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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
26 May 2017 (26.05.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/086983 A1

(51) International Patent Classification:
G06F 17/30 (2006.01) *G06F 17/00* (2006.01)

(21) International Application Number:
PCT/US2015/061651

(22) International Filing Date:
19 November 2015 (19.11.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HEWLETT PACKARD ENTERPRISE DEVELOPMENT LP** [US/US]; 11445 Compaq Center Drive West, Houston, Texas 77070 (US).

(72) Inventor: **SPIEGEL, Ofer**; Altalef st No 5, Shabazi st. No 26, 56100 Yehud (IL).

(74) Agents: **LEE, Rachel Jeong-Eun** et al.; Hewlett Packard Enterprise, 3404 E. Harmony Road, Mail Stop 79, Fort Collins, Colorado 80528 (US).

(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, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, 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).

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

Published:

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

(54) Title: PREDICTION MODELS FOR CONCURRENCY CONTROL TYPES

(57) Abstract: Examples disclosed herein relate to prediction models for concurrency control types. Some of the examples enable generating a prediction model based on training data. The training data may comprise a set of access data associated with a data object. The set of access data may comprise: values for a set of attributes of the data object, and an indication whether a conflict occurred during processing of a request to access the data object.

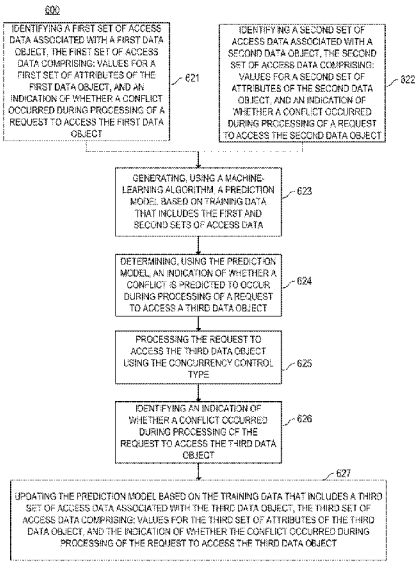


FIG. 6

WO 2017/086983 A1

PREDICTION MODELS FOR CONCURRENCY CONTROL TYPES

BACKGROUND

[0001] Concurrency control is a mechanism that manages and coordinates concurrent accesses to a database in a multi-user database management system (DBMS). In multi-user DBMS environments, database updates performed by one user may conflict with database retrievals and updates performed by another. Concurrency control allows multiple users to simultaneously access the same database while preserving the illusion that each user is executing alone on a dedicated system.

BRIEF DESCRIPTION OF THE DRAWINGS

[0002] The following detailed description references the drawings, wherein:

[0003] FIG. 1 is a block diagram depicting an example environment in which various examples may be implemented as a prediction model system.

[0004] FIG. 2 is a block diagram depicting an example prediction model system.

[0005] FIG. 3 is a block diagram depicting an example machine-readable storage medium comprising instructions executable by a processor for generating prediction models for concurrency control types.

[0006] FIG. 4 is a block diagram depicting an example machine-readable storage medium comprising instructions executable by a processor for generating prediction models for concurrency control types.

[0007] FIG. 5 is a flow diagram depicting an example method for generating prediction models for concurrency control types.

[0008] FIG. 6 is a flow diagram depicting an example method for generating prediction models for concurrency control types.

[0009] FIG. 7 is a table depicting example data objects.

[0010] FIG. 8 is a table depicting example data objects.

DETAILED DESCRIPTION

[0011] The following detailed description refers to the accompanying drawings. Wherever possible, the same reference numbers are used in the drawings and the following description to refer to the same or similar parts. It is to be expressly understood, however, that the drawings are for the purpose of illustration and description only. While several examples are described in this document, modifications, adaptations, and other implementations are possible. Accordingly, the following detailed description does not limit the disclosed examples. Instead, the proper scope of the disclosed examples may be defined by the appended claims.

[0012] Concurrency control is a mechanism that manages and coordinates concurrent accesses to a database in a multi-user database management system (DBMS). In multi-user DBMS environments, database updates performed by one user may conflict with database retrievals and updates performed by another. Concurrency control allows multiple users to simultaneously access the same database while preserving the illusion that each user is executing alone on a dedicated system.

[0013] Example types of concurrency control may include a pessimistic concurrency control type and an optimistic concurrency control type. In pessimistic concurrency control, data may be marked as locked while it is being updated by a user. Other users cannot perform actions that would conflict with the lock until that user commits the update and/or releases the lock. In optimistic concurrency control, before a user commits an update, the system verifies that no other transaction has modified the data being read by the update transaction. If the verification reveals conflicting updates to the same data, the update transaction would be rejected (e.g., the user receives an error). The user may roll back the rejected transaction and start over.

[0014] The pessimistic control type assures data integrity in the price of reduced concurrency (e.g., having transactions wait for other transactions' locks to clear) and/or reduced performance (e.g., managing locks). Thus, the optimistic

concurrency control type is generally used in environments with low data contention (e.g., less likelihood of update conflicts). When conflicts are rare, transactions can complete without the expense of reduced concurrency and performance. However, when conflicts are frequent (e.g., greater likelihood of update conflicts), the pessimistic control type may be a better fit because the cost of repeatedly restarting transactions would significantly hurt performance. Although a particular type of concurrency control could be selected to handle all of the transactions occurring in a database and/or being initiated by an application, it is technically challenging to dynamically determine a concurrency control type to be used for individual data objects.

[0015] Examples disclosed herein provide technical solutions to these technical challenges by generating prediction models for concurrency control types. Some of the examples disclosed herein may enable generating a prediction model based on training data. The training data may comprise a set of access data associated with a data object. The set of access data may comprise: values for a set of attributes of the data object, and an indication whether a conflict occurred during processing of a request to access the data object. In some instances, the set of access data may further include contextual data including, but not being limited to, time of access, and geographical location of the user who requests the access. The prediction model may then determine a probability of a conflict occurring during processing of a request to access another data object. Based on the determined probability of the conflict, a concurrency control type to be used to process this request may be determined.

[0016] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting. As used herein, the singular forms “a,” “an,” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. The term “plurality,” as used herein, is defined as two or more than two. The term “another,” as used herein, is defined as at least a second or more. The term “coupled,” as used herein, is defined as connected, whether directly without any intervening elements or indirectly with at least one intervening elements, unless otherwise indicated. Two elements can be coupled mechanically, electrically, or communicatively linked through a

communication channel, pathway, network, or system. The term “and/or” as used herein refers to and encompasses any and all possible combinations of one or more of the associated listed items. It will also be understood that, although the terms first, second, third, etc. may be used herein to describe various elements, these elements should not be limited by these terms, as these terms are only used to distinguish one element from another unless stated otherwise or the context indicates otherwise. As used herein, the term "includes" means includes but not limited to, the term "including" means including but not limited to. The term "based on" means based at least in part on.

