

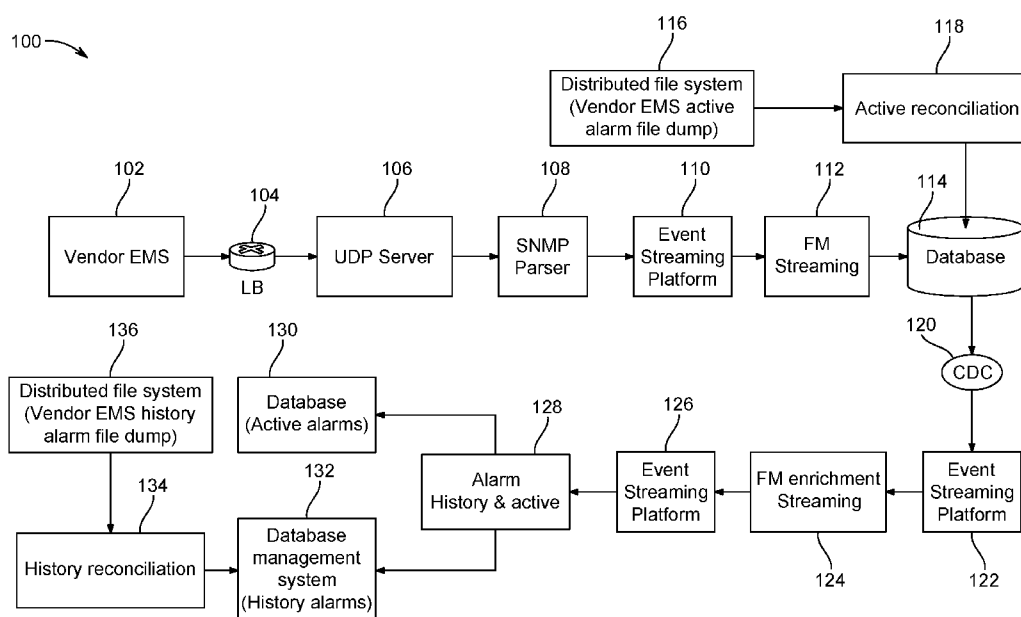


- (51) International Patent Classification:
G06F 16/23 (2019.01) G06F 16/27 (2019.01)
- (21) International Application Number:
PCT/IN2024/050638
- (22) International Filing Date:
30 May 2024 (30.05.2024)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
202321043760 29 June 2023 (29.06.2023) IN
- (71) Applicant: **JIO PLATFORMS LIMITED** [IN/IN]; Of-
fice-101, Saffron, Nr. Centre Point, Panchwati 5 Rasta, Am-
bawadi, Ahmedabad - 380006, Gujarat (IN).
- (72) Inventors: **BHATNAGAR, Aayush**; Tower-7, 15B, Bev-
erly Park, Sector-14, Koper Khairane, Navi Mumbai

- 400701, Maharashtra (IN). **BHATNAGAR, Pradeep Kumar**; Tower-7, 15B, Beverly Park, Sector-14, Koper Khairane, Navi Mumbai - 400701, Maharashtra (IN). **VENKATRAMAN, Rajeshwari**; C-104, Sterling Sharda Nivas, 15th Cross, 6th Main, Indiranagar 2nd Stage, Bangalore - 560038, Karnataka (IN). **PATEL, Himanshu**; A-1/93, Arjun Tower, Ghatlodia, Near C P Nagar, Ahmedabad - 380061, Gujarat (IN). **TIWARI, Shubham**; 41, Raj Samara Colony, Phase 2, J.K. Road, Bhopal 462021, Madhya Pradesh (IN). **BHARDWAJ, Avinash**; J-304, Golf City, P.N. 8, Sector 75, Noida - 201301, Uttar Pradesh (IN).
- (74) Agent: **D J, Solomon** et al.; De Penning & De Penning, 120 Velachery Main Road, Guindy, Chennai - 600032 (IN).
- (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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,

(54) Title: SYSTEM AND METHOD TO OVERCOME DATA RACE CONDITIONS IN A DATABASE

FIG. 1



(57) Abstract: The present disclosure relates to a method for mitigating data race conditions in a database (218). The method comprising requesting, by two or more data sources, one or more changes to be performed on a data stored in the database (218). The method comprising capturing the requested one or more changes in a streaming job to generate one or more alarms related to the one or more changes. The method comprising generating one or more logs related to the captured one or more changes. The method comprising determining a type of operation to be performed on the stored data based on the one or more generated logs and the one or more generated alarms. The method comprising responsive to determining, performing the type of operation on the data stored in the database (218).

HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, 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))*

SYSTEM AND METHOD TO OVERCOME DATA RACE CONDITIONS IN A DATABASE

RESERVATION OF RIGHTS

5 [0001] A portion of the disclosure of this patent document contains material,
which is subject to intellectual property rights such as but are not limited to,
copyright, design, trademark, integrated circuit (IC) layout design, and/or trade
dress protection, belonging to Jio Platforms Limited (JPL) or its affiliates
(hereinafter referred as owner). The owner has no objection to the facsimile
10 reproduction by anyone of the patent document or the patent disclosure, as it
appears in the Patent and Trademark Office patent files or records, but otherwise
reserves all rights whatsoever. All rights to such intellectual property are fully
reserved by the owner.

15 TECHNICAL FIELD

[0002] The present disclosure relates to the field of Database Management
Systems (DBMS) and data integration. More precisely, it relates to a system for an
automated Change Data Capture (CDC) mechanism to overcome a data race
conditions in a database.

20

BACKGROUND

[0003] Background description includes information that may be useful in
understanding the present invention. It is not an admission that any of the
information provided herein is prior art or relevant to the presently claimed
25 invention, or that any publication specifically or implicitly referenced is prior art.

[0004] The organizations may at times need to move data between different
database environments. For example, to create a backup of the data, or to enable
sharing of the data between different database applications. The data replication
systems help address this need, for example by detecting and replicating changes to
30 the data in a database table, as a result of row operations, rather than copying the

entire table and the data therein. The data replication systems can be used to synchronize the data in a target database with the data in a source database.

[0005] However, in the environments that support very large data sets, for example big data environments, present challenges related to availability, scalability, and fault-tolerance. The traditional databases or data replication systems may not scale sufficiently to handle such larger amounts of data.

[0006] Data race conditions happens whenever two processes update the database simultaneously. They occur when multiple threads or processes access shared data concurrently without proper synchronization, leading to unpredictable and erroneous behavior. Data race conditions can result in incorrect data insertion and sometimes leads to data corruptions. Data race conditions arise when at least two threads or processes perform simultaneous read and write operations on the same shared data, and at least one of the operations is a write. The exact interleaving and timing of these operations become unpredictable, potentially leading to inconsistent or unexpected results. Further, the data race conditions may lead to improper updating of the data.

[0007] However, the current techniques are inefficient in handling the race conditions in a database. Thus, there is a need for improved techniques that can overcome the race conditions in an effective manner.

20

OBJECTS OF INVENTION

[0008] Some of the objects of the present disclosure, that at least one embodiment herein satisfy are as listed herein below.

[0009] It is an object of the present disclosure to overcome the above limitations and drawbacks of the existing methods for using CDC to overcome data race conditions.

[0010] It is an object of the present disclosure to address data race conditions is to ensure data consistency in a concurrent environment.

[0011] It is an object of the present disclosure to enables synchronization of data changes across multiple systems or components.

30

[0012] It is an object of the present disclosure to help detect conflicts that arise when multiple threads or processes attempt to modify shared data concurrently.

5 [0013] It is an object of the present disclosure to capture and propagate data changes in near real-time and minimize the time window for potential race conditions to occur and swiftly propagate changes to ensure consistent and up-to-date data across systems.

[0014] It is an object of the present disclosure to enhance the scalability and performance of systems by reducing contention and improving parallel processing capabilities.

10 [0015] It is an object of the present disclosure to contribute to system reliability by providing fault-tolerant mechanisms for capturing and processing data changes.

[0016] It is an object of the present disclosure to simplify the development and maintenance of concurrent systems by providing a structured and standardized approach to handle data races.

15 [0017] It is an object of the present disclosure to align the objectives of using CDC with the specific requirements and challenges of addressing data race conditions to achieve improved data consistency, concurrency control, and overall reliability in the face of concurrent access to shared data.

