each include processes with a "map" phase and "reduce" phase. As described above, these phases may form part of a map/reduce computational paradigm carried out by parallel computational framework 114 (FIG. 1), key-value store 112 (FIG. 1), or both. As shown in FIG. 4B, in order to process link changes, map phase 426 may include determining if there are any more link changes at step 428, retrieving the next link change at step 430, mapping the tail to out-link change at step 432, and mapping the head to in-link change at step 434.

If there are no more link changes at step 428, then, in reduce phase 436, a determination may be made at step 438 that there are more nodes with mapped link changes to process. If so, then the next node and its link changes may be retrieved at step 440. The most recent link changes may be preserved at step 442 while any intermediate link changes are replaced by more recent changes. For example, the timestamp stored in

table 306 (FIG. 3) may be used to determine the time of every link or node change. At step 444, the average out-link user connectivity value may be calculated. For example, if node n_1 has eight out-links with assigned user connectivity values, these eight user connectivity values may be averaged at step 444. At step 446, each out-link's weight may be calculated in accordance with equation (1) or (2) above. At step 448, an output file may be created or appended with the out-links changed and corresponding changed node identifier. For example, one or more (out-links changed, node identifier) records may be written to the output file. Although the term "file" is sometimes used herein, the output need not be in a literal file or even file format. For example, any output stream, whether or not it is recorded, may be used. In some embodiments, some or all of the output file may be passed directly to a calling application, process, or function from a returning application, process, or function in the form of a stream or object return value. If there are no more nodes and link changes to process at step 438, the process may stop at step 450.

As shown in FIG. 4C, in order to save nodes with changed links, map phase 452 may include determining if there are any more changed nodes at step 454, retrieving the next changed node at step 456, and mapping "null" to the node at step 458.

If there are no more changed nodes at step 454, then, in reduce phase 460, a determination may be made at step 462 that there are more nodes to process. If so, then the next node may be retrieved at step 464. At step 466, the in-links and out-links associated with the node may be written to a key-value store (e.g., key-value store 112 of FIG. 1). As described above, the key-value store may implement an HBase cluster (or any other compressed, high performance database system, such as BigTable). If there are no more nodes to process at step 462, the process may stop at step 468.

As shown in FIG. 4D, in order to create path set input files, map phase 470 may include determining if there are any more (out-links changed, node identifier) records in the output file created or appended at step 448 (FIG. 4B). If so, the next record may be retrieved at step 474. At step 476, a determination may be made if an out-link has changed. If so, then at step 478 a "null" value may be mapped to the node. Otherwise, map phase 470 may return to step 472 to determine if there are any more (out-links changed, node identifier) records in the output file.

If there are no more changed records at step 472, then, in reduce phase 480, a determination may be made at step 482 that there are more node to process. If so, then the next node may be retrieved at step 484. At step 486, new records may be written

to the output file. In some embodiments, the records written at step 486 may include records of the form (node identifier, empty path set for the node identifier). If there are no more nodes to process at step 482, the process may stop at step 488.

As shown in FIG. 4E, in order to remove paths with changed nodes, map phase 490 may include determining if there are any more (node identifier, path set) records in the output file at step 492 and retrieving the next such record at step 494. At step 496, for every "in" bucket identifier, the "in" bucket identifier may be mapped to a record of the form (out bucket type, node identifier, set of "out" bucket identifiers) (or any other suitable form). At step 498, for every "out" bucket identifier, the "out" bucket identifier may be mapped to a record of the form (in bucket type, node identifier, set of "in" bucket identifiers) (or any other suitable form). At step 500, the node's "out" buckets may be deleted, and the process may return to step 492 to determine if there are more records to process.

If there are no more records at step 492, then, in reduce phase 502, a determination may be made at step 504 that there are more node identifiers with their mapped (bucket type, changed node identifier, bucket identifiers) records to process. If so, then at step 506, if the bucket type is "out", out-buckets with the given bucket identifiers may be searched and paths with the changed node identifier may be removed. At step 508, if the bucket type is "in", in-buckets with the given bucket identifiers may be searched and paths with the changed node identifier may be removed. If there are no more records to process at step 504, the process may stop at step 510.

As shown in FIG. 4F, in order to grow paths by one link, map phase 512 may include determining if there are any more (node identifier, path set) records in the output file at step 514. If so, then at step 516, if the path set is empty, for each out-link of the node, a link head identifier may be mapped to the link. At step 518, if the path set is not empty, then for each path n in the path set, and for each out-link of a node, a new path may be created by appending (out-link, map link head identifier) to the new path.

If there are no more records at step 514, then, in reduce phase 520, a determination may be made at step 522 that there are more node identifiers with mapped paths to process. If so, then at step 524, new records of the form (node identifier, mapped paths) (or any other suitable form) may be written to the output file. If there are no more records to process at step 522, the process may stop at step 526.

The process shown in FIG. 4F may be executed one or more times, with the result of growing path lengths by one link for each execution. As shown in FIG. 4A,

in some embodiments, three iterations of the process shown in FIG. 4F are used to grow paths by three links. In other embodiments, more or fewer iterations are used.

As shown in FIG. 4G, in order to save the new paths, map phase 528 may include determining if there are any more (node identifier, path set) records in the output file at step 530. If so, then at step 532, for each path in the path set, the path tail identifier may be mapped to the path. At step 534, for each path in the path set, the path head identifier may be mapped to the path.

If there are no more records at step 530, then, in reduce phase 536, a determination may be made at step 538 that there are more node identifiers with mapped paths to process. If so, then at step 540, if the path tail identifier equals the node identifier, then that path may be added to the node's "out" bucket for the path head identifier. At step 542, if the path head identifier equals the node identifier, then that path may be added to the node's "in" bucket for the path tail identifier. At step 544, the node may be saved. If there are no more records to process at step 538, the process may stop at step 546.

As shown in FIG. 4H, in order to join paths that go through changed nodes, map phase 548 may include determining if there are any more (node identifier, path set) records in the output file at step 550. If so, then at step 552, all paths in "in" buckets may be joined with all paths in "out" buckets. At step 554, for each qualified joined path with length less than or equal to three (or the number of iterations of the process shown in FIG. 4F), the path tail identifier may be mapped to the path, and the path head identifier may also be mapped to the path.

If there are no more records at step 550, then, in reduce phase 556, a determination may be made at step 558 that there are more node identifiers with mapped paths to process. If so, then at step 560, if the path tail identifier equals the node identifier, then that path may be added to the node's "out" bucket for the path head identifier. At step 562, if the path head identifier equals the node identifier, then that path may be added to the node's "in" bucket for the path tail identifier. At step 564, the node may be saved. If there are no more records to process at step 558, the process may stop at step 566.

FIG. 5 shows illustrative process 580 for supporting a user query for all paths from a first node to a target node. For example, a first node (representing, for example, a first individual or entity) may wish to know how connected the first node is to some second node (representing, for example, a second individual or entity) in the

network community. In the context of trust described above (and where the user connectivity values represent, for example, at least partially subjective user trust values), this query may return an indication of how much the first node may trust the second node. In general, the more paths connecting the two nodes may yield a greater (or lesser if, for example, adverse ratings are used) network connectivity value (or network trust amount).

At step 582, for each source node "out" bucket, the corresponding "in" bucket of target nodes may be located. For example, column 320 of node table 312 (both of FIG. 3B) may be accessed at step 582. Paths from the source node's "out" bucket may then be joined with paths in the target node's "in" bucket at step 584. Joined paths with paths in the source node's "out" bucket may then be returned for the target node's identifier. Process 580 may stop at step 588.

Having returned all paths between the source and target node (of length less than or equal to three, or any other suitable value depending on the number of iterations of the process shown in FIG. 4F), a network connectivity value may be computed. The path weights assigned to the paths returned at step 586 may then be summed. The path weights may be normalized by dividing each path weight by the computed sum of the path weights. A network connectivity value may then be computed. For example, each path's user connectivity value may be multiplied by its normalized path weight. The network connectivity value may then be computed in some embodiments in accordance with:

$$t_{network} = \sum t_{path} \times w_{path} \tag{7}$$

where t_{path} is the user connectivity value for a path (given in accordance with equation (5)) and w_{path} is the normalized weight for that path. The network connectivity value may then be held, output by processing circuitry of application server 106, and/or stored on data store 110 (FIG. 1). In addition, a decision-making algorithm may access the network connectivity value in order to make automatic decisions (e.g., automatic network-based decisions, such as authentication or identity requests) on behalf of the user. Network connectivity values may additionally or alternatively be outputted to external systems and processes located at third-parties. The external systems and processes may be configured to automatically initiate a transaction (or take some particular course of action) based, at least in part, on the received network connectivity values. For example, some locales or organizations may require identity references in order to apply for a document (e.g., a passport, driver's license, group or club membership

card, etc.). The identity reference or references may vouch that an individual actually exists and/or is the individual the applicant is claiming to be. Network connectivity values may be queried by the document issuer (e.g., a local government agency, such as the Department of Motor Vehicles or a private organization) and used as one (or the sole) metric in order to verify the identity of the applicant, the identity of an identity reference, or both. In some embodiments, network connectivity values may be used as an added assurance of the identity of an applicant or reference in conjunction with more traditional forms of identification (e.g., document verification and knowledge-based identity techniques). If the document issuer (or some other party trusted by the document issuer) has a set of strong paths from the applicant or reference, this may indicate a higher degree of confidence in the identity of the applicant or reference. Such an indication may be outputted to the third-party system or process.

As another example, credit-granting decisions may be made by third parties based, at least in part, on network connectivity values. One or more queries for a network connectivity value may be automatically executed by the credit-granting institution (e.g., a bank, private financial institution, department store) as part of the credit application process. For example, a query for a network connectivity value between the applicant and the credit-granting institution itself (or its directors, board members, etc.) and between the applicant and one or more trusted nodes may be automatically executed as part of the credit application process. The one or more network connectivity values returned to the credit-granting institution may then be used as an input to a proprietary credit-granting decision algorithm. In this way, a credit-granting decision may be based on a more traditional component (e.g., occupation, income, repayment delinquencies, and credit score) and a network connectivity component. Each component may be assigned a weight and a weighted sum or weighted average may be computed. The weighted sum or average may then be used directly to make an automatic credit-granting decision for the applicant. The weights assigned to each component of the weighted sum or average may be based on such factors as the applicant's credit history with the financial institution, the amount of credit requested, the degree of confidence in the trusted nodes, any other suitable factor, or any combination of the foregoing factors. In some embodiments, the credit-granting or other decisions made by third parties may be made based entirely on network connectivity values.

In practice, one or more steps shown in process 580 may be combined with other steps, performed in any suitable order, performed in parallel (e.g., simultaneously or

substantially simultaneously), or removed. In addition, as described above, various threshold functions may be used in order to reduce computational complexity. For example, one or more threshold functions defining the maximum and/or minimum number of links to traverse may be defined. Paths containing more than the maximum number of links or less than the minimum number of links specified by the threshold function(s) may not be considered in the network connectivity determination. In addition, various maximum and/or minimum threshold functions relating to link and path weights may be defined. Links or paths above a maximum threshold weight or below a minimum threshold weight specified by the threshold function(s) may not be considered in the network connectivity determination.