[0017] FIG. 1 is an example environment 100 in which various examples may be implemented as a prediction model system 110. Environment 100 may include various components including server computing device 130 and client computing devices 140 (illustrated as 140A, 140B, …, 140N). Each client computing device 140A, 140B, …, 140N may communicate requests to and/or receive responses from server computing device 130. Server computing device 130 may receive and/or respond to requests from client computing devices 140. Client computing devices 140 may be any type of computing device providing a user interface through which a user can interact with a software application. For example, client computing devices 140 may include a laptop computing device, a desktop computing device, an all-in-one computing device, a thin client, a workstation, a tablet computing device, a mobile phone, an electronic book reader, a network-enabled appliance such as a “Smart” television, and/or other electronic device suitable for displaying a user interface and processing user interactions with the displayed interface. While server computing device 130 is depicted as a single computing device, server computing device 130 may include any number of integrated or distributed computing devices serving at least one software application for consumption by client computing devices 140.

[0018] The various components (e.g., components 129, 130, and/or 140) depicted in FIG. 1 may be coupled to at least one other component via a network 50. Network 50 may comprise any infrastructure or combination of infrastructures that enable electronic communication between the components. For example, network 50 may include at least one of the Internet, an intranet, a PAN (Personal

Area Network), a LAN (Local Area Network), a WAN (Wide Area Network), a SAN (Storage Area Network), a MAN (Metropolitan Area Network), a wireless network, a cellular communications network, a Public Switched Telephone Network, and/or other network. According to various implementations, prediction model system 110 and the various components described herein may be implemented in hardware and/or a combination of hardware and programming that configures hardware. Furthermore, in FIG. 1 and other Figures described herein, different numbers of components or entities than depicted may be used.

[0019] Prediction model system 110 may comprise an access data engine 121, a prediction model engine 122, a request process engine 123, and/or other engines. The term "engine", as used herein, refers to a combination of hardware and programming that performs a designated function. As is illustrated respect to FIGS. 3-4, the hardware of each engine, for example, may include one or both of a processor and a machine-readable storage medium, while the programming is instructions or code stored on the machine-readable storage medium and executable by the processor to perform the designated function.

[0020] Access data engine 121 may identify access data associated with various data objects. For example, access data engine 121 may identify a first set of access data associated with a first data object. The first set of access data may comprise: values for a first set of attributes of the first data object, a concurrency control type that was used to process a request to access the first data object, and/or an indication of whether a conflict occurred during processing of the request to access the first data object.

[0021] A "request to access" a particular data object, as used herein, may include a request to read, delete, and/or update the data object and/or other database transaction related to the data object.

[0022] A "concurrency control type that was used to process a request to access a particular data object," as used herein, may comprise a pessimistic concurrency control type, an optimistic concurrency control type, and/or other types of concurrency control. For example, in pessimistic concurrency control, data may be marked as locked while it is being updated by a user. Other users cannot perform actions that would conflict with the lock until that user commits the

update and/or releases the lock. In optimistic concurrency control, before a user commits an update, the system verifies that no other transaction has modified the data being read by the update transaction. If the verification reveals conflicting updates to the same data, the update transaction would be rejected (e.g., the user receives an error). The user may roll back the rejected transaction and start over. Note that although two concurrency control types (e.g., pessimistic and optimistic types) are discussed in the examples described herein (e.g., examples illustrated in FIGS. 7-8), other types of concurrency control known in the art are also contemplated.

[0023] An "indication of a conflict that occurred," as used herein, may comprise: a first indication that no occurrence of a conflict was detected during processing of a request to access a particular data object (e.g., denoted by "None"), a second indication that an occurrence of a conflict was detected during the processing (e.g., denoted by "Conflict"), and/or other indications of a conflict. In some implementations, the second indication may indicate a type of a conflict that occurred: that the conflict occurred due to a locking of that particular object (e.g., denoted by "Lock"), that the conflict occurred due to a rejection of an update to the particular data object (e.g., denoted by "Reject"), and/or other types of a conflict.

[0024] Suppose that an example dataset as illustrated in FIG. 7 represents access data associated with 11 data objects (e.g., Data Object IDs #1-11). In one example, for Data Object ID #1 (e.g., the first data object), the data object is associated with a set of attributes including Attributes A-H. The values of Attributes A-H include: 3, 66, East, Base, F, 0.123, Desk, and 12:52, respectively. The access data may further include a concurrency control type (e.g., optimistic concurrency control) that was used during processing of a request to access Data Object ID #1. The access data may further include an indication of whether a conflict actually occurred (e.g., "None" meaning no conflict occurred) during the processing (e.g., given that the optimistic concurrency control was used). In another example, for Data Object ID #4 (e.g., a second data object), the data object is associated with a set of attributes including Attributes A-H. The values of Attributes A-H include: 3, 72, East, Base, M, 0.343, Desk, and 12:54, respectively.

The access data may further include a concurrency control type (e.g., optimistic concurrency control) that was used during processing of a request to access Data Object ID #4. The access data may further include an indication of whether a conflict actually occurred (e.g., "Reject" meaning the access request was rejected) during the processing (e.g., given that the optimistic concurrency control was used). In yet another example, for Data Object ID #10 (e.g., a third data object), the data object is associated with a set of attributes including Attributes A-H. The values of Attributes A-H include: 1, 29, Center, Gold, F, 0.123, Desk, and 13:58, respectively. The access data may further include a concurrency control type (e.g., pessimistic concurrency control) that was used during processing of a request to access Data Object ID #10. The access data may further include an indication of whether a conflict actually occurred (e.g., "Lock" meaning Data Object ID #10 is locked by another user) during the processing (e.g., given that the pessimistic concurrency control was used).

[0025] The access data of various data objects like the example dataset shown in FIG. 7 may be used as training data to train a prediction model, as further discussed herein with respect to prediction model engine 122.

[0026] Prediction model engine 122 may generate, using a machine-learning algorithm, a prediction model based on the training data that includes the access data (e.g., identified by access data engine 121). Any machine-learning algorithm known in the art may be used to generate a prediction model. The prediction model may be trained to recognize patterns in the training data. In some implementations, a trained prediction model may be tested using test data to ensure that its output is validated within an acceptable margin of error. A properly trained and/or tested prediction model may use the knowledge discovered from the training data to predict an output (e.g., an indication of whether a conflict is predicted to occur) given a new data object and/or given at least a portion of its access data (e.g., values for a set of attributes of the new data object). Prior to processing a request to access a new data object, the new data object may be identified and/or submitted to the prediction model to predict an indication of whether a conflict is predicted to occur. Based on this indication of a predicted

conflict, prediction model engine 122 may determine a concurrency control type to be used to process the request to access that new data object.