SUMMARY

[0018] In an exemplary embodiment, the present invention discloses a method for mitigating data race conditions in a database. The method comprising requesting, by two or more data sources, one or more changes to be performed on a data stored in the database. The method comprising capturing the requested one or more changes in a streaming job to generate one or more alarms related to the one or more changes. The method comprising generating one or more logs related to the captured one or more changes. The method comprising determining a type of operation to be performed on the stored data based on the one or more generated

logs and the one or more generated alarms. The method comprising responsive to determining, performing the type of operation on the data stored in the database.

[0019] In an embodiment, the one or more changes are related to performing data manipulation operations on the data in the database.

5 [0020] In an embodiment, the data manipulation operations include at least one of an insert operation, an update operation, or a delete operation.

[0021] In an embodiment, the one or more alarms include messages indicating the one or more changes to be performed on the data in the database.

10 [0022] In an embodiment, the messages are stored in a distributed event streaming platform.

[0023] In an embodiment, the one or more alarms include at least one of a new alarm or a clear alarm.

[0024] In an embodiment, the new alarm indicates performing the insert operation on the data in the database.

15 [0025] In an embodiment, an insert query is created when the new alarm is generated.

[0026] In an embodiment, the clear alarm indicates performing the update operation on the data in the database.

20 [0027] In an embodiment, the update operation includes updating the data with a clear time as indicated in the clear alarm.

[0028] In an embodiment, the method further comprising storing the updated data with the clear time to a history table of a distributed computing framework.

25 [0029] In an exemplary embodiment, the present invention discloses a system for mitigating data race conditions in a database. The system comprising a receiving unit configured for receiving a request, from two or more data sources, for performing one or more changes on a data stored in the database. The receiving unit configured for capturing the requested one or more changes in a streaming job to generate one or more alarms related to the one or more changes. A processing
30 unit configured for generating one or more logs related to the captured one or more changes and determining a type of operation to be performed on the stored data

based on the one or more generated logs and the one or more generated alarms. The processing unit configured for responsive to determining, performing the type of operation on the data stored in the database.

5 [0030] In an embodiment, the one or more changes are related to performing data manipulation operations on the data in the database.

[0031] In an embodiment, the data manipulation operations include at least one of an insert operation, an update operation, or a delete operation.

[0032] In an embodiment, the one or more alarms include messages indicating the one or more changes to be performed on the data in the database.

10 [0033] In an embodiment, the messages are stored in a distributed event streaming platform.

[0034] In an embodiment, the one or more alarms include at least one of a new alarm or a clear alarm.

15 [0035] In an embodiment, the new alarm indicates performing the insert operation on the data in the database.

[0036] In an embodiment, an insert query is created when the new alarm is generated.

[0037] In an embodiment, the clear alarm indicates performing the update operation on the data in the database.

20 [0038] In an embodiment, the update operation includes updating the data with a clear time as indicated in the clear alarm.

[0039] In an embodiment, the system further configured for storing the updated data with the clear time to a history table of a distributed computing framework.

25

BRIEF DESCRIPTION OF DRAWINGS

[0040] The specifications of the present disclosure are accompanied with drawings of the system and method to aid in better understanding of the said invention. The drawings are in no way limitations of the present disclosure, rather
30 are meant to illustrate the ideal embodiments of the said disclosure.

[0041] In the figures, similar components and/or features may have the same reference label. Further, various components of the same type may be distinguished by following the reference label with a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

[0042] FIG. 1 illustrates a block diagram of the entire system with individual components for change data capture (CDC) to overcome data race conditions, in accordance with an embodiment of the present disclosure.

10 [0043] FIG. 2 illustrates an exemplary block diagram of all the modules of the system, in accordance with an embodiment of the present disclosure.

[0044] FIG. 3 illustrates an exemplary flow structure of a sample data race condition, in accordance with an embodiment of present disclosure.

15 [0045] FIG. 4 illustrates an exemplary flow diagram of a sample CDC operation, in accordance with an embodiment of the present disclosure.

[0046] FIG. 5 illustrates a block diagram of a sample CDC operation, in accordance with an embodiment of the present disclosure.

20 [0047] FIG. 6 illustrates an exemplary computer system in which or with which embodiments of the present invention can be utilized, in accordance with an embodiment of present disclosure.

[0048] FIG. 7 illustrates a flow diagram of a method for mitigating data race conditions in a database, in accordance with an embodiment of the present disclosure.

25 LIST OF REFERENCE NUMERALS

100 – Block diagram

200 – Exemplary block diagram of a system

202- Receiving unit

30 204-Memory

- 206- Interfacing unit
- 208- Processing unit
- 210-Log module
- 212-Change data capture (CDC) module
- 5 214- Synchronization module
- 216-Other modules
- 218-Database
- 300 – Flow structure
- 400- Flow diagram
- 10 500- Block diagram
- 600- A computer system
- 602 – Input devices
- 604- Central processing unit (CPU)
- 608-Output devices
- 15 610-Secondary storage devices
- 612-Control unit
- 614-Arithmetic & Logical unit
- 700- Flow Diagram

20 **DETAILED DESCRIPTION**

- [0049] In the following description, for explanation, various specific details are outlined in order to provide a thorough understanding of embodiments of the present disclosure. It will be apparent, however, that embodiments of the present disclosure may be practiced without these specific details. Several features
- 25 described hereafter can each be used independently of one another or with any

combination of other features. An individual feature may not address all of the problems discussed above or might address only some of the problems discussed above. Some of the problems discussed above might not be fully addressed by any of the features described herein.

5 **[0050]** The ensuing description provides exemplary embodiments only and is not intended to limit the scope, applicability, or configuration of the disclosure. Rather, the ensuing description of the exemplary embodiments will provide those skilled in the art with an enabling description for implementing an exemplary embodiment. It should be understood that various changes may be made in the
10 function and arrangement of elements without departing from the spirit and scope of the disclosure as set forth.

[0051] Specific details are given in the following description to provide a thorough understanding of the embodiments. However, it will be understood by one of ordinary skill in the art that the embodiments may be practiced without these
15 specific details. For example, circuits, systems, networks, processes, and other components may be shown as components in block diagram form in order not to obscure the embodiments in unnecessary detail. In other instances, well-known circuits, processes, algorithms, structures, and techniques may be shown without unnecessary detail to avoid obscuring the embodiments.

20 **[0052]** Also, it is noted that individual embodiments may be described as a process that is depicted as a flowchart, a flow diagram, a data flow diagram, a structure diagram, or a block diagram. Although a flowchart may describe the operations as a sequential process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be re-arranged.
25 A process is terminated when its operations are completed but could have additional steps not included in a figure. A process may correspond to a method, a function, a procedure, a subroutine, a subprogram, etc. When a process corresponds to a function, its termination can correspond to a return of the function to the calling function or the main function.

30 **[0053]** The word “exemplary” and/or “demonstrative” is used herein to mean serving as an example, instance, or illustration. For the avoidance of doubt,

the subject matter disclosed herein is not limited by such examples. In addition, any aspect or design described herein as “exemplary” and/or “demonstrative” is not necessarily to be construed as preferred or advantageous over other aspects or designs, nor is it meant to preclude equivalent exemplary structures and techniques
5 known to those of ordinary skill in the art. Furthermore, to the extent that the terms “includes,” “has,” “contains,” and other similar words are used in either the detailed description or the claims, such terms are intended to be inclusive like the term “comprising” as an open transition word without precluding any additional or other elements.

10 **[0054]** Reference throughout this specification to “one embodiment” or “an embodiment” or “an instance” or “one instance” means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the present disclosure. Thus, the appearances of the phrases “in one embodiment” or “in an embodiment” in various places throughout
15 this specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

[0055] The terminology used herein is to describe particular embodiments only and is not intended to be limiting the disclosure. As used herein, the singular
20 forms “a”, “an”, and “the” are intended to include the plural forms as well, unless the context indicates otherwise. It will be further understood that the terms “comprises” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other
25 features, integers, steps, operations, elements, components, and/or groups thereof. As used herein, the term “and/or” includes any combinations of one or more of the associated listed items.