Although process 580 describes a single user query for all paths from a first node to a target node, in actual implementations groups of nodes may initiate a single query for all the paths from each node in the group to a particular target node. For example, multiple members of a network community may all initiate a group query to a target node. Process 580 may return an individual network connectivity value for each querying node in the group or a single composite network connectivity value taking into account all the nodes in the querying group. For example, the individual network connectivity values may be averaged to form a composite value or some weighted average may be used. The weights assigned to each individual network connectivity value may be based on seniority in the community (e.g., how long each node has been a member in the community or other indicators of stability or seniority), rank, or social stature. In addition, in some embodiments, a user may initiate a request for network connectivity values for multiple target nodes in a single query. For example, node n_1 may wish to determine network connectivity values between it and multiple other nodes. For example, the multiple other nodes may represent several candidates for initiating a particular transaction with node n_1. By querying for all the network connectivity values in a single query, the computations may be distributed in a parallel fashion to multiple cores so that some or all of the results are computed substantially simultaneously.

In addition, queries may be initiated in a number of ways. For example, a user (represented by a source node) may identify another user (represented by a target node) in order to automatically initiate process 580. A user may identify the target node in any suitable way, for example, by selecting the target node from a visual display, graph, or tree, by inputting or selecting a username, handle, network address, email address, telephone number, geographic coordinates, or unique identifier associated with

the target node, or by speaking a predetermined command (e.g., "query node 1" or "query node group 1, 5, 9" where 1, 5, and 9 represent unique node identifiers). After an identification of the target node or nodes is received, process 520 may be automatically executed. The results of the process (e.g., the individual or composite network connectivity values) may then be automatically sent to one or more third-party services or processes as described above.

In an embodiment, a user may utilize access application 102 to generate a user query that is sent to access application server 106 over communications network 104 (*see also*, FIG. 1) and automatically initiate process 580. For example, a user may access an Apple iOS, Android, or Webs application or any suitable application for use in accessing application 106 over communications network 104. The application may display a searchable list of relationship data related to that user (e.g., "friend" or "follower" data) from one or more of Face book, MySpace, open Social, Friendster, Bebop, hi5, Rout, PerfSpot, Yahoo! 360, LinkedIn, Twitter, Google Buzz, Really Simple Syndication readers or any other social networking website, information service, affiliation database, or affinity database. In some embodiments, a user may search for relationship data that is not readily listed – i.e., search Face book, Twitter, or any suitable database of information for target nodes that are not displayed in the searchable list of relationship data. A user may select a target node as described above (e.g., select an item from a list of usernames representing a "friend" or "follower") to request a measure of how connected the user is to the target node. Using the processes described with respect to FIGs. 3A-C and 4A-H, this query may return an indication of how much the user may trust the target node. The returned indication may be displayed to the user using any suitable indicator. In some embodiments, indicator may be a percentage that indicates how trustworthy the target node is to the user.

In some embodiments, a user may utilize access application 102 to provide manual assignments of at least partially subjective indications of how trustworthy the target node is. For example, the user may specify that he or she trusts a selected target node (e.g., a selected "friend" or "follower") to a particular degree. The particular degree may be in the form of a percentage that represents the user's perception of how trustworthy the target node is. The user may provide this indication before, after, or during process 580 described above. The indication provided by the user (e.g., the at least partially subjective indications of trustworthiness) may then be automatically sent to one or more third-party services or processes as described above. In some embodiments,

the indications provided by the user may cause a node and/or link to change in a network community. This change may cause a determination to be made that at least one node and/or link has changed in the network community, which in turn triggers various processes as described with respect to FIGs. 3A-C and 4A-4H.

In some embodiments, a user may utilize access application 102 to interact with or explore a network community. For example, a user may be presented with an interactive visualization that includes one or more implicit or explicit representations of connectivity values between the user and other individuals and/or entities within the network community. This interactive visualization may allow the user to better understand what other individuals and/or entities they may trust within a network community, and/or may encourage and/or discourage particular interactions within a user's associated network community or communities.

In some embodiments, a path counting approach may be used in addition to or in place of the weighted link approach described above. Processing circuitry (e.g., of application server 106 (FIG. 1)) may be configured to count the number of paths between a first node n_1 and a second node n_2 within a network community. A connectivity rating R_{n1n2} may then be assigned to the nodes. The assigned connectivity rating may be proportional to the number of paths, or relationships, connecting the two nodes. A path with one or more intermediate nodes between the first node n_1 and the second node n_2 may be scaled by an appropriate number (e.g., the number of intermediate nodes) and this scaled number may be used to calculate the connectivity rating.

FIG. 6 shows illustrative process 600 for logging into the connectivity system. At step 602, a user request to login may be received. For example, application server 106 (FIG. 1) may receive a login attempt from access application 102 (FIG. 1). At step 604, one or more external login mechanisms may be accessed. For example, the user may be redirected to a login mechanism associated with an email or social networking service, like Facebook, Hotmail, Gmail, or the like. After the external login mechanism is accessed, the user may be redirected to the application server at step 606. For example, the user may be redirected back to the page associated with application server 106 (FIG. 1). At step 608, a determination is made whether the external login mechanism was completed successfully. For example, the external login mechanism may return a token, timestamp, username, handle, email address, unique identifier, cryptographic hash (e.g., of a username or unique identifier associated with the user), any other identity information, or any combination of the foregoing in the URL to the redirected application

server page. The information may be verified using any known authentication protocol. If the external login mechanism was successful, then at step 610 application server 106 (FIG. 1) may lookup a corresponding sign-in profile in order to identify the user. For example, the provider of the external login mechanism may pass its name as a string along with a unique identifier to application server 106 (FIG. 1). Application server 106 (FIG. 1) may then look this information up in table 332 (FIG. 3C). If a corresponding sign-in profile record is located, this profile may be used to identify the user.

In practice, one or more steps shown in process 600 may be combined with other steps, performed in any suitable order, performed in parallel (e.g., simultaneously or substantially simultaneously), or removed.

FIG. 7 shows illustrative process 700 for facilitating a financial transaction. Although the described embodiments sometimes refer to a loan or donation financial application or transaction, the present invention may be used to facilitate any type of financial transaction. For example, financial transactions may include purchases, sales, donations of cash, donations of property, loans, mortgages, liens, credit applications, credit-granting or -denying decisions, insurance underwriting, hiring decisions, recruiting decisions, employee assignment decisions, or any other type of financial transaction involving the change in status of finances or change in legal status between two or more individuals, nodes, users, institutions, organizations, pieces of property, tangible assets, or things. At step 702, a first user may initiate a new financial transaction. For example, the user may access a loan or donation application at step 702. The application may include a series of electronic forms (e.g., web pages) to be filled out by the user and submitted for approval. At step 704, a determination is made whether the transaction is a public or private transaction. In some embodiments, users may designate specific transactions as public or private. In some embodiments, the financial application itself may also determine whether a transaction is public or private. For example, charitable contributions may always be designated as public transactions whereas personal loans may always be designated as private transactions. By way of further example, hiring decisions are often public whereas insurance underwriting and sales credit decisions are often private.

At step 706, a publication group is determined. For example, all users or nodes meeting or exceeding a minimum threshold connectivity value and/or not exceeding a maximum threshold connectivity value with the first user may be added to the publication group. As another example, all nodes or users meeting or exceeding some

minimum threshold path weight and/or not exceeding a maximum threshold path weight to the first user may be added to the publication group. In some embodiments, the first user is given an opportunity to select the publication group or groups to which the user wants transaction information to be published. For example, the user may specify custom connectivity value maximum/minimum thresholds, custom path weight maximum/minimum thresholds, or both. This threshold value (or values) may then be used to determine the appropriate publication group. The user may also be given an opportunity to view a listing of publication group members, add additional members, and remove existing members, if desired.

In some embodiments, publication groups may be further refined using additional information known about other nodes or users in the network. For example, a first user may initiate a donation transaction for a wildlife refuge. In determining the appropriate publication group, nodes with high connectivity values with a known wildlife affinity or support group may be automatically added to the publication group, whether or not they meet the path weight or connectivity threshold values. Application server 106 (FIG. 1) may automatically compare attribute flags and other metadata associated with the financial application (for example, stored in the description field in financial application table 344 (FIG. 3C)) with attributes known about other nodes or users in the network and use the results of this comparison in adding additional members to, or removing otherwise qualifying members from, publication groups. For example, "LIKE" and "DISLIKE" flags (as described above with regard to FIG. 3C) may be read from financial application table 344 (FIG. 3C) and used to refine publication group membership using information other than (or in addition to) connectivity values and path weights. Users matching a "LIKE" flag may be automatically added to the publication group whether or not they meet one or more threshold values in some embodiments. In other embodiments, users or nodes must both match any defined "LIKE" flag and meet applicable threshold values in order to be added to a publication group. Similarly, users matching a "DISLIKE" flag may be automatically removed from the publication group even if they meet one or more threshold values in some embodiments.

At step 708, transaction information may be published to the selected publication group or groups. Publication may take a variety of forms, including email messages, text messages, voicemails, listings on a homepage, listings on a profile page, listings on a shared-access or community page, postings to a discussion forum, notification messages, other suitable notifications, or any combination of the foregoing.

The type of notifications may be dependent on the active sign-in profile, in some embodiments. For example, if the active sign-in profile is for an email account provider, at least some of the notifications may take the form of email messages. If the active sign-in profile is for a social networking service provider, at least some of the notifications may take the form of provider notifications, wall postings, profile page postings, or the like.

At step 710, a determination is made whether a second user (e.g., a member of the publication group) has accessed the same financial application. In some embodiments, the second user may access the same financial application directly from the publication. For example, a published notification may include a link (e.g., hyperlink) to the financial application. The second user may directly access the financial application by activating the link (e.g., by clicking or selecting the link). In some embodiments, at least some of the information from the first user's financial transaction is automatically carried over to the second user's transaction, allowing the second user to efficiently execute a partly or wholly-identical transaction as the first user. For example, if the transaction is a donation, the donation amount (or more generically the principal) from the first user's transaction may be pre-populated in the electronic forms associated with the second user's transaction. In that way, users may be encouraged to donate (or borrow) the same amount as the first user. In some embodiments, users are not allowed to change pre-populated information (e.g., so as to encourage a minimum level of charitable giving). In other embodiments, pre-populated information may be changed by the user. If at step 710 the second user does access the same financial application, a new financial transaction may be processed on behalf of the second user at step 712. If applicable, a repayment schedule may also be automatically generated at step 714. For example table 346 may be automatically populated, if the financial transaction is a loan.

In processing financial transactions, connectivity values may be used to determine eligibility of the lender, borrower, or both (in the case of a loan transaction). For example, eligible borrowers may need to meet a threshold connectivity value with the lender, the lending institution, one or more officers or directors of the lending institution, or any combination of the foregoing. In addition, as described above, third-party processes may make automatic transaction decisions based, at least in part, the connectivity values. For example, in some embodiments, at least three threshold network connectivity values may be defined, N_1, N_2, and N_3, where $N_1 > N_2 > N_3$. Potential borrowers may be automatically approved for the financial transaction if they meet the

threshold network connectivity value N_1. If borrowers fail to meet the threshold network connectivity value N_1, but meet threshold network connectivity value N_2, then a composite score based on the actual network connectivity value and a third-party ratings agency (such as a credit ratings bureau score) may be used to determine the approval status for the financial transaction. If potential borrowers do not meet threshold network connectivity value N_2, but meet threshold network connectivity value N_3, these potential borrowers may be referred for manual processing. If potential borrowers do not meet threshold network connectivity value N_3, these potential borrowers may be automatically denied participation in the financial transaction. The values of N_1, N_2, and N_3 may be specified by the lending institution, an officer of the lending institution, or the financial application.