[0027] An "indication of a conflict that is predicted to occur," as used herein, may comprise: a first indication that no conflict is predicted to occur during processing of a request to access a particular data object (e.g., denoted by "None"), a second indication that a conflict is predicted to occur during the processing (e.g., denoted by "Conflict"), and/or other indications of a conflict. In some implementations, the second indication may indicate a type of a conflict that is predicted to occur: that the conflict is predicted to occur due to a locking of that particular object (e.g., denoted by "Lock"), that the conflict is predicted to due to a rejection of an update to the particular data object (e.g., denoted by "Reject"), and/or other types of a conflict.

[0028] For example, suppose that an example dataset as illustrated in FIG. 8 includes 3 new data objects (e.g., Data Object IDs #12-14) for which a concurrency control type should be determined using the prediction model. Data Object ID #12 having values for Attributes A-H (e.g., 432, 33, East, Gold, F, 0.523, Tech, and 12:52, respectively) may be submitted to the prediction model (e.g., that has been trained and/or tested by prediction model engine 122 as discussed above). The prediction model may analyze patterns in the values for those attributes and use the result of the analysis to predict an indication of whether a conflict is predicted to occur during processing of a request to access Data Object ID #12. In other words, the prediction model may determine a probability of a conflict occurring during the processing of the request to access Data Object ID #12.

[0029] In the example illustrated in FIG. 8, the prediction model, based on the values of the attributes associated with Data Object ID #12, determines that a conflict (e.g., "Reject") is predicted to occur during processing of the request to access Data Object ID #12. Based on the conflict (e.g., "Reject") that is predicted to occur, prediction model engine 122 may select a particular type of concurrency control (e.g., pessimistic) to process the request to access Data Object ID #12.

[0030] With respect to Data Object ID #13 in FIG. 8, in response to determining that a conflict (e.g., "Lock") is predicted to occur during processing of a request to

access Data Object ID #13 using the prediction model, prediction model engine 122 may determine that the pessimistic concurrency control type should be used to process the request to access Data Object ID #13 given the indication that a conflict is likely to occur. On the other hand, with respect to Data Object ID #14 in FIG.8, prediction model engine 122 determines that no conflict (e.g., "None") is predicted to occur based on the values of attributes associated with Data Object ID #14. In response to that indication, prediction model engine 122 may select a particular type of concurrency control (e.g., optimistic).

[0031] Request process engine 123 may process a request to access a data object. For example, request process engine 123 may read, delete, and/or update the data object according to the request. The processing may result in the requested transaction being successfully committed or in the requested transaction failing to commit (e.g., because a conflict occurred). During processing of a request to access a data object, request process engine 123 may use a particular concurrency control type that has been selected, determined, and/or suggested by prediction model engine 122 for that particular data object. Returning to the example illustrated in FIG. 8, request process engine 123 may process a request to access Data Object ID #12 using the pessimistic concurrency control type (e.g., which was selected based on the indication that a conflict, "Reject," was predicted to occur).

[0032] In some implementations, after applying the concurrency control type that was selected, determined, and/or suggested by prediction model engine 122 to process a request to access a new data object, prediction model engine 122 may monitor for an indication of whether a conflict actually occurred or not during the processing of the request to access the new data object. Returning to the example illustrated in FIG. 8, after applying the pessimistic concurrency control type in processing a request to access Data Object ID #12, an occurrence of a conflict (e.g., "Lock") was detected. This information may be included in the training data so that the prediction model may be continuously trained, tested, and/or improved. As such, prediction model engine 122 may update the prediction model based on the training data that includes a set of access data associated with Data Object ID #12. The set of access data may comprise: values

for the set of attributes of Data Object ID #12 (e.g., 432, 33, East, Gold, F, 0.523, Tech, and 12:52), a concurrency control type (e.g., pessimistic), and/or an indication of whether a conflict occurred during processing of the request to access Data Object ID #12 (e.g., "Lock").

[0033] In performing their respective functions, engines 121-123 may access data storage 129 and/or other suitable database(s). Data storage 129 may represent any memory accessible to prediction model system 110 that can be used to store and retrieve data. Data storage 129 and/or other database may comprise random access memory (RAM), read-only memory (ROM), electrically-erasable programmable read-only memory (EEPROM), cache memory, floppy disks, hard disks, optical disks, tapes, solid state drives, flash drives, portable compact disks, and/or other storage media for storing computer-executable instructions and/or data. Prediction model system 110 may access data storage 129 locally or remotely via network 50 or other networks.

[0034] Data storage 129 may include a database to organize and store data. The database may reside in a single or multiple physical device(s) and in a single or multiple physical location(s). The database may store a plurality of types of data and/or files and associated data or file description, administrative information, or any other data.

[0035] FIG. 2 is a block diagram depicting an example prediction model system 210. Prediction model system 210 may comprise an access data engine 221, a prediction model engine 222, and/or other engines. Engines 221-222 represent engines 121-122, respectively.

[0036] FIG. 3 is a block diagram depicting an example machine-readable storage medium 310 comprising instructions executable by a processor for generating prediction models for concurrency control types.

[0037] In the foregoing discussion, engines 121-123 were described as combinations of hardware and programming. Engines 121-123 may be implemented in a number of fashions. Referring to FIG. 3, the programming may be processor executable instructions 321 stored on a machine-readable storage medium 310 and the hardware may include a processor 311 for executing those instructions. Thus, machine-readable storage medium 310 can be said to store

program instructions or code that when executed by processor 311 implements prediction model system 110 of FIG. 1.

[0038] In FIG. 3, the executable program instructions in machine-readable storage medium 310 are depicted as prediction model instructions 321. Instructions 321 represent program instructions that, when executed, cause processor 311 to implement engine 122.

[0039] FIG. 4 is a block diagram depicting an example machine-readable storage medium 410 comprising instructions executable by a processor for generating prediction models for concurrency control types.

[0040] Referring to FIG. 4, the programming may be processor executable instructions 421-423 stored on a machine-readable storage medium 410 and the hardware may include a processor 411 for executing those instructions. Thus, machine-readable storage medium 410 can be said to store program instructions or code that when executed by processor 411 implements prediction model system 110 of FIG. 1.

[0041] In FIG. 4, the executable program instructions in machine-readable storage medium 410 are depicted as access data instructions 421, prediction model instructions 422, and request process instructions 423. Instructions 421-423 represent program instructions that, when executed, cause processor 411 to implement engines 121-123, respectively.

[0042] Machine-readable storage medium 310 (or machine-readable storage medium 410) may be any electronic, magnetic, optical, or other physical storage device that contains or stores executable instructions. In some implementations, machine-readable storage medium 310 (or machine-readable storage medium 410) may be a non-transitory storage medium, where the term “non-transitory” does not encompass transitory propagating signals. Machine-readable storage medium 310 (or machine-readable storage medium 410) may be implemented in a single device or distributed across devices. Likewise, processor 311 (or processor 411) may represent any number of processors capable of executing instructions stored by machine-readable storage medium 310 (or machine-readable storage medium 410). Processor 311 (or processor 411) may be integrated in a single device or distributed across devices. Further, machine-

readable storage medium 310 (or machine-readable storage medium 410) may be fully or partially integrated in the same device as processor 311 (or processor 411), or it may be separate but accessible to that device and processor 311 (or processor 411).