[0056] The present invention provides improved techniques that can overcome the race conditions in an effective manner. The present invention employs a
30 change data capture (CDC) technique that is used to identify and capture changes made to data in a database or data source. In an aspect, the CDC captures and

records all data modifications, including inserts, updates, and deletes, as they occur, enabling real-time or near-real-time synchronization and replication of data between systems. The CDC employs an open-source distributed platform captures changes in data from the streaming data and flags them as “insert, modify and delete” for further processing. The distributed platform connects to the database’s, usually Structured Query Language (SQL) databases, transaction log or replication log, depending on the supported database system. The distributed platform reads the log in a non-intrusive manner, without impacting the performance or integrity of the database. The distributed platform extracts the captured changes from the database log, including inserts, updates, and deletes. The distributed platform translates these changes into a standardized representation of events. The extracted change events are serialized into a specific format, such as JSON or Avro. The serialization enables the events to be easily transmitted and consumed by downstream systems.

15 [0057] The distributed platform distributes the serialized change events to downstream systems using an open-source distributed streaming platform that acts as a scalable and fault-tolerant platform, ensuring reliable delivery of the change events to consumers.

[0058] In an aspect, the present disclosure employs the CDC mechanism with the open source distributed platform to achieve a real-time, event-driven architecture where changes in databases are efficiently captured, converted, and propagated for downstream consumption. Thus, with the present disclosure changes can be made in a database to create, read, update and delete data and to overcome any race conditions due to which data may not get updated in the database properly.

25 [0059] The various embodiments throughout the disclosure will be explained in more detail with reference to FIGs. 1-7.

[0060] The present disclosure relates to the field of Database Management Systems (DBMS) and data integration. More precisely, it relates to a system for an automated Change Data Capture (CDC) mechanism to overcome a data race condition.

30

[0061] Fig. 1 presents a schematic of a system (100) designed for the mitigation of data race conditions by employing the methodology of CDC, in accordance with one embodiment of the present disclosure. The depiction is aligned with the current embodiment, where the system (100) integrates a multitude of components to facilitate the real-time synchronization and processing of data modifications emanating from diverse sources.

[0062] According to the configuration, the system (100) includes an element management system (vendor) (EMS) (102). The EMS (102) executes the transmission of feature data using a Location Based Service (104) alongside a User Datagram Protocol (UDP) Server (106). The Vendor EMS (102) is configured for dispatching data pertinent to the manufacturing process, such as telemetry from machinery or production metrics to central data processing facilities.

[0063] Subsequent to transmission, the Simple Network Management Protocol (SNMP) Parser (108) analyses the data. The SNMP parser (108) is a component that is configured to interpret and analyze SNMP messages or packets. The SNMP parser is a protocol used for network management and monitoring, allowing network administrators to manage devices and monitor their performance. The SNMP parser typically takes raw SNMP messages or packets as input and parses them to extract relevant information such as device status, performance metrics, or configuration data. This parsed information can then be processed, displayed, or used for various network management tasks.

[0064] The SNMP Parser (108), in accordance with the embodiment, operates as a diagnostic tool that deciphers and categorizes data, similar to how a sensor array interprets various stimuli. For example, it might categorize data packets based on urgency or type of data, such as distinguishing between normal operational data and error messages.

[0065] The processed information is then relayed to the first Event Streaming Platform (110), which includes integral components, such as a second event streaming platform (122) and a third event streaming platform (126).

[0066] In conjunction, the Distributed File System (116) serves as the data archival system connected to the Event Streaming Platform (110). The Distributed

File System (116) acts as a vast library, archiving vast amounts of data for future reference or analysis. In practical terms, it could store historical production data for trend analysis or maintain logs for compliance purposes.

5 [0067] In one embodiment, the system (100) includes a fault management (FM) Streaming module (112) for establishing a connection with a Database (114). The FM streaming module (112) implements a distributed computational task to efficiently handle events related to data changes and alarm management. Working alongside the Database (114) is an active reconciliation component (118), operational through a data processing task. The active reconciliation component 10 (118) is configured for quality assurance process, verifying the accuracy and consistency of data after it undergoes changes. By utilizing distributed data processing capabilities, the active reconciliation component (118) can, for example, ensure that transaction records from multiple retail locations are harmonized and accurately reflected in a central inventory system.

15 [0068] The system (100) further includes the CDC module (120) for capturing and coordinating data operations to avert race conditions. The CDC module (120) conducts continuous surveillance over the database (114), much like a surveillance system overseeing a secured facility, to log any alterations within the data, ensuring that all transactions are recorded, and no conflicting operations occur.

20 [0069] The CDC module (120) transmits the captured data to an open source distributed streaming platform which aggregates the data from various sources. The open source distributed streaming platform (122) aggregates the data, ensuring its readiness for further action by another event streaming platform (126). The FM enrichment streaming (124) implements a distributed computational task to 25 efficiently handle events related to data changes.

[0070] In addition, the system (100) is equipped with an Alarm History & Active component (128) interfacing with a database (130), a database dedicated to the supervision of active alarms. The database (130) can be likened to a dynamic ledger, capable of recording new transactions such as alarm activations and 30 updating existing ones when an alarm is resolved. For example, when a security

breach is detected and then neutralized, the database (130) logs the event and updates the status accordingly.

[0071] For the archival of historical alarm data, a column-oriented, non-relational database management system (132) is utilized, coupled with a distributed
5 file system (136). This pairing functions similarly to a museum archive, where past exhibits, in this case, alarm histories, are catalogued and stored for retrospective examination or compliance checks, facilitated by a Historical Data Reconciliation component (134) that uses a distributed computational task.

[0072] The CDC module (120) is also scalable, designed to adapt to fluctuating
10 data volumes by horizontally scaling, which could involve adding more servers to handle increased loads.

[0073] FIG. 2 provides an exemplary system (100) configured to manage and synchronize data alterations within a database environment, in accordance with one embodiment.

[0074] The system (100) includes, but may not be limited to, receiving unit
15 (202), a memory (204), an interfacing unit (206), a processing unit (208), and a database (218). The processing unit (208) further comprises a Log module (210), a change data capture (CDC) module (212), a Synchronization module (214), and various other modules (216) which execute numerous processes. The modules are
20 controlled by the processing unit (208) which execute instructions retrieved from the memory (204). The processing unit (208) further interact with the interfacing unit (206) to facilitate user interaction and to provide options for managing and configuring the system (100). The processing unit (208) may be implemented as
25 one or more microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, logic circuitries, and/or any devices that process data based on operational instructions. Among other capabilities, the processing unit (208) may be configured to fetch and execute computer-readable instructions stored in a memory (204) of the system (100). The memory (204) may
30 be configured to store one or more computer-readable instructions or routines in a non-transitory computer readable storage medium. The memory (204) may comprise any non-transitory storage device including, for example, volatile

memory such as random-access memory (RAM), or non-volatile memory such as erasable programmable read only memory (EPROM), flash memory, and the like.

[0075] In an embodiment, the interfacing unit (206) may comprise a variety of interfaces, for example, interfaces for data input and output devices (I/O), storage devices, and the like. The interfacing unit (206) may facilitate communication through the system (100). The interfacing unit (206) may also provide a communication pathway for various other units/modules (216) of the system (100).

[0076] The log module (210) is configured for addressing potential data race conditions by capturing and maintaining a consistent and reliable record of data changes. The log module (210) employs a Write-Ahead Logging strategy, which ensures changes are recorded in a log before being applied to the database (218). The processing unit (208) supports integration with database (218) allowing the database (218) to capture changes from diverse data sources. The database (218) may include relational databases, NoSQL databases, or message queues. The processing unit (208) is capable of capturing and monitoring changes made to the source data in real-time or near real-time. The processing unit (208) also detects and extracts the individual data modifications, including inserts, updates, and deletes.

[0077] In an embodiment, the log module (210) is configured to overcome data race conditions and ensure reliable and accurate capture of data changes. The log module (210) is implemented for providing durability, atomicity, and consistency during the change capture process. The log module (210) ensures atomicity and durability of data changes which further ensures that a transaction's changes are either entirely committed or entirely rolled back. The log module (210) prevents partial or inconsistent updates to the shared data and avoids data race conditions resulting from incomplete transactions. The log module (210) also follows a Write-Ahead Logging strategy, where changes are first written to the log before being applied to the database. This provides a reliable record of the changes and helps in recovering from failures or crashes. In the event of a failure, the CDC module (212) can use the log to recover and reapply the captured changes, ensuring data integrity. The log module (210) may employ synchronization mechanisms, such as locks or

semaphores, to ensure exclusive access to the log during write operations. This prevents multiple threads from concurrently writing to the log, preventing data race conditions. The log module (210) provides the necessary synchronization, atomicity, and durability guarantees, reducing the chances of data race conditions and ensuring reliable and accurate change capture.