In practice, one or more steps shown in process 700 may be combined with other steps, performed in any suitable order, performed in parallel (e.g., simultaneously or substantially simultaneously), or removed. In some embodiments, process 700 may be used to facilitate other transactions, such as identity assessments, security risk assessments, or any other transaction that can take advantage of user connectivity values.

Each equation presented above should be construed as a class of equations of a similar kind, with the actual equation presented being one representative example of the class. For example, the equations presented above include all mathematically equivalent versions of those equations, reductions, simplifications, normalizations, and other equations of the same degree.

The above described embodiments of the invention are presented for purposes of illustration and not of limitation. The following claims give additional embodiments of the present invention.

What is Claimed is

1. A method for facilitating financial transactions comprising:

transmitting, to a first user and potential members of a publication group, software for communications relating to trust-based transactions;

receiving with a server, a request from the first user's software to initiate a first trust-based transaction;

automatically determining a publication group to publish at least some information relating to the transaction, wherein determining the publication group comprises:

accessing, with at least one server, information in a datastore relating to a network community;

identifying, with the at least one server, paths in the network community from the first user to at least one potential member of the publication group; and

identifying sub-processes required to calculate a connectivity value between the first user and the at least one potential member, wherein identifying the sub-processes comprises:

identifying a plurality of links in the identified paths;

for one or more identified links, accessing a data structure to identify nodes connected to the identified link; and

for each identified node, creating an indication of a sub-process, wherein the sub-process comprises calculating an out-link weight for one or more out-links of the identified node;

distributing the indications of the sub-processes to a plurality of processors arranged in a parallel computational framework;

receiving, from the plurality of processors, calculated out-link weights for the identified nodes;

calculating the connectivity value based on the calculated out-link weights; and

adding the at least one potential member to the publication group based on the calculated connectivity value;

publishing the information relating to the transaction to the publication group; and

transmitting, to the publication group's software, information related to the transaction with which information a member of the publication group may initiate a second trust-based transaction.

2. The method of claim 1 further comprising determining whether the first transaction is public, and

performing the steps of automatically determining a publication group and publishing the information, only if the first transaction is public.

3. The method of claim 1 wherein publishing the information comprises publishing a link to an application related to the first transaction.

4. The method of claim 3, wherein the published link allows the member of the publication group to access the application by activating the link.

5. The method of claim 1, further comprising pre-populating at least some information in the second transaction with information from the first transaction.

6. The method of claim 1 wherein determining the publication group comprises comparing the determined connectivity value to a threshold connectivity value.

7. The method of claim 1 wherein determining the publication group comprises determining a threshold path weight value.

8. The method of claim 1 wherein determining the publication group comprises accessing attribute flag information associated with the first transaction, wherein the attribute flag information is indicative of at least other users who may be interested in the first transaction.

9. The method of claim 8 wherein the attribute flag information identifies at least one affinity group whose members may be interested in the first transaction.

10. An system for facilitating financial transactions comprising processing circuitry configured to:

receive with a server, a request from the first user's software to initiate a first trust-based transaction;

automatically determine a publication group to publish at least some information relating to the transaction,

wherein determining the publication group comprises:

accessing, with at least one server, information in a datastore relating to a network community;

identifying, with the at least one server, paths in the network community from the first user to at least one potential member of the publication group; and

identifying sub-processes required to calculate a connectivity value between the first user and the at least one potential member,

wherein identifying the sub-processes comprises:

identifying a plurality of links in the identified paths;

for one or more identified links, accessing a data structure to identify nodes connected to the identified link; and

for each identified node, creating an indication of a sub-process,

wherein the sub-process comprises calculating an out-link weight for one or more out-links of the identified node;

distributing the indications of the sub-processes to a plurality of processors arranged in a parallel computational framework;

receiving, from the plurality of processors, calculated out-link weights for the identified nodes;

calculating the connectivity value based on the calculated out-link weights; and

adding the at least one potential member to the publication group based on the calculated connectivity value;

publish the information relating to the transaction to the publication group; and

transmit, to the publication group's software, information related to the transaction with which information a member of the publication group may initiate a second trust-based transaction.

11. The system of claim 10 wherein the processing circuitry is further configured to determine if the first transaction is public, wherein the processing circuitry is configured to at least determine the publication group and publish the information relating to the first transaction to the publication group in response to determining that the first transaction is public.

12. The system of claim 10 wherein the processing circuitry is configured to publish the information by publishing a link to an application related to the first transaction.

13. The system of claim 12, wherein the published link allows the member of the publication group to access the application directly from activating the link.

14. The system of claim 10, wherein the processing circuitry is further configured to pre-populate at least some information in the second transaction with information from the first transaction.

15. The system of claim 14, wherein the processing circuitry is configured to pre-populate at least some information by pre-populating at least a principal amount.

16. The system of claim 10 wherein the processing circuitry is configured to determine the publication group by comparing the determined connectivity value to a threshold connectivity value.

17. The system of claim 10 wherein the processing circuitry is configured to determine the publication group by determining a threshold path weight value.

18. The system of claim 10 wherein the processing circuitry is configured to determine the publication group by accessing attribute flag information associated with an application related to the first transaction, wherein the attribute flag information is indicative of other users who may be interested in the application.

19. A method for facilitating financial transactions comprising:

transmitting, to a first user and potential members of a publication group, software for communications relating to trust-based transactions;

receiving with a server, a request from the first user's software to initiate a first trust-based transaction;

determining whether the first transaction is public;

if the first transaction is public, automatically determining a publication group to publish at least some information relating to the transaction, wherein determining the publication group comprises:

accessing, with at least one server, information in a datastore relating to a network community;

identifying, with the at least one server, paths in the network community from the first user to at least one potential member of the publication group; and

identifying sub-processes required to calculate a connectivity value between the first user and the at least one potential member, wherein identifying the sub-processes comprises:

identifying a plurality of links in the identified paths;

for one or more identified links, accessing a data structure to identify nodes connected to the identified link; and

for each identified node, creating an indication of a sub-process, wherein the sub-process comprises calculating an out-link weight for one or more out-links of the identified node;

distributing the indications of the sub-processes to a plurality of processors arranged in a parallel computational framework;

receiving, from the plurality of processors, calculated out-link weights for the identified nodes;

calculating the connectivity value based on the calculated out-link weights; and

adding the at least one potential member to the publication group based on the calculated connectivity value.

20. The method of claim 19 further comprising:

if the first transaction is public:

publishing the information relating to the transaction to the publication group; and

transmitting, to the publication group's software, information related to the transaction with which information a member of the publication group may initiate a second trust-based transaction.

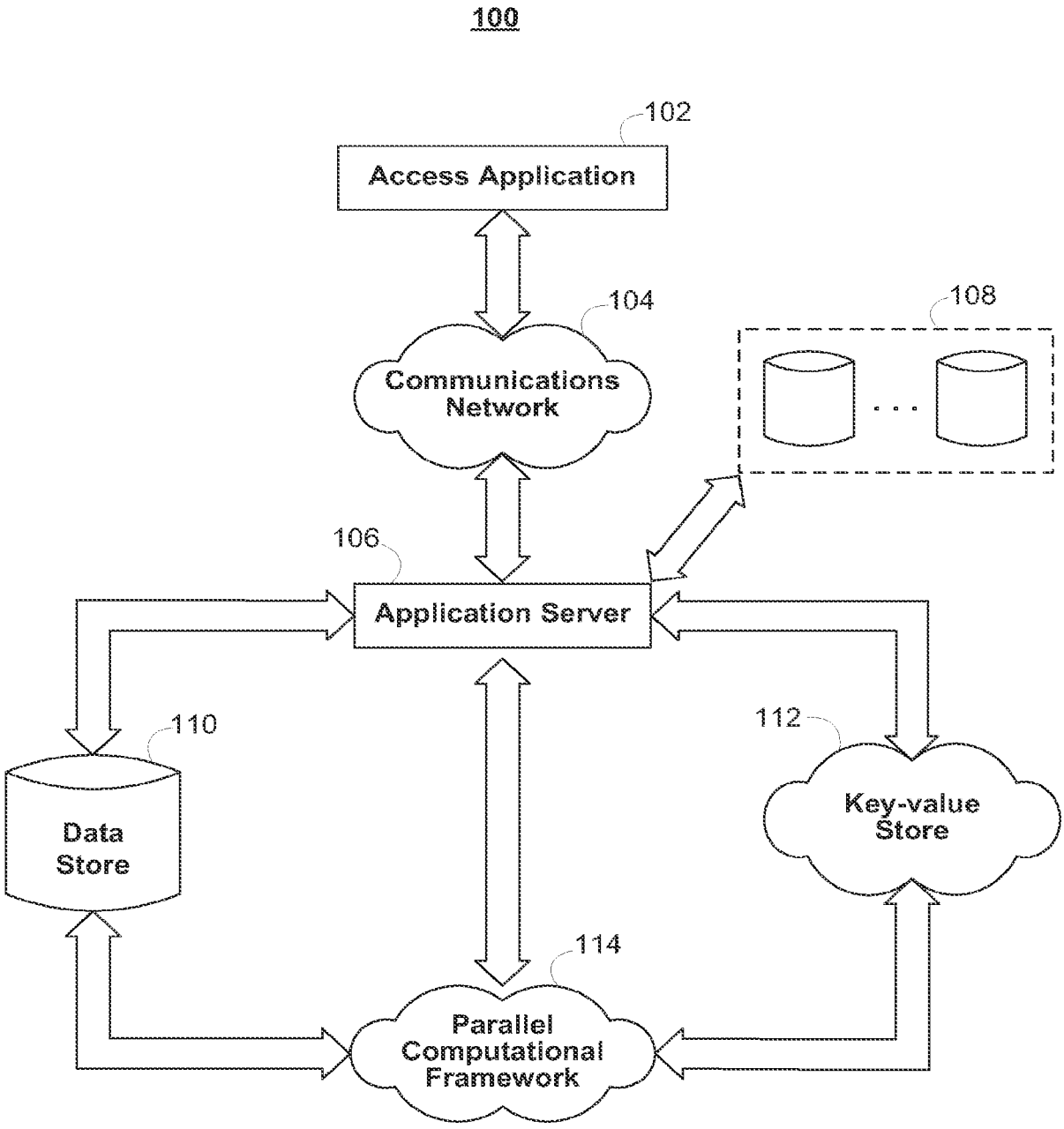

100
102
Access Application
104
Communications Network
108
106
Application Server
110
Data Store
112
Key-value Store
114
Parallel Computational Framework