[0043] In one example, the program instructions may be part of an installation package that when installed can be executed by processor 311 (or processor 411) to implement prediction model system 110. In this case, machine-readable storage medium 310 (or machine-readable storage medium 410) may be a portable medium such as a floppy disk, CD, DVD, or flash drive or a memory maintained by a server from which the installation package can be downloaded and installed. In another example, the program instructions may be part of an application or applications already installed. Here, machine-readable storage medium 310 (or machine-readable storage medium 410) may include a hard disk, optical disk, tapes, solid state drives, RAM, ROM, EEPROM, or the like.

[0044] Processor 311 may be at least one central processing unit (CPU), microprocessor, and/or other hardware device suitable for retrieval and execution of instructions stored in machine-readable storage medium 310. Processor 311 may fetch, decode, and execute program instructions 321, and/or other instructions. As an alternative or in addition to retrieving and executing instructions, processor 311 may include at least one electronic circuit comprising a number of electronic components for performing the functionality of instructions 321, and/or other instructions.

[0045] Processor 411 may be at least one central processing unit (CPU), microprocessor, and/or other hardware device suitable for retrieval and execution of instructions stored in machine-readable storage medium 410. Processor 411 may fetch, decode, and execute program instructions 421-423, and/or other instructions. As an alternative or in addition to retrieving and executing instructions, processor 411 may include at least one electronic circuit comprising a number of electronic components for performing the functionality of at least one of instructions 421-423, and/or other instructions.

[0046] FIG. 5 is a flow diagram depicting an example method 500 for generating prediction models for concurrency control types. The various

processing blocks and/or data flows depicted in FIG. 5 (and in the other drawing figures such as FIG. 6) are described in greater detail herein. The described processing blocks may be accomplished using some or all of the system components described in detail above and, in some implementations, various processing blocks may be performed in different sequences and various processing blocks may be omitted. Additional processing blocks may be performed along with some or all of the processing blocks shown in the depicted flow diagrams. Some processing blocks may be performed simultaneously. Accordingly, method 500 as illustrated (and described in greater detail below) is meant to be an example and, as such, should not be viewed as limiting. Method 500 may be implemented in the form of executable instructions stored on a machine-readable storage medium, such as storage medium 310, and/or in the form of electronic circuitry.

[0047] In block 521, method 500 may include identifying a first set of access data associated with a first data object. The first set of access data may comprise: values for a first set of attributes of the first data object, and an indication of whether a conflict occurred during processing of a request to access the first data object. Referring back to FIG. 1, access data engine 121 may be responsible for implementing block 521.

[0048] In block 522, method 500 may include identifying a second set of access data associated with a second data object. The second set of access data may comprise: values for a second set of attributes of the second data object, and an indication of whether a conflict occurred during processing of a request to access the second data object. Referring back to FIG. 1, access data engine 121 may be responsible for implementing block 522.

[0049] In block 523, method 500 may include generating, using a machine-learning algorithm, a prediction model based on training data that includes the first and second sets of access data. Referring back to FIG. 1, prediction model engine 122 may be responsible for implementing block 523.

[0050] FIG. 6 is a flow diagram depicting an example method 600 for generating prediction models for concurrency control types. Method 600 as illustrated (and described in greater detail below) is meant to be an example and,

as such, should not be viewed as limiting. Method 600 may be implemented in the form of executable instructions stored on a machine-readable storage medium, such as storage medium 310, and/or in the form of electronic circuitry.

[0051] In block 621, method 600 may include identifying a first set of access data associated with a first data object. The first set of access data may comprise: values for a first set of attributes of the first data object, and an indication of whether a conflict occurred during processing of a request to access the first data object. Referring back to FIG. 1, access data engine 121 may be responsible for implementing block 621.

[0052] In block 622, method 600 may include identifying a second set of access data associated with a second data object. The second set of access data may comprise: values for a second set of attributes of the second data object, and an indication of whether a conflict occurred during processing of a request to access the second data object. Referring back to FIG. 1, access data engine 121 may be responsible for implementing block 622.

[0053] In block 623, method 600 may include generating, using a machine-learning algorithm, a prediction model based on training data that includes the first and second sets of access data. Referring back to FIG. 1, prediction model engine 122 may be responsible for implementing block 623.

[0054] In block 624, method 600 may include determining, using the prediction model, an indication of whether a conflict is predicted to occur during processing of a request to access to a third data object. Referring back to FIG. 1, prediction model engine 122 may be responsible for implementing block 624.

[0055] In block 625, method 600 may include processing the request to access the third data object using the concurrency control type. Referring back to FIG. 1, request process engine 123 may be responsible for implementing block 625.

[0056] In block 626, method 600 may include identifying an indication of whether a conflict occurred during processing of the request to access the third data object. Referring back to FIG. 1, request process engine 123 may be responsible for implementing block 626.

[0057] In block 627, method 600 may include updating the prediction model based on the training data that includes a third set of access data associated

with the third data object. The third set of access data may comprise: values for the third set of attributes of the third data object, and the indication of whether the conflict occurred during processing of the request to access the third data object. Referring back to FIG. 1, prediction model engine 122 may be responsible for implementing block 627.

[0058] FIGS. 7-8 are discussed herein with respect to FIG.1.

[0059] The foregoing disclosure describes a number of example implementations for prediction models for concurrency control types. The disclosed examples may include systems, devices, computer-readable storage media, and methods for prediction models for concurrency control types. For purposes of explanation, certain examples are described with reference to the components illustrated in FIGS. 1-4. The functionality of the illustrated components may overlap, however, and may be present in a fewer or greater number of elements and components.

[0060] Further, all or part of the functionality of illustrated elements may co-exist or be distributed among several geographically dispersed locations. Moreover, the disclosed examples may be implemented in various environments and are not limited to the illustrated examples. Further, the sequence of operations described in connection with FIGS. 5-6 are examples and are not intended to be limiting. Additional or fewer operations or combinations of operations may be used or may vary without departing from the scope of the disclosed examples. Furthermore, implementations consistent with the disclosed examples need not perform the sequence of operations in any particular order. Thus, the present disclosure merely sets forth possible examples of implementations, and many variations and modifications may be made to the described examples. All such modifications and variations are intended to be included within the scope of this disclosure and protected by the following claims.