[0078] The database (218) offers functionality to manage the capture, storage, and retrieval of data changes. The database (218) employs various transaction isolation levels and locking mechanisms to regulate access to shared data, ensuring that operations, such as inserts, updates, and deletes are executed in a controlled manner. The database (218) also contributes to managing the capture, storage, and retrieval of data changes in a concurrent and synchronized manner. The database (218) provides appropriate transaction isolation levels to ensure that concurrent transactions do not interfere with each other. Isolation levels like Serializable or Repeatable Read can be implemented to prevent data races by providing consistency and preventing dirty reads, non-repeatable reads, and phantom reads. The database (218) implements a locking and concurrency control mechanisms to regulate access to shared data. By acquiring locks on relevant resources, the database (218) can ensure exclusive access during data capture and prevent concurrent modifications that may lead to data races. The database (218) can provide mechanisms for change tracking and logging. This can be achieved through transaction logs, change tables, or triggers that capture data modifications at the source database. These logs and change tables serve as reliable sources of captured changes and help overcome data race conditions by providing an ordered and accurate record of modifications.

[0079] In an embodiment, the CDC module (212) is configured for avoiding the race condition in the database (218) when two or more sources are trying to access the data in the database (218). In an embodiment, to avoid this race condition, a CDC architecture is used which refers to the process of identifying and capturing changes made to the data in the database (218) and then delivering those changes in real-time to a downstream process/system.

[0080] In an embodiment, a synchronization module (214) is configured for implementing a synchronization to facilitate a concurrent access to shared resources. It implements locks, such as mutexes or semaphores, that can be utilized to achieve mutual exclusion and synchronize access to shared resources. By acquiring locks before accessing critical sections of code or shared data, threads can ensure that only one thread at a time can perform write operations, preventing data race conditions. Atomic operations provide a way to perform operations on shared resources in an indivisible and thread-safe manner. These operations ensures that no other thread can access the shared resource simultaneously, preventing race conditions. Examples include atomic variables or compare-and-swap instructions. The synchronization module (214) also utilizes database transaction mechanisms that can help overcome data race conditions. By encapsulating multiple operations within a transaction, the database ensures atomicity and isolation. The synchronization module (214) allows for consistency by providing a well-defined commit point, ensuring that either all changes within a transaction are applied, or none of them are. The other executing modules in the processing unit (208) are used for all other executing processes in the system.

[0081] The modules within the processing unit (208) are configured to integrate various types of source databases (218), including relational databases, NoSQL databases, or message queues. Thus, the system (100) is enabled to monitor and capture data changes in real-time or near real-time from a wide array of data sources, reflecting the system's versatility and adaptability in different database environments.

[0082] FIG. 3 illustrates a flow diagram (300) of an exemplary flow structure of a sample data race condition, in accordance with an embodiment of present disclosure.

[0083] The race condition manifests when two separate threads (302, 312) engage with a common resource concurrently and perform write operations simultaneously.

[0084] In the depicted sequence, each thread attempts to increment a shared numerical counter. Initially, Thread 1 (302) reads the counter, observing a starting

value of 1 (304). Subsequently, it increments this value (306), and writes the incremented value, which is now 2, back to the shared counter (308), before completing its cycle (310). Parallel to this, Thread 2 (312) executes an analogous set of operations, reading the same initial value of 1 (314), incrementing it (316),
5 and then writing back the new value of 2 to the shared counter (318), culminating its process (320).

[0085] The concurrent execution of these threads without appropriate synchronization leads to a data race condition, whereby the final state of the counter may not accurately reflect the total increments performed by both threads. The
10 absence of a prescribed execution order for Thread 1 (302) and Thread 2 (312) renders the end result of the counter's value indeterminate, varying with each execution due to potential mutual interference.

[0086] To circumvent such race conditions, the implementation of synchronization controls is imperative. These controls, such as locking mechanisms
15 or atomic operations, ensure singular access to the shared counter at any given time, thereby allowing one thread to complete its incrementation before the other commences. Effective synchronization ensures orderly and sequential increments, thereby preserving the integrity of the counter's value following concurrent operations by Thread 1 (302) and Thread 2 (312).

20 [0087] The flowchart (300) depicts a process of synchronization in multithreaded systems and the challenges that arise when such controls are not in place. The process emphasizes the necessity of mechanisms that enforce exclusive access to shared resources to maintain consistent and error-free operations in programs with concurrent processes.

25 [0089] FIG. 4 offers a schematic representation (400) of the operational workflow within a Change Data Capture (CDC) system, specifically highlighting a CDC mechanism (120) in action. This diagram elucidates the step-by-step process that occurs once a database operation is initiated.

[0090] The workflow initiates when messages is passed through a database,
30 marked by the start of the process (402). The workflow can involve a variety of database operations, but for illustrative purposes, consider an insert query that is

created (404). The workflow represents a new entry being added to a database, such as a new customer record in a sales database.

[0091] Upon the creation of this insert query, the CDC mechanism (120), here represented as a part of the process flow (406), reads the logs of the database. It is responsible for identifying and executing related database operations based on the type of query detected. For example, the CDC mechanism (120) may note that a new sales transaction has occurred and will thus capture this change.

[0092] Subsequently, the captured data is passed into an event streaming platform (408) configured to facilitate real-time processing and analysis of event data, ensuring that any change or event is immediately available for necessary action or response. In the context of the sales database, the event streaming platform (408) would enable the real-time tracking of sales transactions as they occur.

[0093] Once the data is within the event streaming platform, the CDC mechanism (120) continues to monitor and react to the data. If the data corresponds to an insert operation (410), the system may raise a new alarm (412). This alarm could serve as a notification for stakeholders, indicating that a new sales transaction has been processed and may require further action, such as inventory updates or order fulfilment.

[0094] If an active alarm has to be discontinued, the CDC mechanism (120) ensures that the record is updated with the clear time (414), thereby keeping the system's records current and accurate. Completing the cycle, the alarm records with the updated clear time are then moved into a history table (416), serving as an archive for all processed events.

[0095] This comprehensive workflow demonstrates the CDC mechanism's (120) ability to capture and propagate changes throughout the system, allowing applications to respond to these changes in real-time. It enables data synchronization across multiple systems and maintains a historical record of changes, thus supporting various data analysis tasks. The workflow's systematic approach ensures that all features, as claimed, are supported literally, with the interconnectivity of each feature clearly illustrated through the flow diagram (400).

[0096] FIG. 5 presents a structured depiction (500) of a Change Data Capture (CDC) system operation according to an embodiment of the current disclosure. This block diagram delineates the flow of data from its origination to its final destination, facilitated by CDC (504).

5 [0097] At the source (502), new data is generated. This could be transactional records in a database from a retail sales system, where each transaction represents a new entry. As the data is generated, it is captured by the CDC mechanism (504), which is designed to monitor and log changes in the source (502).

[0098] This CDC mechanism (504) is a trigger-based system configured to act
10 upon insert, update, and delete operations performed on the data. When an operation is conducted on the source data, such as a new sales transaction being entered the trigger is activated, capturing the details of this operation.

[0099] The captured information is then published and made available for
subscription and consumption by integration processes (506-1, 506-2). These
15 integration processes may represent various applications or systems within an enterprise that rely on real-time data, such as inventory management systems or customer relationship management software. For instance, integration process 1 (506-1) might update an inventory database to reflect a sale, reducing the stock level of the sold item. Simultaneously, integration process 2 (506-2) could update a
20 customer's purchase history in a separate system.

[0100] The targets (508-1, 508-2) represent the final repositories for the
processed data. These could be databases in a data warehouse where comprehensive records are kept for analytical purposes or data lakes where raw data is stored for future processing. Target 1 (508-1) might store a detailed transaction log for
25 financial auditing, while target 2 (508-2) could hold customer behavior data for marketing analysis.

[0101] The system's ability to capture and integrate changes in real-time ensures that the data in targets (508-1, 508-2) is always current and reflective of the latest state of the source (502). This immediate reflection of changes allows
30 organizations to make informed decisions based on the most recent data, avoiding

the consequences of outdated information which could lead to missed opportunities or business errors.

[0102] Each component within this system (500) is interconnected, ensuring that data flows seamlessly from the point of creation to the point of utilization. The CDC mechanism (504) acts as a central hub in this process, guaranteeing that every change at the source (502) is tracked and mirrored across all systems and platforms that depend on this data.