FIG. 1

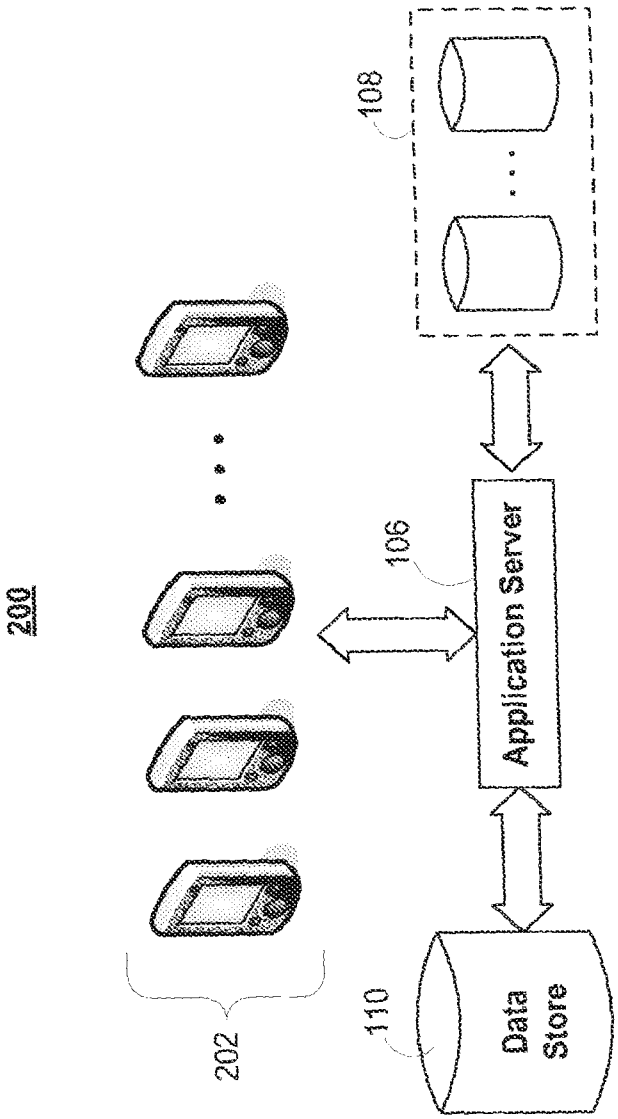


FIG. 2

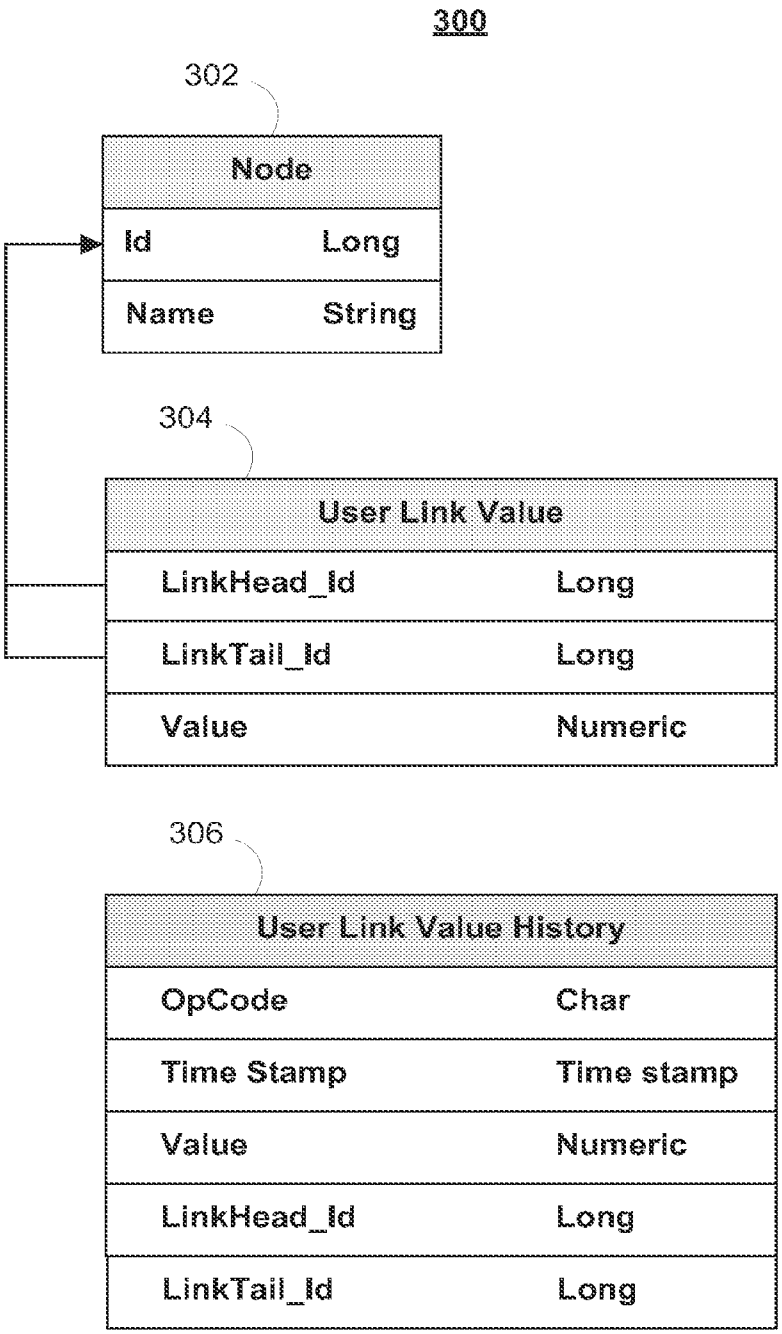

300
302
Node
Id Long
Name String
304
User Link Value
LinkHead_Id Long
LinkTail_Id Long
Value Numeric
306
User Link Value History
OpCode Char
Time Stamp Time stamp
Value Numeric
LinkHead_Id Long
LinkTail_Id Long

FIG. 3A

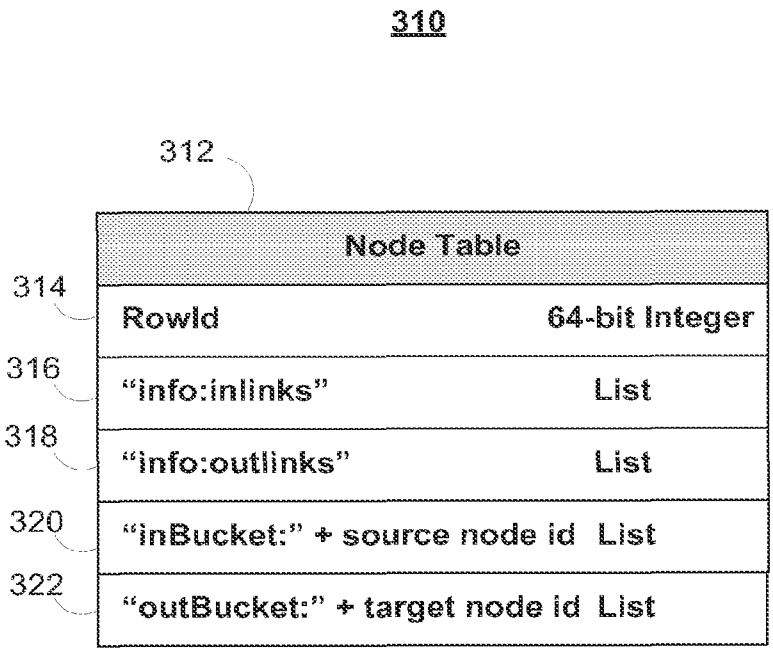

310
312
Node Table
314
RowId
64-bit Integer
316
"info:inlinks"
List
318
"info:outlinks"
List
320
"inBucket:" + source node id
List
322
"outBucket:" + target node id
List

FIG. 3B

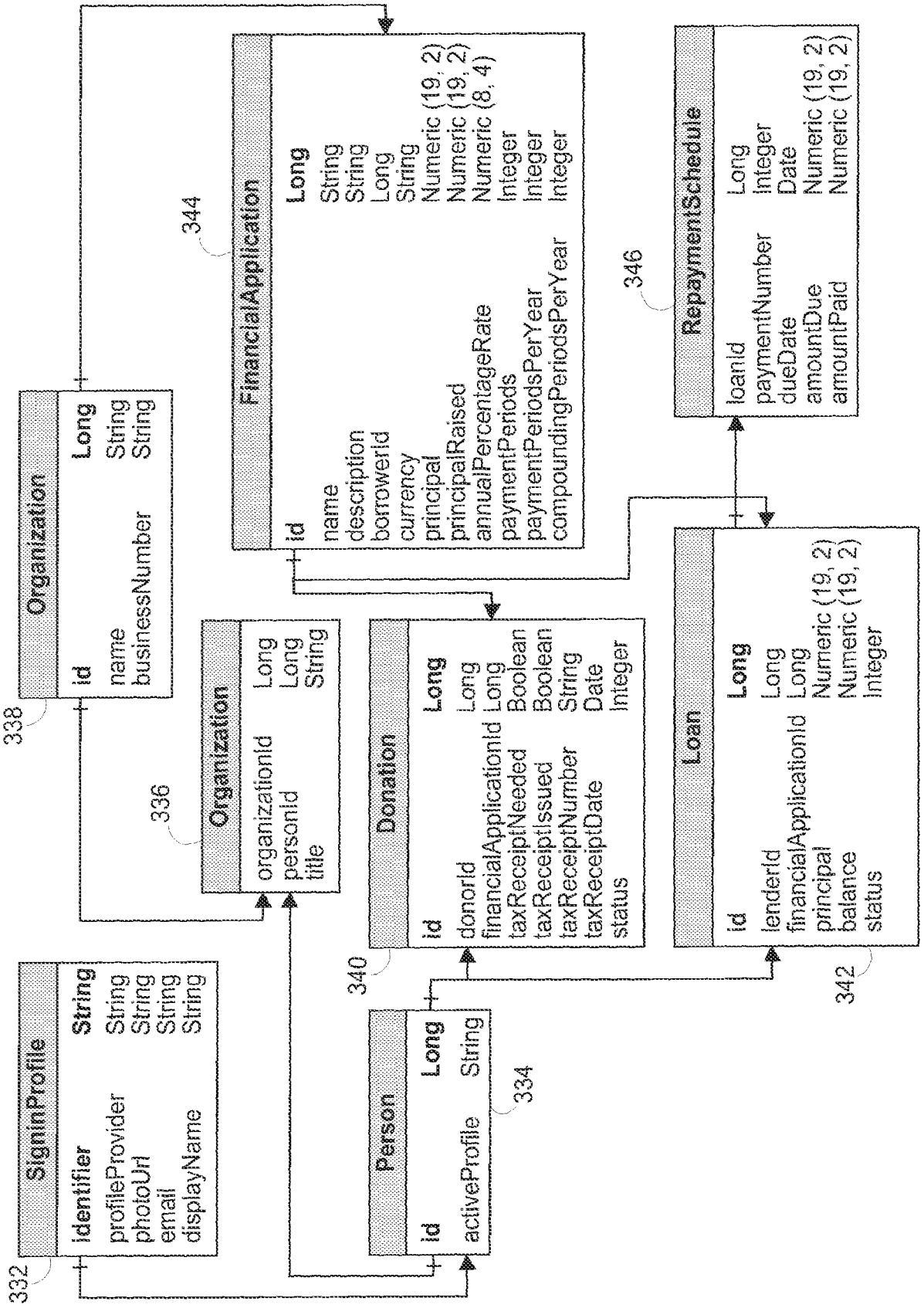

330
SigninProfile
identifier String
profileProvider String
photoUrl String
email String
displayName String
332
Organization
id Long
name String
businessNumber String
338
Organization
organizationId Long
personId Long
title String
336
Person
id Long
activeProfile String
334
FinancialApplication
id Long
name String
description String
borrowerId Long
currency String
principal Numeric (19, 2)
principalRaised Numeric (19, 2)
annualPercentageRate Numeric (8, 4)
paymentPeriods Integer
paymentPeriodsPerYear Integer
compoundingPeriodsPerYear Integer
344
Donation
id Long
donorId Long
financialApplicationId Long
taxReceiptNeeded Boolean
taxReceiptIssued Boolean
taxReceiptNumber String
taxReceiptDate Date
status Integer
340
Loan
id Long
lenderId Long
financialApplicationId Long
principal Numeric (19, 2)
balance Numeric (19, 2)
status Integer
342
RepaymentSchedule
loanId Long
paymentNumber Integer
dueDate Date
amountDue Numeric (19, 2)
amountPaid Numeric (19, 2)
346