CLAIMS

1. A method for generating prediction models for concurrency control types, the method comprising:

identifying a first set of access data associated with a first data object, the first set of access data comprising: values for a first set of attributes of the first data object, and an indication of whether a conflict occurred during processing of a request to access the first data object;

identifying a second set of access data associated with a second data object, the second set of access data comprising: values for a second set of attributes of the second data object, and an indication of whether a conflict occurred during processing of a request to access the second data object; and

generating, using a machine-learning algorithm, a prediction model based on training data that includes the first and second sets of access data.

2. The method of claim 1, further comprising:

determining, using the prediction model, an indication of whether a conflict is predicted to occur during processing of a request to access a third data object; and

determining, based on the determined indication, a concurrency control type to be used to process the request to access the third data object.

3. The method of claim 2, wherein determining, using the prediction model, the indication of whether the conflict is predicted to occur further comprises:

identifying values for a third set of attributes of the third data object; and

determining, using the prediction model, the indication of whether the conflict is predicted to occur based on the values for the third set of attributes.

4. The method of claim 2, further comprising:

processing the request to access the third data object using the concurrency control type;

identifying an indication of whether a conflict occurred during processing of the request to access the third data object; and

updating the prediction model based on the training data that includes a third set of access data associated with the third data object, the third set of access data comprising: values for the third set of attributes of the third data object, and the indication of whether the conflict occurred during processing of the request to access the third data object.

5. The method of claim 2, wherein the concurrency control type comprises a pessimistic concurrency control type or an optimistic concurrency control type

6. The method of claim 2, wherein the indication of whether the conflict is predicted to occur during processing of the request to access the third data object comprises a first indication that no conflict is predicted to occur, a second indication that the conflict is predicted to occur due to a locking of the third data object, or a third indication that the conflict is predicted to occur due to a rejection of an update to the third data object.

7. The method of claim 6, further comprising:

in response to the first indication, processing the request to access the third data object using an optimistic concurrency control type; and

in response to the second or third indication, processing the request to access the third data object using a pessimistic concurrency control type.

8. A non-transitory machine-readable storage medium comprising instructions executable by a processor of a computing device for generating prediction models for concurrency control types, the machine-readable storage medium comprising:

instructions to identify a prediction model that is generated based on training data comprising: (i) values for a first set of attributes of the first data object, and (ii) an indication of whether a conflict occurred during processing of a request to access the first data object;

instructions to determine, using the prediction model, a probability of a conflict occurring during processing of a request to access a second data object; and

instructions to determine, based on the probability of the conflict for the second data object, a concurrency control type to be used to process the request to access the second data object.

9. The non-transitory machine-readable storage medium of claim 8, further comprising:

instructions to process the request to access the second data object using the concurrency control type;

instructions to identify an indication of whether a conflict occurred during processing of the request to access the second data object; and

instructions to include a second set of access data associated with the second data object in the training data, the second set of access data comprising: values for a second set of attributes of the second data object, and the indication of whether the conflict occurred during processing of the request to access the second data object.

10. The non-transitory machine-readable storage medium of claim 8, wherein the second data object includes values of a second set of attributes of the second data object, further comprising:

instructions to use the values of the second set of attributes to determine the probability of the conflict occurring during processing of the request to access the second data object.

11. The non-transitory machine-readable storage medium of claim 8, further comprising:

instructions to determine, using the prediction model, a probability of a conflict occurring during processing of a request to access a third data object; and

instructions to determine, based on the probability of the conflict for the third data object, a concurrency control type to be used to process the request to access the third data object.

12. A system for generating prediction models for concurrency control types comprising:

a processor that:

identifies a first set of access data associated with a first data object, the first set of access data comprising: values for a first set of attributes of the first data object, and an indication of whether a conflict occurred during processing of a request to access the first data object;

generates, using a machine-learning algorithm, a prediction model based on training data that includes the first set of access data;

identifies a second data object;

determines, using the prediction model, an indication of whether a conflict is predicted to occur during processing of a request to access the second data object; and

determines, based on the determined indication, the concurrency control type to be used to process the request to access the second data object.

13. The system of claim 12, wherein the first set of access data comprises a concurrency control type that was used to process the request to access the first data object.

14. The system of claim 12, the processor that:

identifies a third data object for which a concurrency control type is determined using the prediction model;

determines, using the prediction model, an indication of whether a conflict is predicted to occur during processing of a request to access the third data object; and

determines, based on the determined indication, the concurrency control type to be used to process the request to access the third data object.

15. The system of claim 14, wherein the determination of the concurrency control type for the second data object and the determination of the concurrency control type for the third data object occur within a designated time period prior to processing of the request to access the second and third data objects.