[0103] FIG. 6 illustrates an exemplary computer system in which or with which embodiments of the present invention can be utilized, in accordance with an embodiment of present disclosure.

[0104] Referring to FIG. 6, a block diagram of an exemplary computer system is disclosed. The computer system includes input devices (602) connected through I/O peripherals. The system also includes a Central Processing Unit (CPU) (604), and Output Devices (608), connected through the I/O peripherals. The CPU (604) is also attached to a memory unit 616 along with an Arithmetic and Logical Unit (ALU) (614), a control unit (612), along with secondary storage devices (610) such as Hard Disks and a Secure Digital Card (SD). The data flow and control flow (606) is indicated by a straight and dashed arrow respectively. The CPU consists of data registers that hold the data bits, pointers, cache, Random Access Memory (RAM) (204), and a main processing unit containing the processing unit (208). The system also consists of communication buses that are used to transport the data internally in the system.

[0105] FIG. 7 is a flowchart depicting a method for mitigating data race conditions in a database system.

[0106] At step 702, the method comprising requesting, by two or more data sources, one or more changes to be performed on a data stored in the database. In an aspect, the data sources may include a streaming data source, or a file based data source. The streaming data sources are a continuous and real-time provider of data that emits data records over the time. The streaming data sources generate data continuously and produce large volumes of data at high frequencies. In an aspect, a file-based data source refers to a data source where data is stored and organized in

files on a file system. These files can contain structured or unstructured data and are typically stored in formats such as text files, comma separated values (CSV) files, JavaScript object notation (JSON) files, extensible markup language (XML) files etc.

5 **[0107]** At step 704, the method comprising capturing the requested one or more changes in a streaming job to generate one or more alarms related to the one or more changes.

[0108] At step 706, the method comprising generating one or more logs related to the captured one or more changes.

10 **[0109]** At step 708, the method comprising determining a type of operation to be performed on the stored data based on the one or more generated logs and the one or more generated alarms.

[0110] At step 710, the method comprising responsive to determining, performing the type of operation on the data stored in the database.

15 **[0111]** In an embodiment, the one or more changes are related to performing data manipulation operations on the data in the database.

[0112] In an embodiment, the data manipulation operations include at least one of an insert operation, an update operation, or a delete operation.

[0113] In an embodiment, the one or more alarms include messages
20 indicating the one or more changes to be performed on the data in the database.

[0114] In an embodiment, the messages are stored in a distributed event streaming platform.

[0115] In an embodiment, the one or more alarms include at least one of a new alarm or a clear alarm.

25 **[0116]** In an embodiment, the new alarm indicates performing the insert operation on the data in the database.

[0117] In an embodiment, an insert query is created when the new alarm is generated.

[0118] In an embodiment, the clear alarm indicates performing the update
30 operation on the data in the database.

[0119] In an embodiment, the update operation includes updating the data with a clear time as indicated in the clear alarm.

[0120] In an embodiment, the method further comprising storing the updated data with the clear time to a history table of a distributed computing
5 framework.

[0121] In an exemplary embodiment, the present invention discloses a system for mitigating data race conditions in a database. The system comprising a receiving unit configured for receiving a request, from two or more data sources, for performing one or more changes on a data stored in the database. The receiving
10 unit configured for capturing the requested one or more changes in a streaming job to generate one or more alarms related to the one or more changes. A processing unit configured for generating one or more logs related to the captured one or more changes and determining a type of operation to be performed on the stored data based on the one or more generated logs and the one or more generated alarms. The
15 processing unit configured for responsive to determining, performing the type of operation on the data stored in the database.

[0122] In an embodiment, the one or more changes are related to performing data manipulation operations on the data in the database.

[0123] In an embodiment, the data manipulation operations include at least
20 one of an insert operation, an update operation, or a delete operation.

[0124] In an embodiment, the one or more alarms include messages indicating the one or more changes to be performed on the data in the database.

[0125] In an embodiment, the messages are stored in a distributed event streaming platform.

[0126] In an embodiment, the one or more alarms include at least one of a
25 new alarm or a clear alarm.

[0127] In an embodiment, the new alarm indicates performing the insert operation on the data in the database.

[0128] In an embodiment, an insert query is created when the new alarm is
30 generated.

[0129] In an embodiment, the clear alarm indicates performing the update operation on the data in the database.

[0130] In an embodiment, the update operation includes updating the data with a clear time as indicated in the clear alarm.

5 [0131] In an embodiment, the system further configured for storing the updated data with the clear time to a history table of a distributed computing framework.

[0132] Further, if one source, deletes one of the records and second source is trying to add the record, the present invention will create two entries, one deletion
10 and one insertion for the same record. Thus, by comparing the two entries for the same record and ignoring the insertion of the record as it has been deleted.

[0133] It is to be appreciated by a person skilled in the art that while various embodiments of the present disclosure have been elaborated for using CDC (120) to overcome data race conditions. However, the teachings of the present disclosure
15 are also applicable for other types of applications as well, and all such embodiments are well within the scope of the present disclosure. However, the system and method for sign language conversion is also equally implementable in other industries as well, and all such embodiments are well within the scope of the present disclosure without any limitation.

20 [0134] Moreover, in interpreting the specification, all terms should be interpreted in the broadest possible manner consistent with the context. In particular, the terms “comprises” and “comprising” should be interpreted as referring to elements, components, or steps in a non-exclusive manner, indicating that the referenced elements, components, or steps may be present, or utilized, or
25 combined with other elements, components, or steps that are not expressly referenced. Where the specification claims refer to at least one of something selected from the group consisting of A, B, C....and N, the text should be interpreted as requiring only one element from the group, not A plus N, or B plus N, etc.

30 [0135] While considerable emphasis has been placed herein on the preferred embodiments it will be appreciated that many embodiments can be made and that

many changes can be made in the preferred embodiments without departing from the principles of the disclosure. These and other changes in the preferred embodiments of the disclosure will be apparent to those skilled in the art from the disclosure herein, whereby it is to be distinctly understood that the foregoing
5 descriptive matter is to be implemented merely as illustrative of the disclosure and not as a limitation.

ADVANTAGES OF THE INVENTION

[0136] The proposed invention provides a system for CDC for overcoming data
10 race conditions.

[0137] The proposed invention provides a system that eliminates the risk of concurrent conflicting operations that can lead to data inconsistencies or incorrect outcomes, providing reliable and accurate data capture.

[0138] The proposed invention provides a system that mitigates the impact of
15 failures, such as system crashes or network disruptions, by ensuring proper sequencing of operations and enabling recovery from unexpected events.

[0139] The proposed invention provides a system that supports concurrent access to shared resources, allowing for improved concurrency and scalability.

[0140] The proposed invention provides a system that eliminates unnecessary
20 retries, rollbacks or duplicated operations that can waste computational resources.

[0141] The proposed invention provides a system that overcomes data race conditions which allows for real-time or near real-time data capture.

[0142] The proposed invention provides a system that provides clear guidelines and mechanisms for handling data race conditions in CDC processes.

[0143] The proposed invention provides a system that integrates data from
25 different sources or propagating changes to downstream systems, ensuring synchronized and coherent data flow.

[0144] The proposed invention provides a system that can handle growing workloads and adapt to changing environments, providing a scalable and adaptable
30 solution for capturing data changes.

CLAIMS

We claim:

1. A method (700) for mitigating data race conditions in a database (218), the
5 method (700) comprising:
 - requesting (702), by two or more data sources, one or more changes to
be performed on a data stored in the database (218);
 - capturing (704) the requested one or more changes in a streaming job
to generate one or more alarms related to the one or more changes;
 - 10 generating (706) one or more logs related to the captured one or more
changes;
 - determining (708) a type of operation to be performed on the stored data
based on the one or more generated logs and the one or more generated
alarms; and
 - 15 responsive to determining, performing (710) the type of operation on
the data stored in the database (218).
2. The method as claimed in claim 1, wherein the one or more changes are related
to performing data manipulation operations on the data in the database (218).
- 20 3. The method as claimed in claim 2, wherein the data manipulation operations
include at least one of an insert operation, an update operation, or a delete operation.
4. The method as claimed in claim 1, wherein the one or more alarms include
25 messages indicating the one or more changes to be performed on the data in the
database (218).
5. The method as claimed in claim 4, wherein the messages are stored in a
distributed event streaming platform.