FIG. 3C

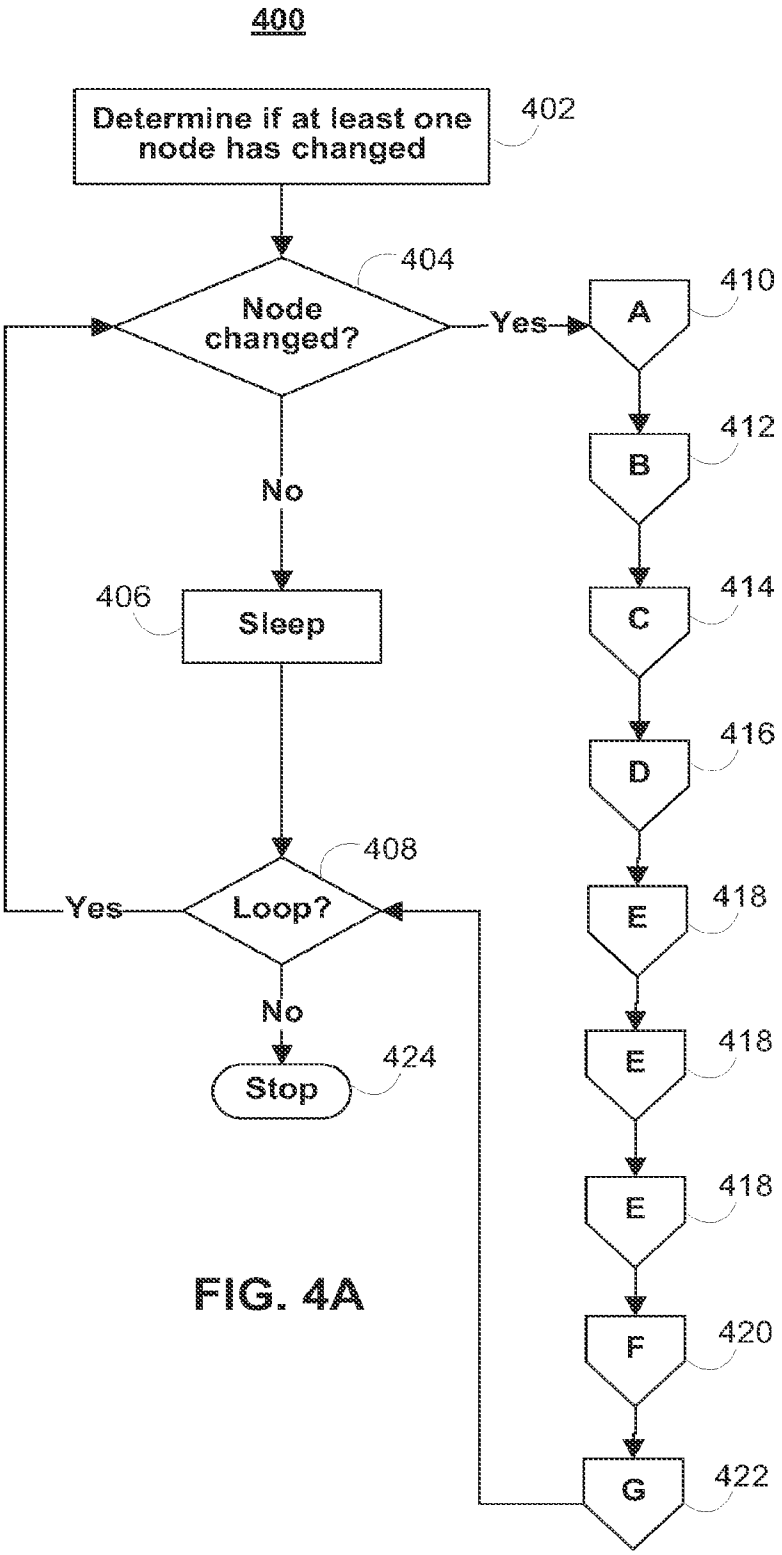

400
Determine if at least one node has changed
402
404
Node changed?
Yes
No
406
Sleep
408
Loop?
Yes
No
424
Stop
410
A
412
B
414
C
416
D
418
E
418
E
418
E
420
F
422
G

FIG. 4A

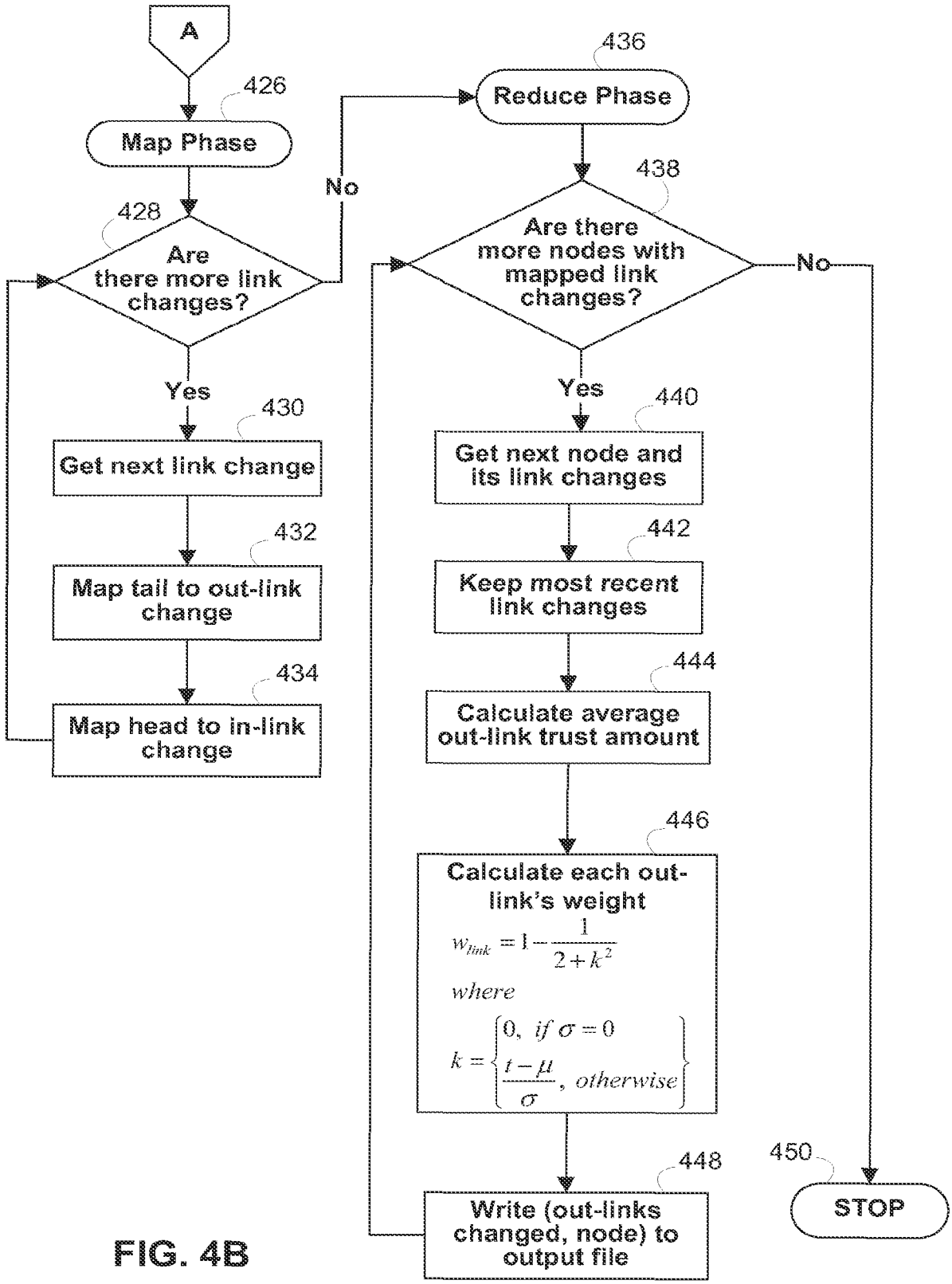

A
426
Map Phase
428
Are there more link changes?
No
Yes
430
Get next link change
432
Map tail to out-link change
434
Map head to in-link change
436
Reduce Phase
438
Are there more nodes with mapped link changes?
No
Yes
440
Get next node and its link changes
442
Keep most recent link changes
444
Calculate average out-link trust amount
446
Calculate each out-link's weight
$w_{link} = 1 - \frac{1}{2+k^2}$
where
$k = \begin{cases} 0, & if\ \sigma = 0 \\ \frac{t-\mu}{\sigma}, & otherwise \end{cases}$
448
Write (out-links changed, node) to output file
450
STOP

FIG. 4B

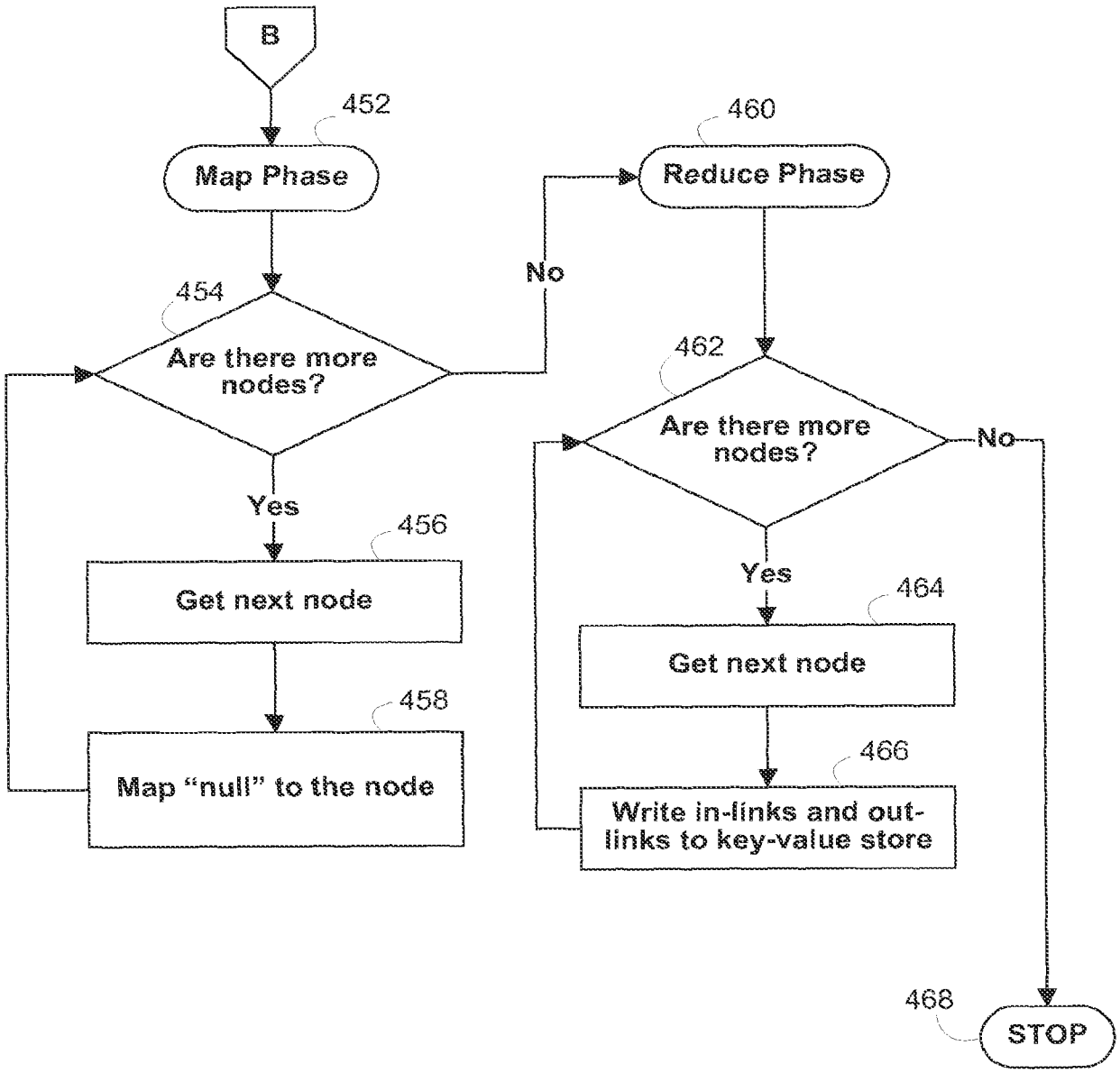

B
452
Map Phase
454
Are there more nodes?
Yes
456
Get next node
458
Map "null" to the node
No
460
Reduce Phase
462
Are there more nodes?
Yes
464
Get next node
466
Write in-links and out-links to key-value store
No
468
STOP