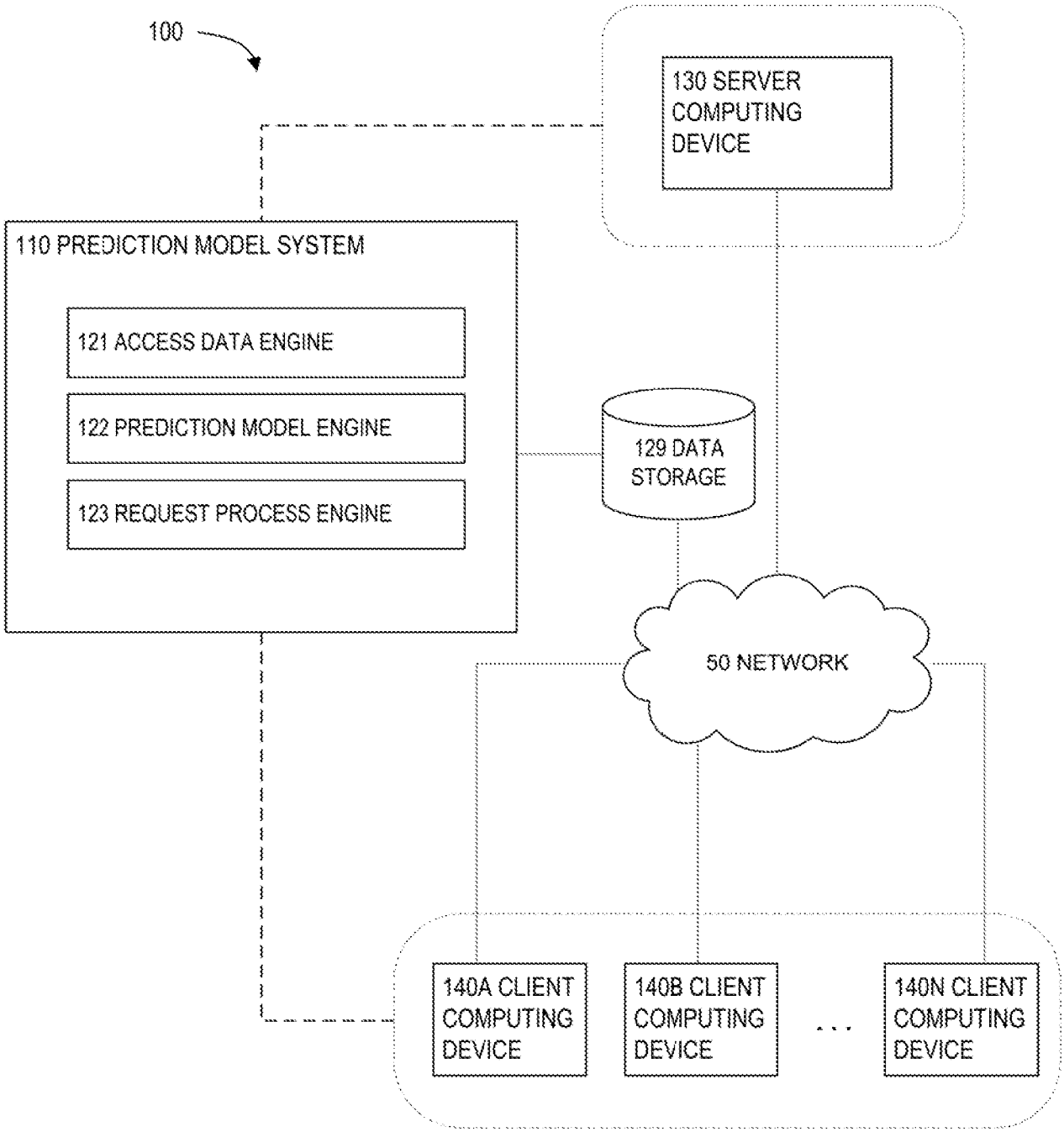

100
130 SERVER COMPUTING DEVICE
110 PREDICTION MODEL SYSTEM
121 ACCESS DATA ENGINE
122 PREDICTION MODEL ENGINE
123 REQUEST PROCESS ENGINE
129 DATA STORAGE
50 NETWORK
140A CLIENT COMPUTING DEVICE
140B CLIENT COMPUTING DEVICE
. . .
140N CLIENT COMPUTING DEVICE

FIG. 1

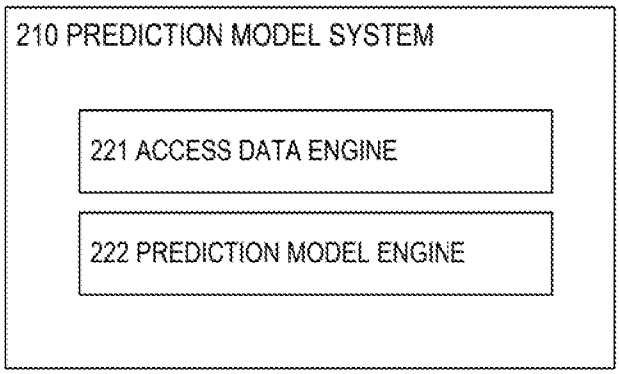

210 PREDICTION MODEL SYSTEM
221 ACCESS DATA ENGINE
222 PREDICTION MODEL ENGINE

FIG. 2

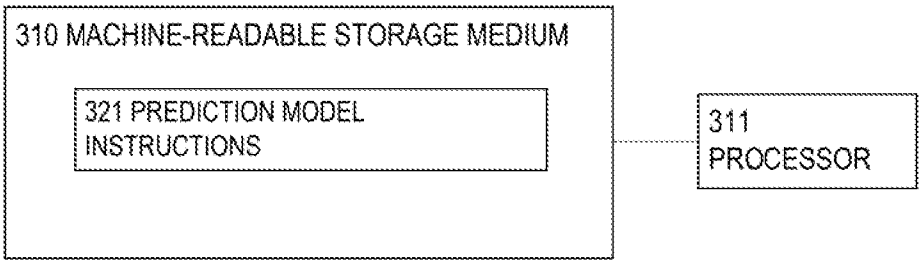

310 MACHINE-READABLE STORAGE MEDIUM
321 PREDICTION MODEL INSTRUCTIONS
311 PROCESSOR

FIG. 3

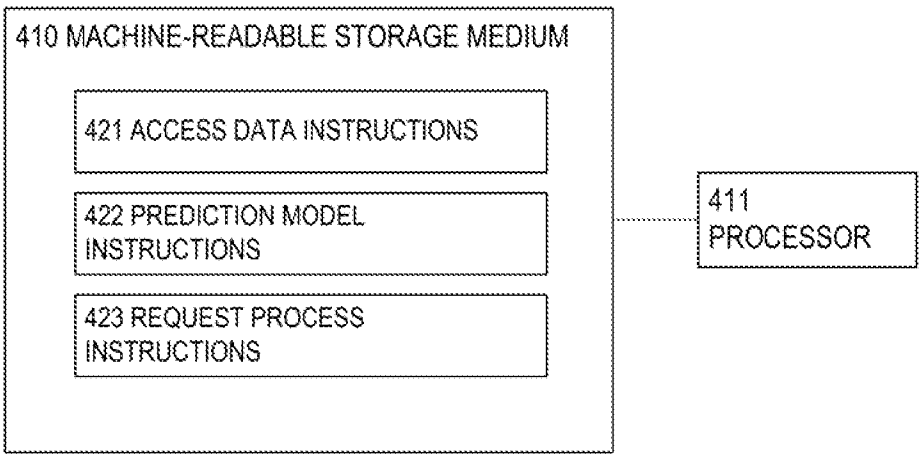

410 MACHINE-READABLE STORAGE MEDIUM
421 ACCESS DATA INSTRUCTIONS
422 PREDICTION MODEL INSTRUCTIONS
423 REQUEST PROCESS INSTRUCTIONS
411 PROCESSOR