30

6. The method as claimed in claim 1, wherein the one or more alarms include at least one of a new alarm or a clear alarm.
7. The method as claimed in claim 6, wherein the new alarm indicates performing the insert operation on the data in the database (218).
8. The method as claimed in claim 7, wherein an insert query is created when the new alarm is generated.
9. The method as claimed in claim 6, wherein the clear alarm indicates performing the update operation on the data in the database (218).
10. The method as claimed in claim 6, wherein the update operation includes updating the data with a clear time as indicated in the clear alarm.
11. The method as claimed in claim 10, further comprising storing the updated data with the clear time to a history table of a distributed computing framework.
12. A system (100) for mitigating data race conditions in a database (218), the system (100) comprising:
- a receiving unit (202) configured for:
 - receiving a request, from two or more data sources, for performing one or more changes on a data stored in the database (218); and
 - capturing the requested one or more changes in a streaming job to generate one or more alarms related to the one or more changes;
 - a processing unit (208) configured for:
 - generating one or more logs related to the captured one or more changes;
 - determining a type of operation to be performed on the stored data based on the one or more generated logs and the one or more generated alarms; and

responsive to determining, performing the type of operation on the data stored in the database (218).

13. The system (100) as claimed in claim 12, wherein the one or more changes are
5 related to performing data manipulation operations on the data in the database (218).

14. The system (100) as claimed in claim 13, wherein the data manipulation
operations include at least one of an insert operation, an update operation, or a delete
10 operation.

15. The system (100) as claimed in claim 12, wherein the one or more alarms
include messages indicating the one or more changes to be performed on the data
in the database (218).

16. The system (100) as claimed in claim 15, wherein the messages are stored in a
distributed event streaming platform.

17. The system (100) as claimed in claim 12, wherein the one or more alarms
20 include at least one of a new alarm or a clear alarm.

18. The system (100) as claimed in claim 17, wherein the new alarm indicates
performing the insert operation on the data in the database (218).

25 19. The system (100) as claimed in claim 18, wherein an insert query is created
when the new alarm is generated.

20. The system (100) as claimed in claim 17, wherein the clear alarm indicates
performing the update operation on the data in the database (218).

30

21. The system (100) as claimed in claim 17, wherein the update operation includes updating the data with a clear time as indicated in the clear alarm.

22. The system (100) as claimed in claim 21, further comprising storing the updated
5 data with the clear time to a history table of a distributed computing framework.

23. A computer program product comprising a non-transitory computer-readable
medium comprising instructions that, when executed by one or more processors,
cause the one or more processors to perform a method (700) for mitigating data race
10 conditions in a database (218), the method (700) comprising:

requesting (702), by two or more data sources, one or more changes to
be performed on a data stored in the database (218);

capturing (704) the requested one or more changes in a streaming job
to generate one or more alarms related to the one or more changes;

15 generating (706) one or more logs related to the captured one or more
changes;

determining (708) a type of operation to be performed on the stored data
based on the one or more generated logs and the one or more generated
alarms; and

20 responsive to determining, performing (710) the type of operation on
the data stored in the database (218).