FIG. 4C

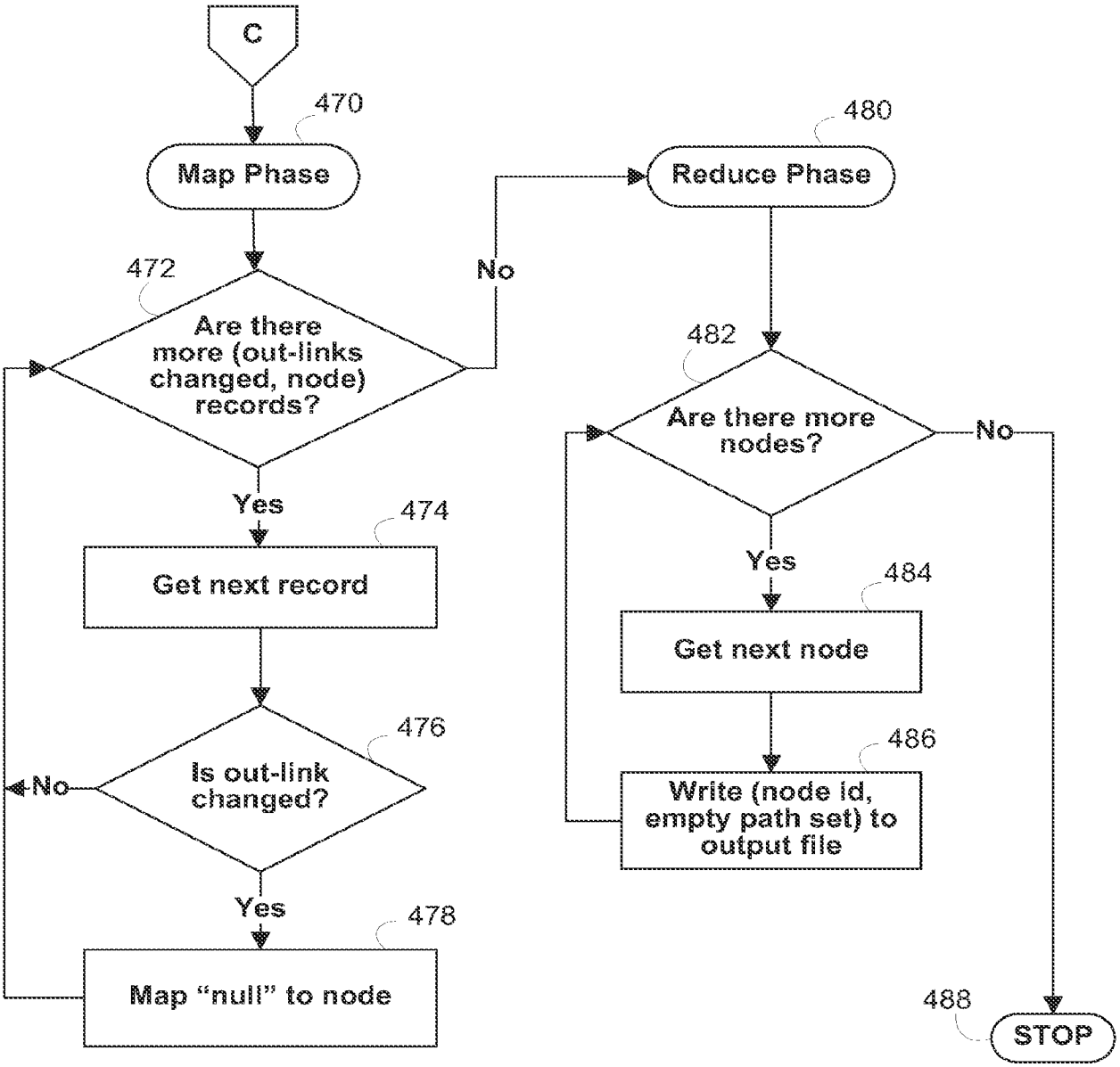

C
470
Map Phase
472
Are there more (out-links changed, node) records?
No
Yes
474
Get next record
476
Is out-link changed?
No
Yes
478
Map "null" to node
480
Reduce Phase
482
Are there more nodes?
No
Yes
484
Get next node
486
Write (node id, empty path set) to output file
488
STOP

FIG. 4D

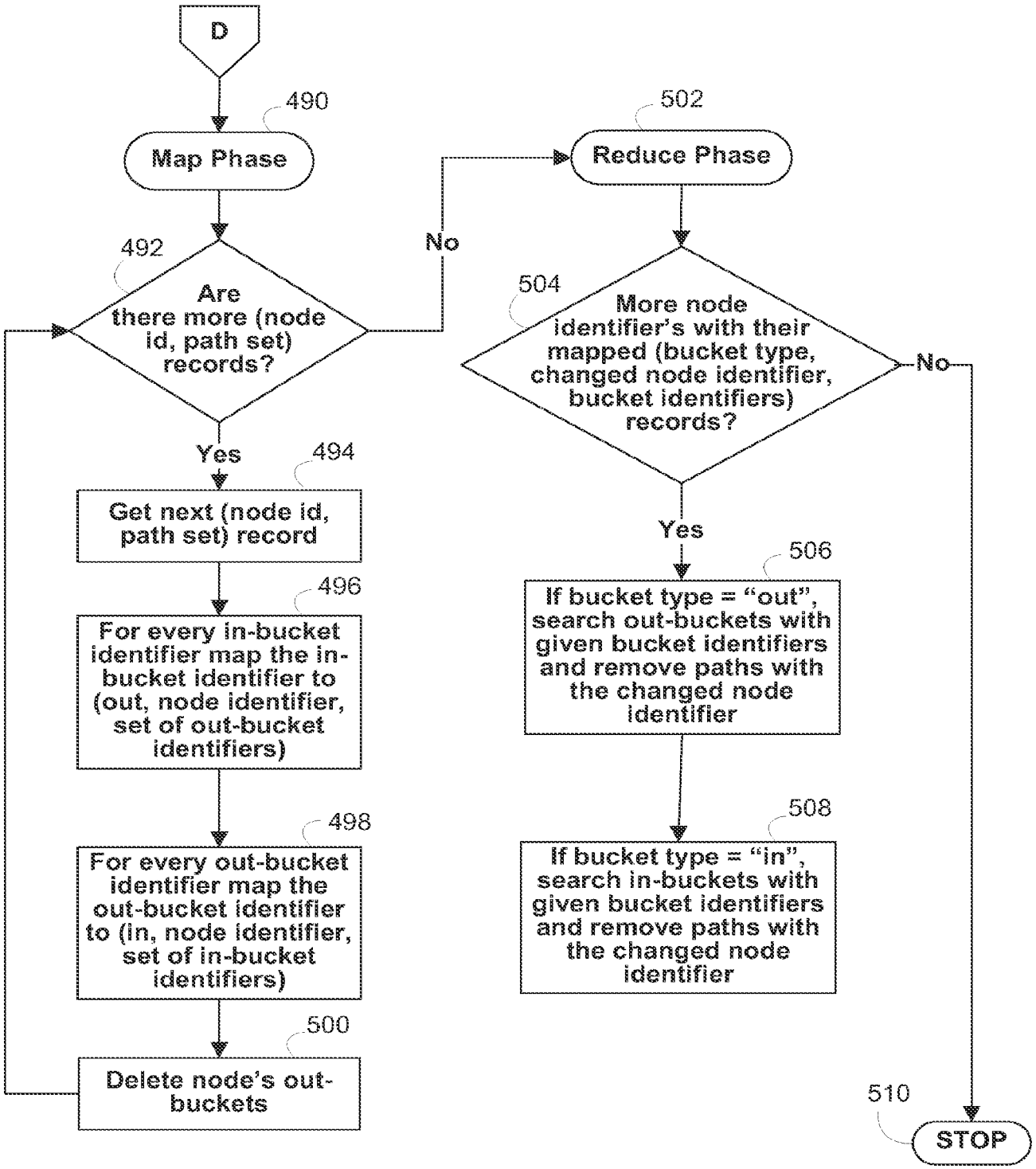

D
490
Map Phase
492
Are there more (node id, path set) records?
No
Yes
494
Get next (node id, path set) record
496
For every in-bucket identifier map the in-bucket identifier to (out, node identifier, set of out-bucket identifiers)
498
For every out-bucket identifier map the out-bucket identifier to (in, node identifier, set of in-bucket identifiers)
500
Delete node's out-buckets
502
Reduce Phase
504
More node identifier's with their mapped (bucket type, changed node identifier, bucket identifiers) records?
No
Yes
506
If bucket type = "out", search out-buckets with given bucket identifiers and remove paths with the changed node identifier
508
If bucket type = "in", search in-buckets with given bucket identifiers and remove paths with the changed node identifier
510
STOP