FIG. 4

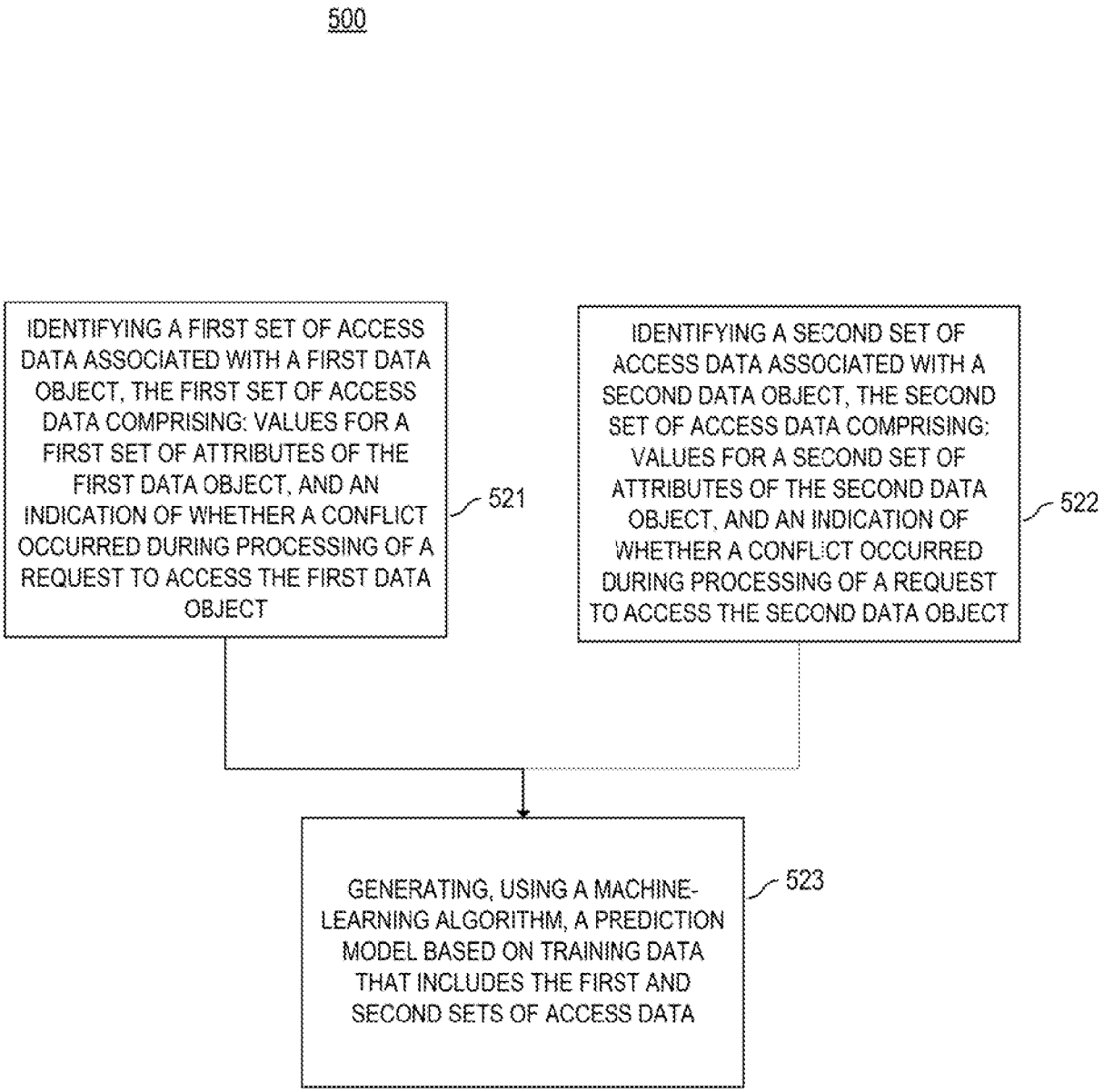

500
IDENTIFYING A FIRST SET OF ACCESS DATA ASSOCIATED WITH A FIRST DATA OBJECT, THE FIRST SET OF ACCESS DATA COMPRISING: VALUES FOR A FIRST SET OF ATTRIBUTES OF THE FIRST DATA OBJECT, AND AN INDICATION OF WHETHER A CONFLICT OCCURRED DURING PROCESSING OF A REQUEST TO ACCESS THE FIRST DATA OBJECT
521
IDENTIFYING A SECOND SET OF ACCESS DATA ASSOCIATED WITH A SECOND DATA OBJECT, THE SECOND SET OF ACCESS DATA COMPRISING: VALUES FOR A SECOND SET OF ATTRIBUTES OF THE SECOND DATA OBJECT, AND AN INDICATION OF WHETHER A CONFLICT OCCURRED DURING PROCESSING OF A REQUEST TO ACCESS THE SECOND DATA OBJECT
522
GENERATING, USING A MACHINE-LEARNING ALGORITHM, A PREDICTION MODEL BASED ON TRAINING DATA THAT INCLUDES THE FIRST AND SECOND SETS OF ACCESS DATA
523