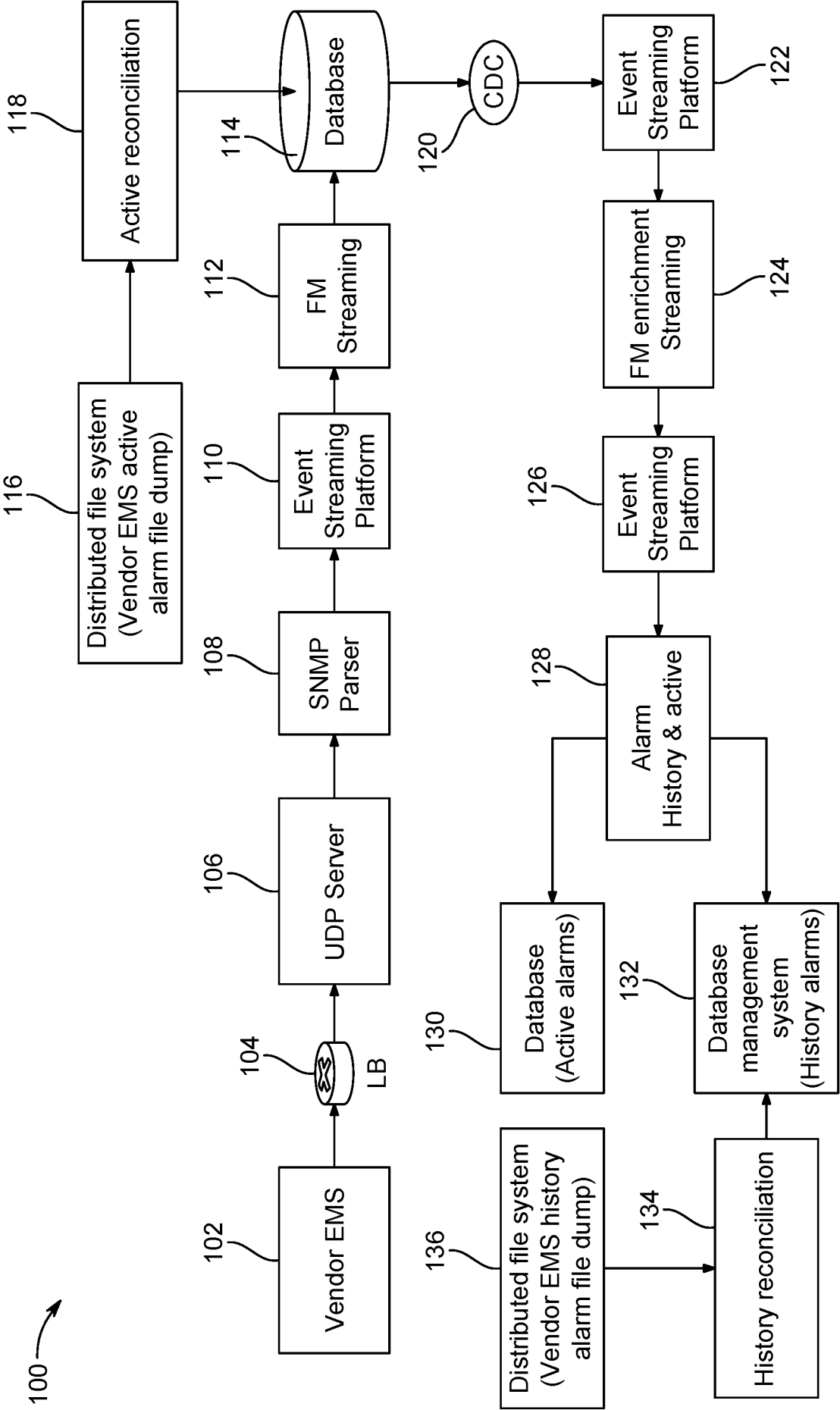


FIG. 1

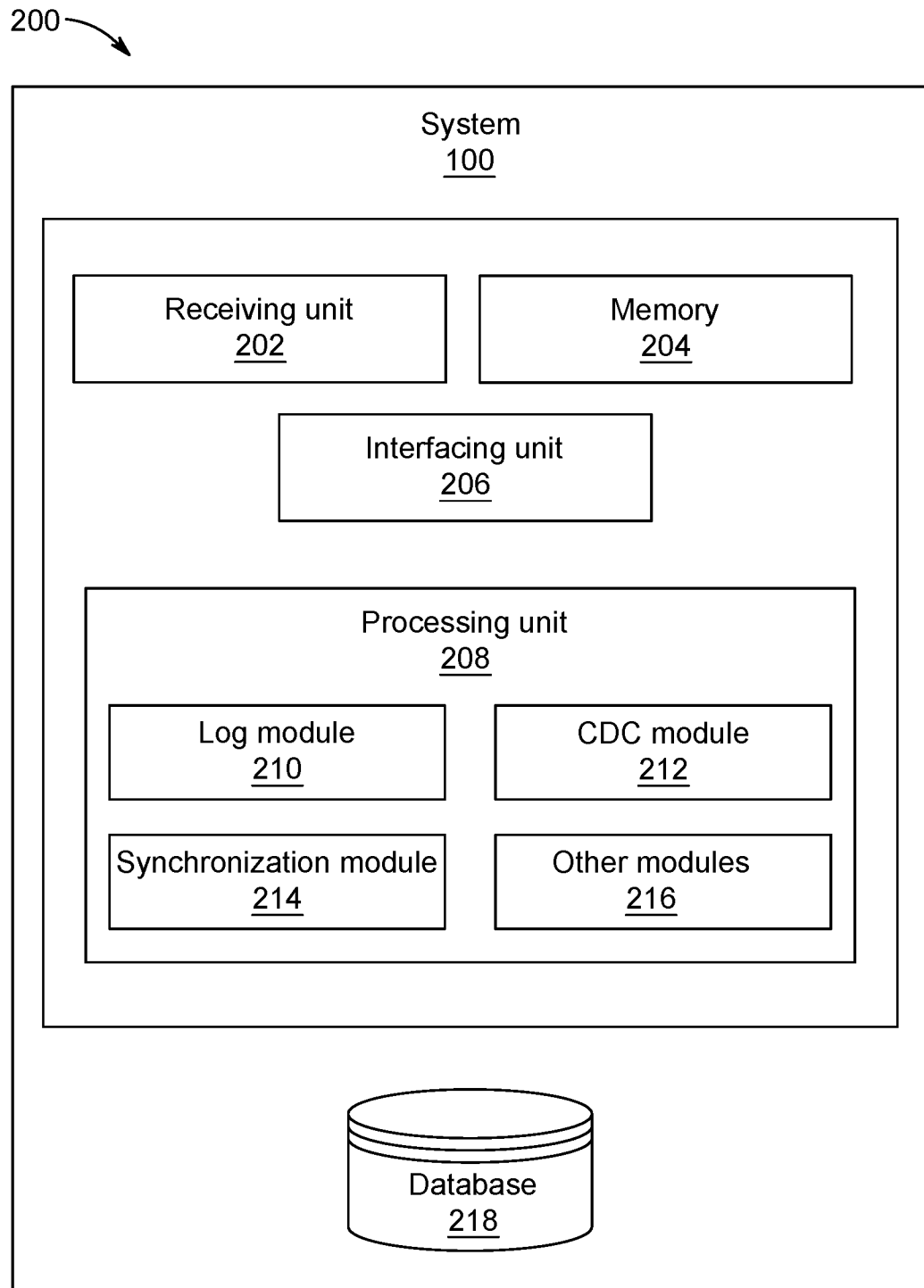


FIG. 2

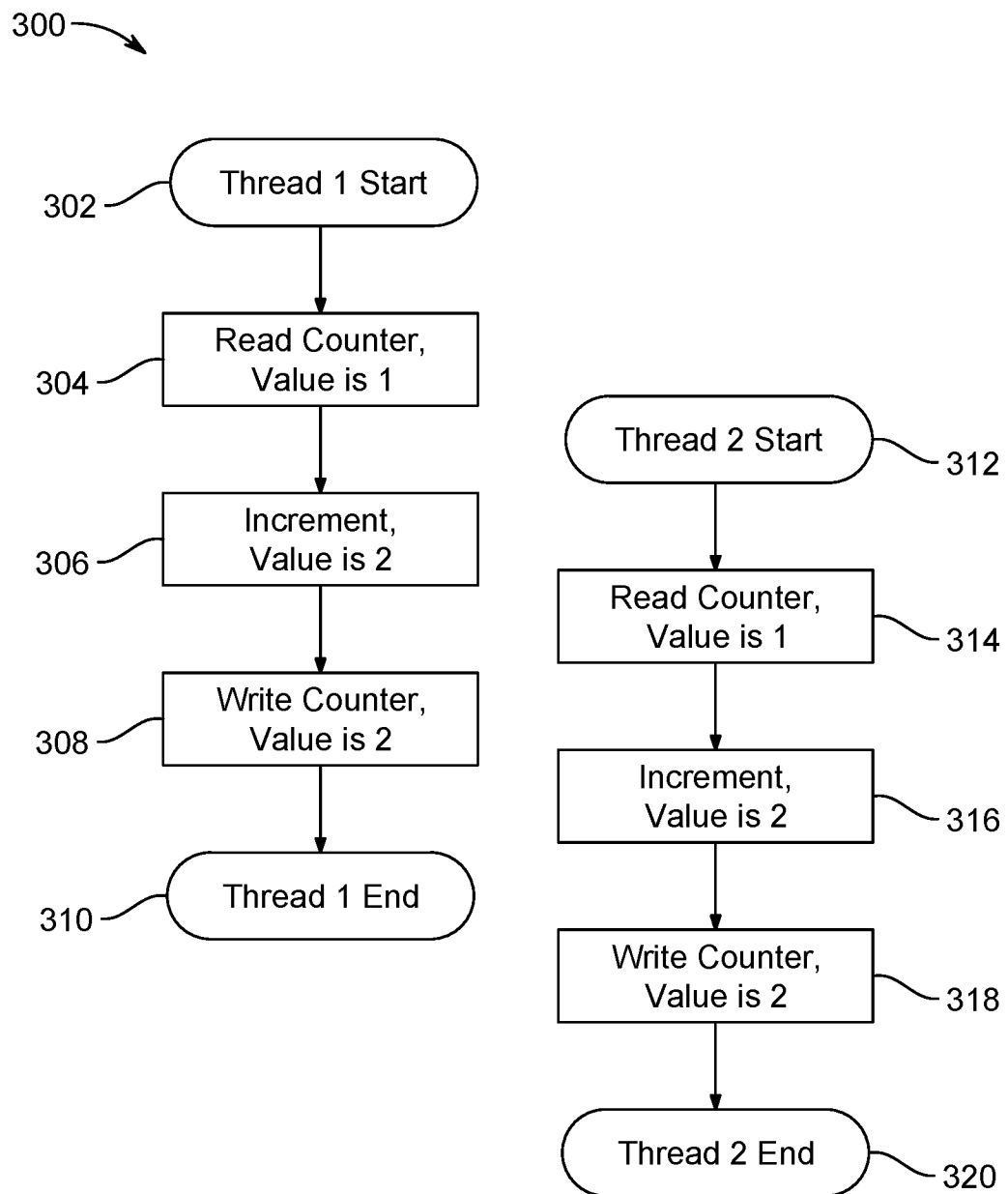


FIG. 3