FIG. 4E

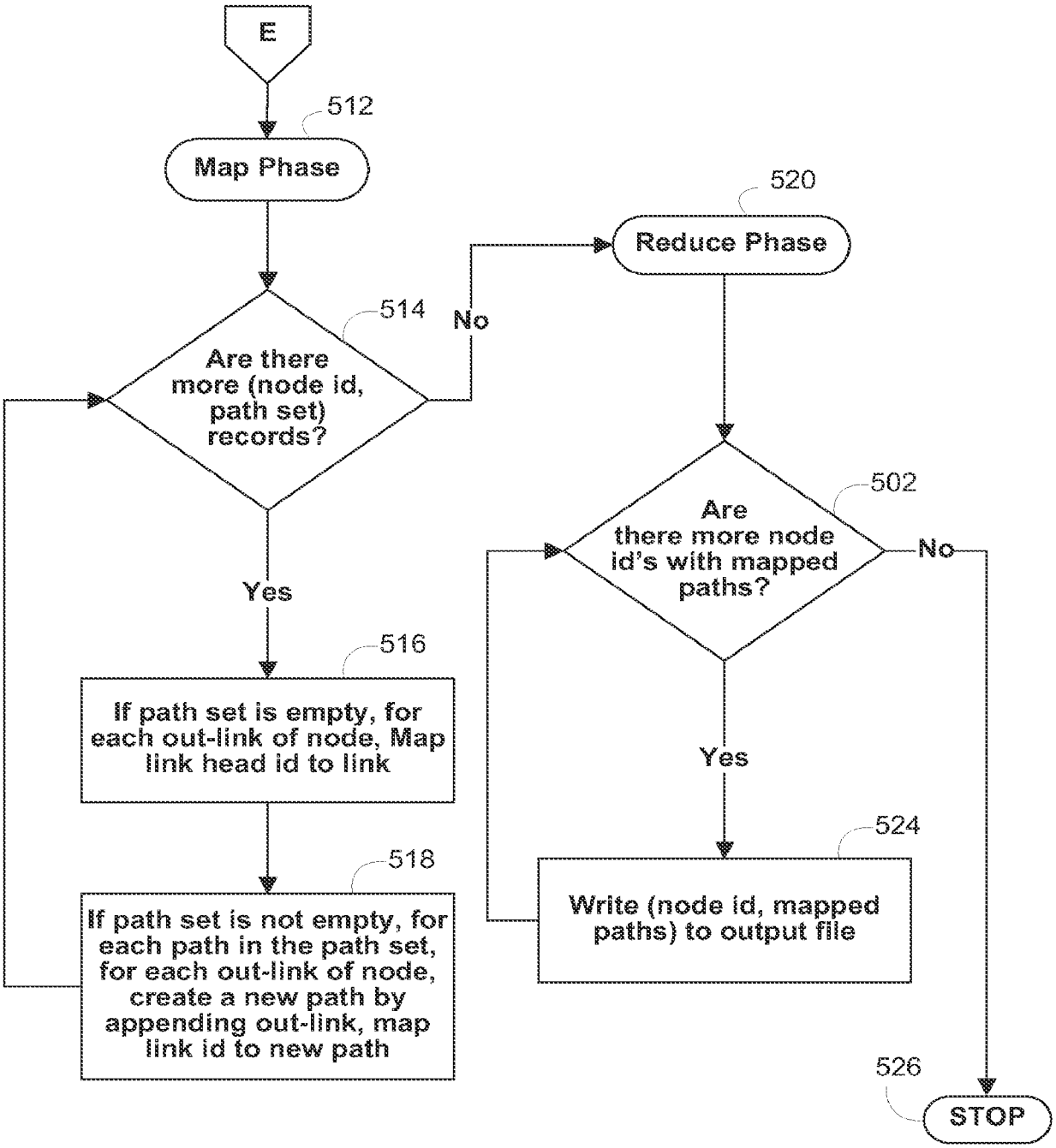

E
512
Map Phase
514
Are there more (node id, path set) records?
No
Yes
516
If path set is empty, for each out-link of node, Map link head id to link
518
If path set is not empty, for each path in the path set, for each out-link of node, create a new path by appending out-link, map link id to new path
520
Reduce Phase
502
Are there more node id's with mapped paths?
No
Yes
524
Write (node id, mapped paths) to output file
526
STOP

FIG. 4F

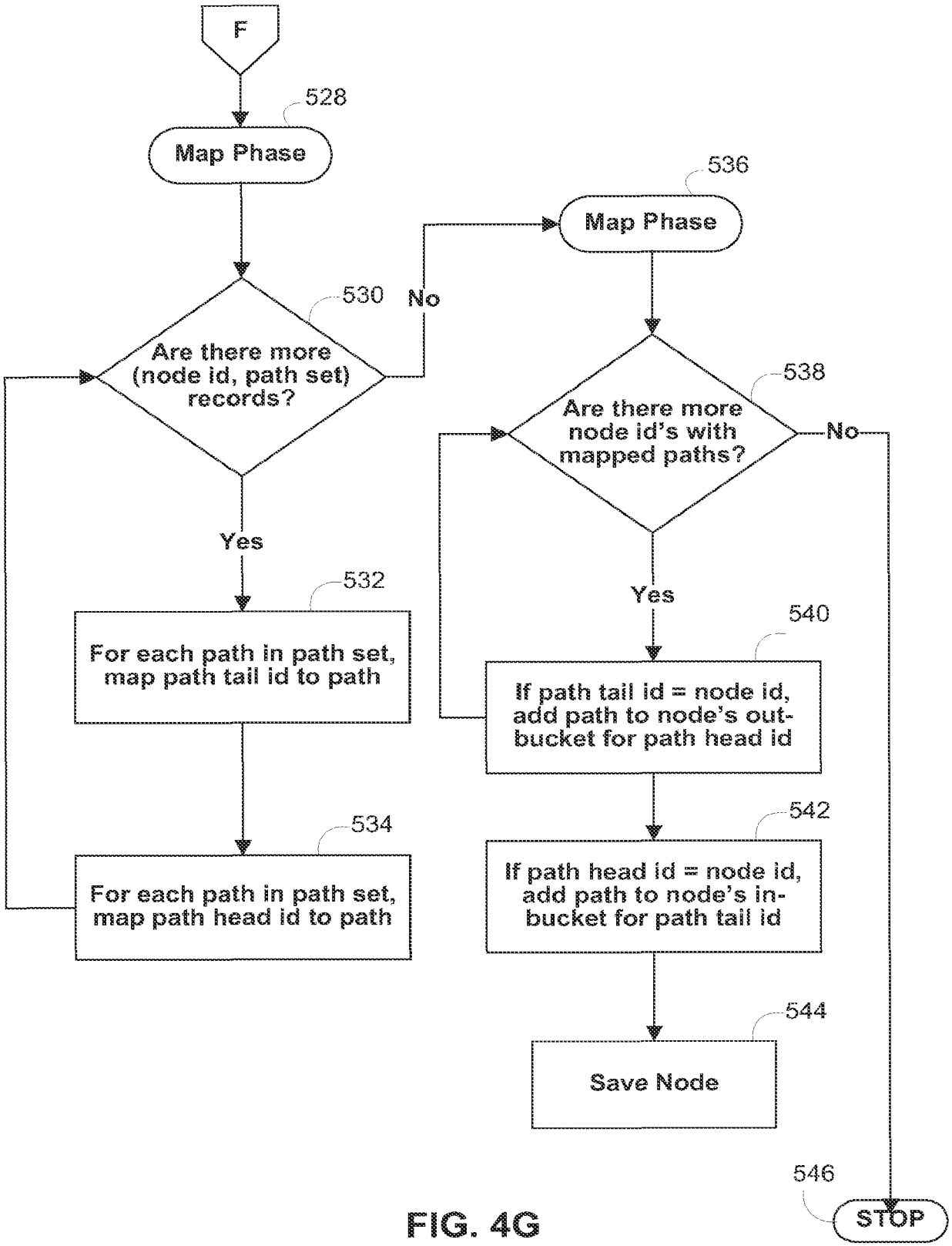

F
528
Map Phase
530
Are there more (node id, path set) records?
No
Yes
532
For each path in path set, map path tail id to path
534
For each path in path set, map path head id to path
536
Map Phase
538
Are there more node id's with mapped paths?
No
Yes
540
If path tail id = node id, add path to node's out-bucket for path head id
542
If path head id = node id, add path to node's in-bucket for path tail id
544
Save Node
546
STOP

FIG. 4G

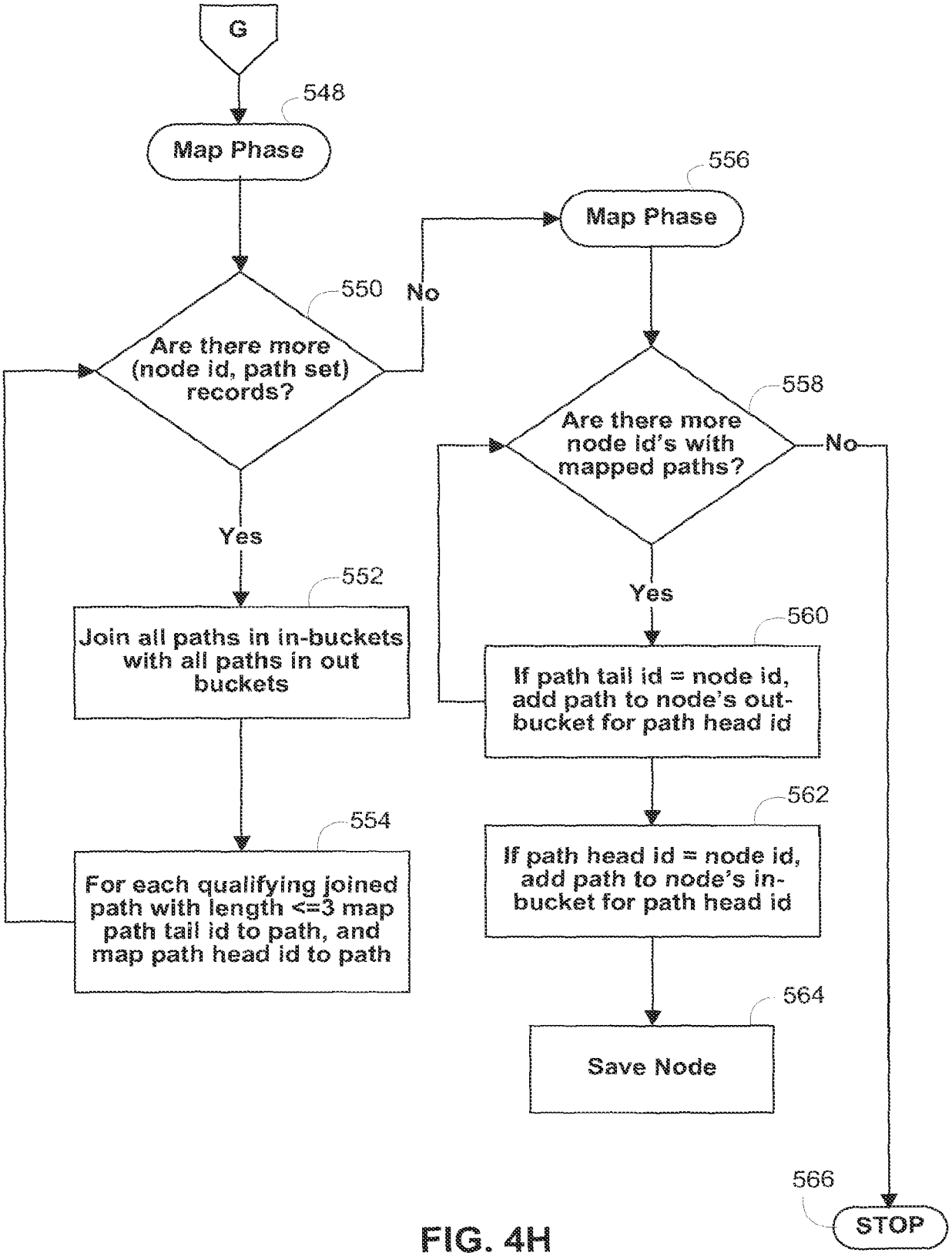

G
548
Map Phase
550
Are there more (node id, path set) records?
No
Yes
552
Join all paths in in-buckets with all paths in out buckets
554
For each qualifying joined path with length <=3 map path tail id to path, and map path head id to path
556
Map Phase
558
Are there more node id's with mapped paths?
No
Yes
560
If path tail id = node id, add path to node's out-bucket for path head id
562
If path head id = node id, add path to node's in-bucket for path head id
564
Save Node
566
STOP