FIG. 5

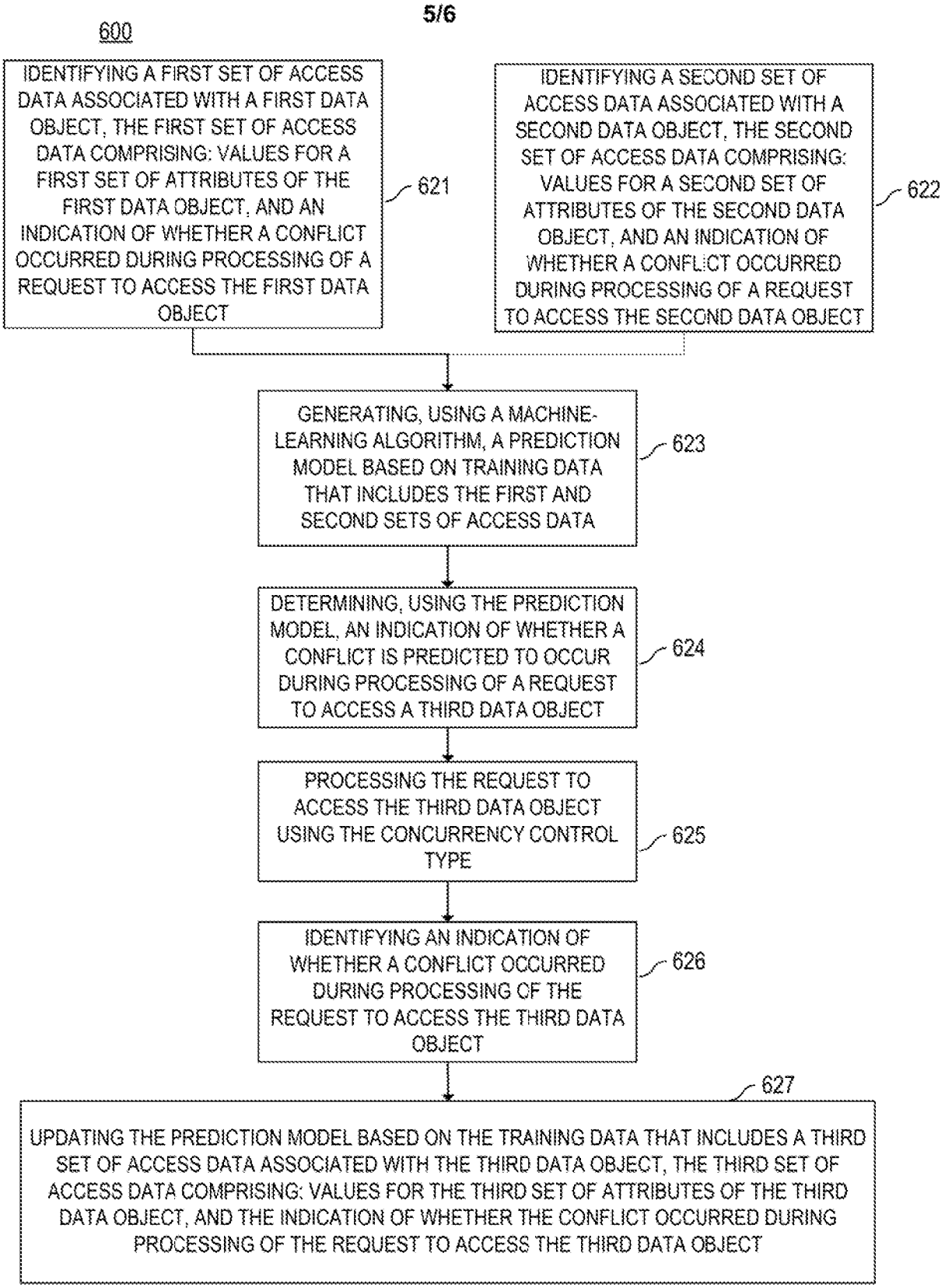

600
IDENTIFYING A FIRST SET OF ACCESS DATA ASSOCIATED WITH A FIRST DATA OBJECT, THE FIRST SET OF ACCESS DATA COMPRISING: VALUES FOR A FIRST SET OF ATTRIBUTES OF THE FIRST DATA OBJECT, AND AN INDICATION OF WHETHER A CONFLICT OCCURRED DURING PROCESSING OF A REQUEST TO ACCESS THE FIRST DATA OBJECT
621
IDENTIFYING A SECOND SET OF ACCESS DATA ASSOCIATED WITH A SECOND DATA OBJECT, THE SECOND SET OF ACCESS DATA COMPRISING: VALUES FOR A SECOND SET OF ATTRIBUTES OF THE SECOND DATA OBJECT, AND AN INDICATION OF WHETHER A CONFLICT OCCURRED DURING PROCESSING OF A REQUEST TO ACCESS THE SECOND DATA OBJECT
622
GENERATING, USING A MACHINE-LEARNING ALGORITHM, A PREDICTION MODEL BASED ON TRAINING DATA THAT INCLUDES THE FIRST AND SECOND SETS OF ACCESS DATA
623
DETERMINING, USING THE PREDICTION MODEL, AN INDICATION OF WHETHER A CONFLICT IS PREDICTED TO OCCUR DURING PROCESSING OF A REQUEST TO ACCESS A THIRD DATA OBJECT
624
PROCESSING THE REQUEST TO ACCESS THE THIRD DATA OBJECT USING THE CONCURRENCY CONTROL TYPE
625
IDENTIFYING AN INDICATION OF WHETHER A CONFLICT OCCURRED DURING PROCESSING OF THE REQUEST TO ACCESS THE THIRD DATA OBJECT
626
627
UPDATING THE PREDICTION MODEL BASED ON THE TRAINING DATA THAT INCLUDES A THIRD SET OF ACCESS DATA ASSOCIATED WITH THE THIRD DATA OBJECT, THE THIRD SET OF ACCESS DATA COMPRISING: VALUES FOR THE THIRD SET OF ATTRIBUTES OF THE THIRD DATA OBJECT, AND THE INDICATION OF WHETHER THE CONFLICT OCCURRED DURING PROCESSING OF THE REQUEST TO ACCESS THE THIRD DATA OBJECT

FIG. 6

Data Object ID	Attribute A	Attribute B	Attribute C	Attribute D	Attribute E	Attribute F	Attribute G	Attribute H	Concurrency Control Type Used	Indication of Conflict Occurred
1	3	66	East	Base	F	0.123	Desk	12:52	Optimistic	None
2	6	43	East	Base	M	0.342	Asst	12:52	Optimistic	None
3	21	34	Center	Base	M	0.534	Asst	12:54	Optimistic	None
4	3	72	East	Base	M	0.343	Desk	12:54	Optimistic	Reject
5	20	30	Center	Gold	F	0.342	Desk	12:55	Optimistic	None
6	4	3	West	Base	M	0.432	Tech	13:56	Optimistic	None
7	70	56	West	Base	F	0.123	Desk	12:56	Optimistic	None
8	3	23	West	Base	F	0.544	Asst	13:57	Pessmistic	None
9	432	22	Center	Gold	F	0.545	Asst	13:58	Pessmistic	None
10	1	29	Center	Gold	F	0.123	Desk	13:58	Pessmistic	Lock
11	15	43	East	Gold	F	0.123	Desk	13:59	Pessmistic	None

FIG. 7

Data Object ID	Attribute A	Attribute B	Attribute C	Attribute D	Attribute E	Attribute F	Attribute G	Attribute H	Indication of Predicted Conflict	Concurrency Control Type To Be Used	Indication of Conflict Occurred
12	432	33	East	Gold	F	0.523	Tech	12:52	Reject	Pessmistic	Lock
13	15	22	Center	Gold	M	0.123	Tech	13:55	Lock	Pessmistic	None
14	15	43	West	Base	M	0.545	Asst	13:55	None	Optimistic	Reject

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2015/061651

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/30(2006.01)i, G06F 17/00(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 17/30; G06F 12/16; G06F 7/00; G06F 9/46; G06F 12/00; G06F 17/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: prediction model, concurrency, conflict control, access, transaction, learning, database, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2002-0138483 A1 (ROBERT F. BRETL et al.) 26 September 2002 See paragraphs [0002] and [0020]-[0022]; claims 1 and 7; and figure 2.	1-15
A	US 2011-0185359 A1 (DHRUVA CHAKRABARTI) 28 July 2011 See paragraphs [0027]-[0028] and figures 7-8.	1-15
A	US 2011-0302143 A1 (DAVID B. LOMET) 08 December 2011 See paragraphs [0031]-[0032] and [0167] and figures 2-3.	1-15
A	US 2002-0091717 A1 (VICTOR T. SHI et al.) 11 July 2002 See paragraphs [0019]-[0020]; claim 1; and figure 1.	1-15
A	WO 2008-018963 A1 (MICROSOFT CORPORATION) 14 February 2008 See paragraphs [033]-[034] and figure 6.	1-15

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 August 2016 (30.08.2016)	**30 August 2016 (30.08.2016)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
NHO, Ji Myong
Telephone No. +82-42-481-8528

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2015/061651

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2002-0138483 A1	26/09/2002	US 6681226 B2	20/01/2004
US 2011-0185359 A1	28/07/2011	US 8364909 B2	29/01/2013
US 2011-0302143 A1	08/12/2011	None	
US 2002-0091717 A1	11/07/2002	US 2003-0236786 A1	25/12/2003
		US 7051028 B2	23/05/2006
		US 7089244 B2	08/08/2006
WO 2008-018963 A1	14/02/2008	EP 2047369 A1	15/04/2009
		EP 2047369 A4	26/05/2010
		MX 2009000859 A	03/02/2009
		RU 2009103633 A	10/08/2010
		US 2008-0034172 A1	07/02/2008
		US 2008-0319997 A1	25/12/2008
		US 7434010 B2	07/10/2008
		US 7676638 B2	09/03/2010