FIG. 4H

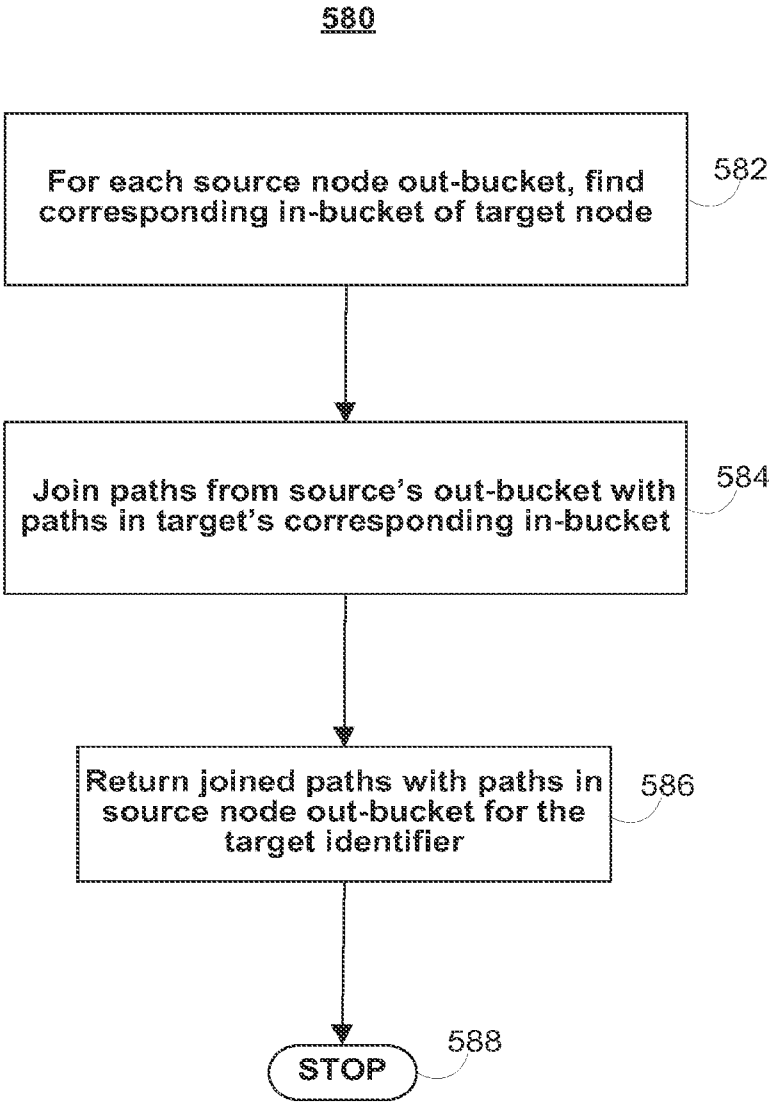

580
For each source node out-bucket, find corresponding in-bucket of target node
582
Join paths from source's out-bucket with paths in target's corresponding in-bucket
584
Return joined paths with paths in source node out-bucket for the target identifier
586
STOP
588

FIG. 5

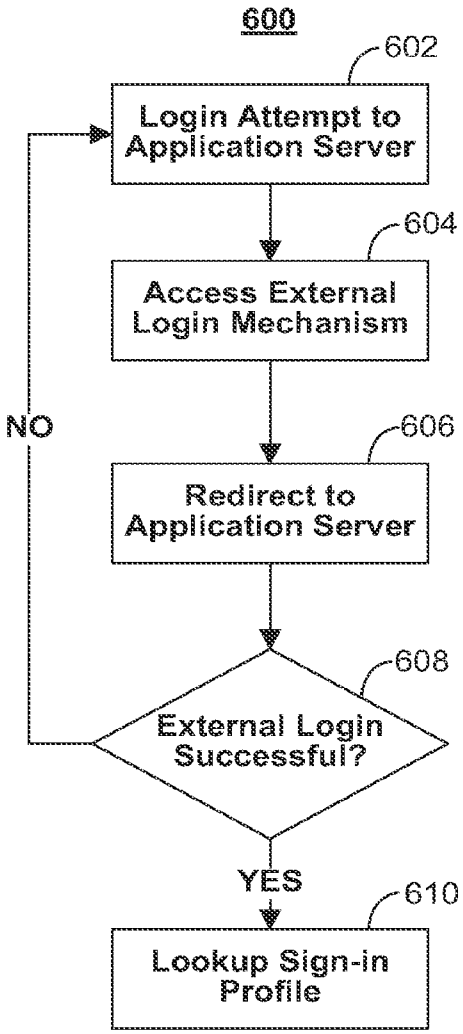

600
602
Login Attempt to Application Server
604
Access External Login Mechanism
606
Redirect to Application Server
608
External Login Successful?
NO
YES
610
Lookup Sign-in Profile

FIG. 6

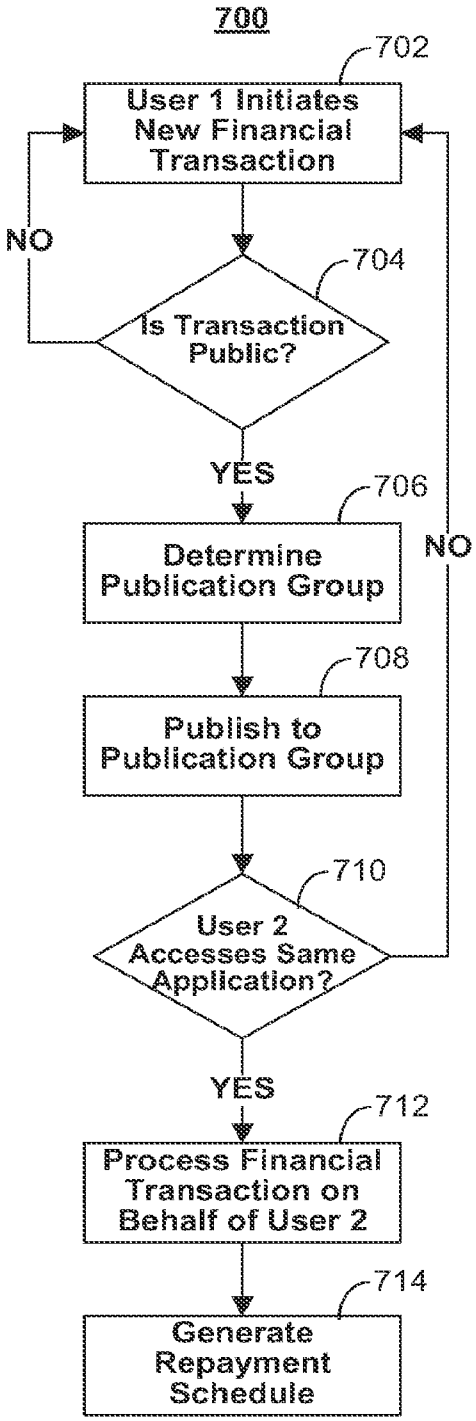
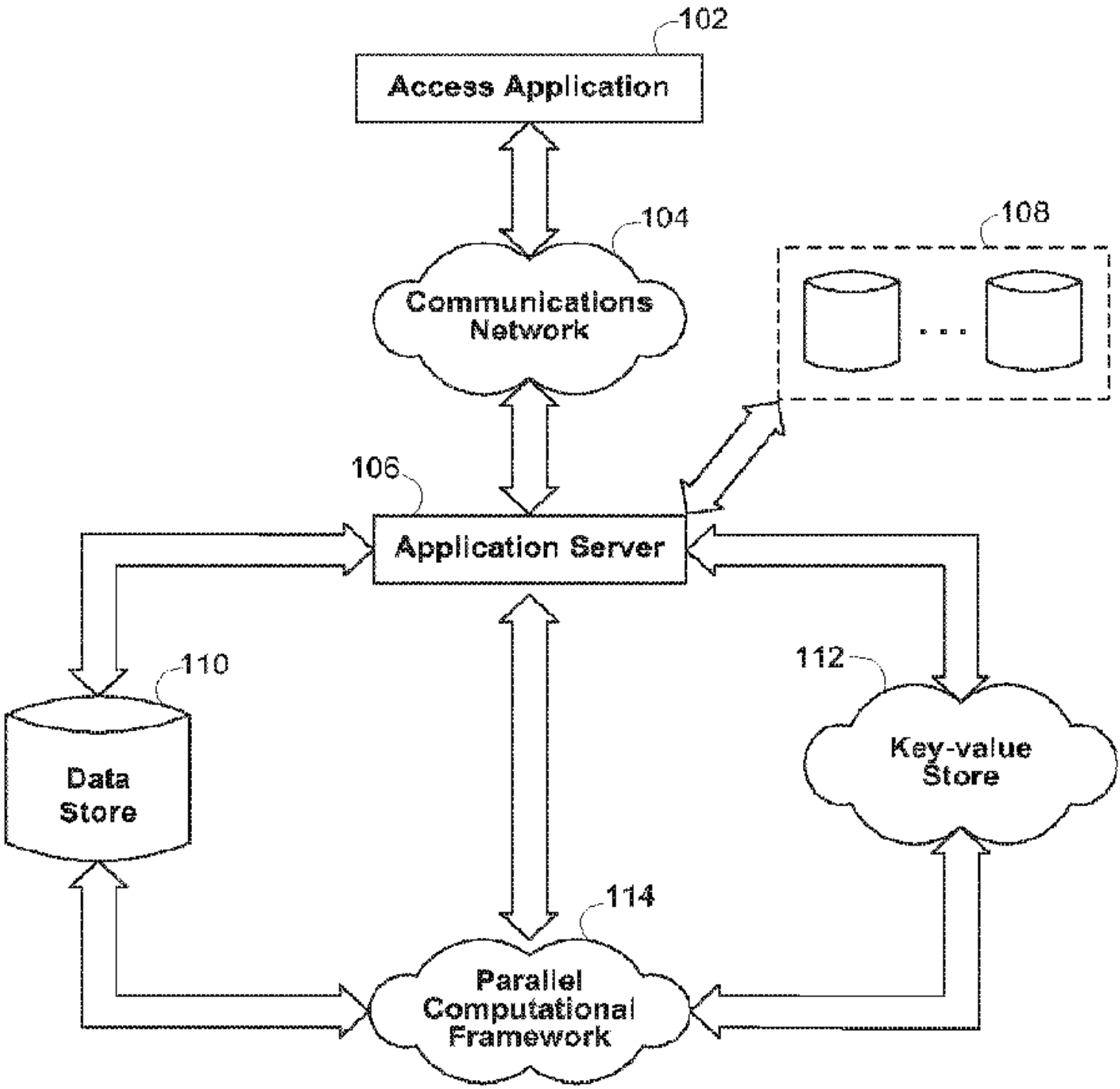

700
702
User 1 Initiates New Financial Transaction
NO
704
Is Transaction Public?
YES
706
Determine Publication Group
NO
708
Publish to Publication Group
710
User 2 Accesses Same Application?
YES
712
Process Financial Transaction on Behalf of User 2
714
Generate Repayment Schedule

FIG. 7


100
102
Access Application
104
Communications Network
108
106
Application Server
110
Data Store
112
Key-value Store
114
Parallel Computational Framework