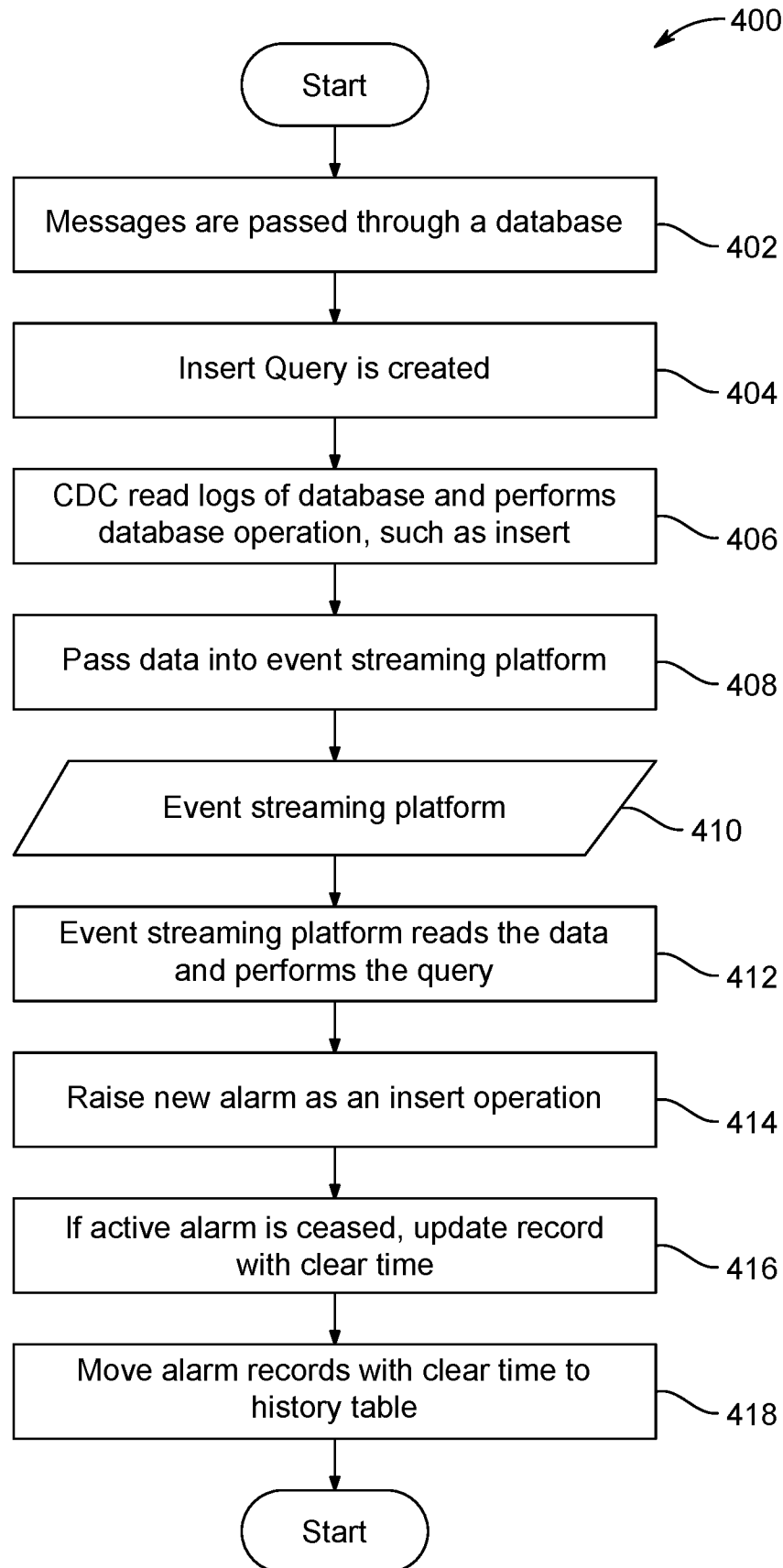


FIG. 4

500

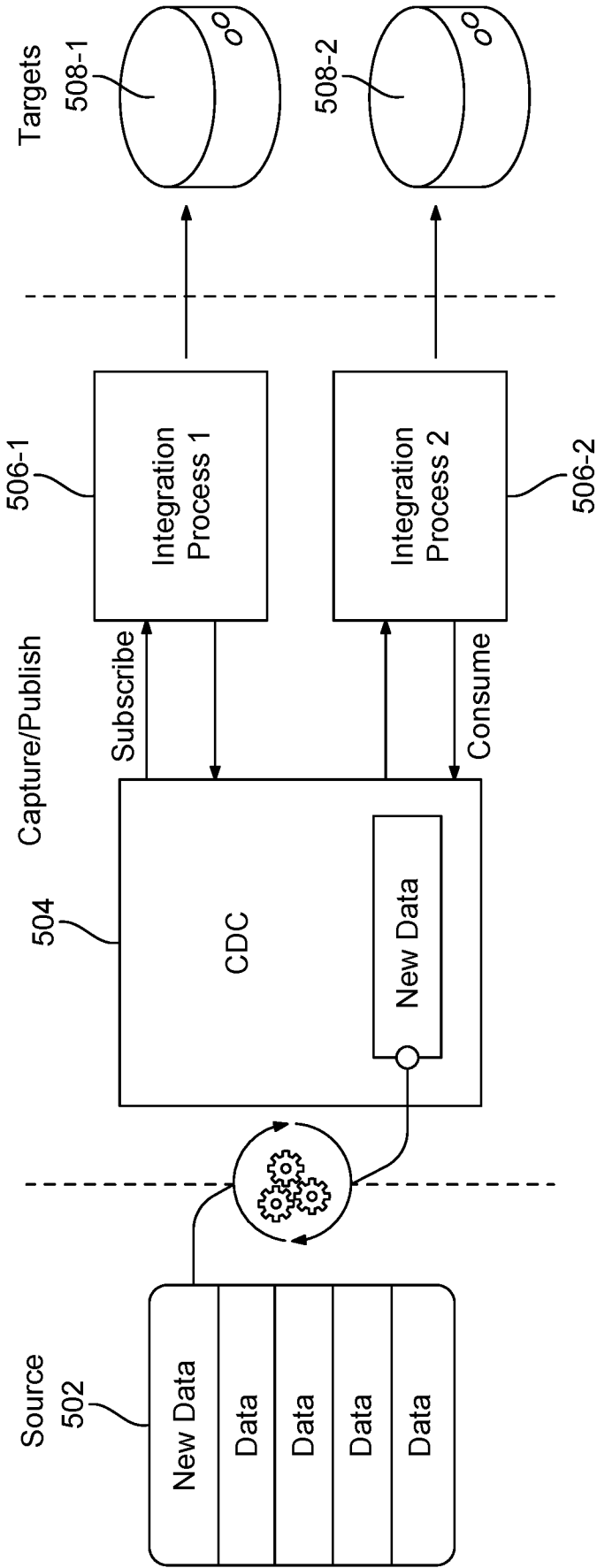


FIG. 5

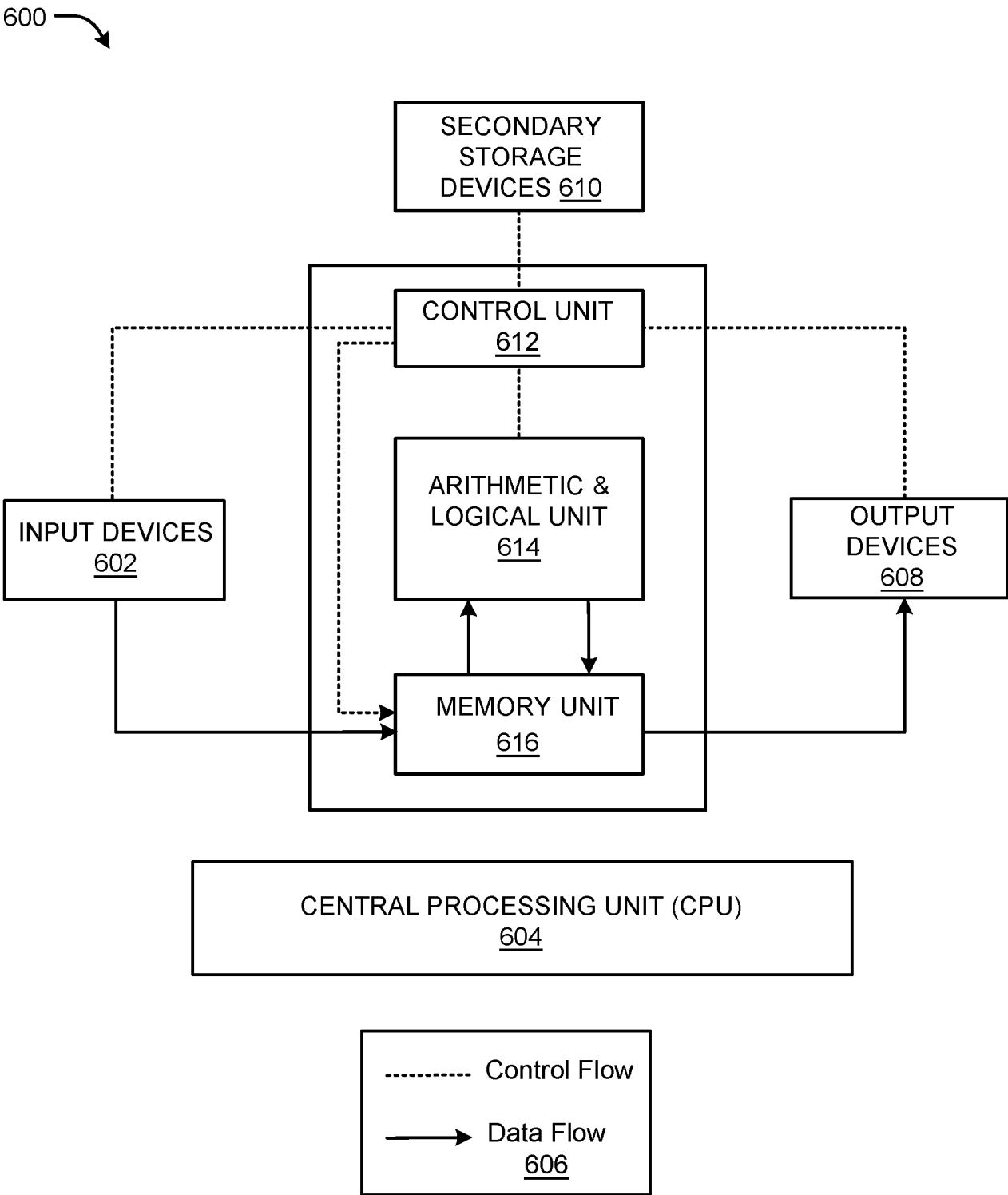


FIG. 6

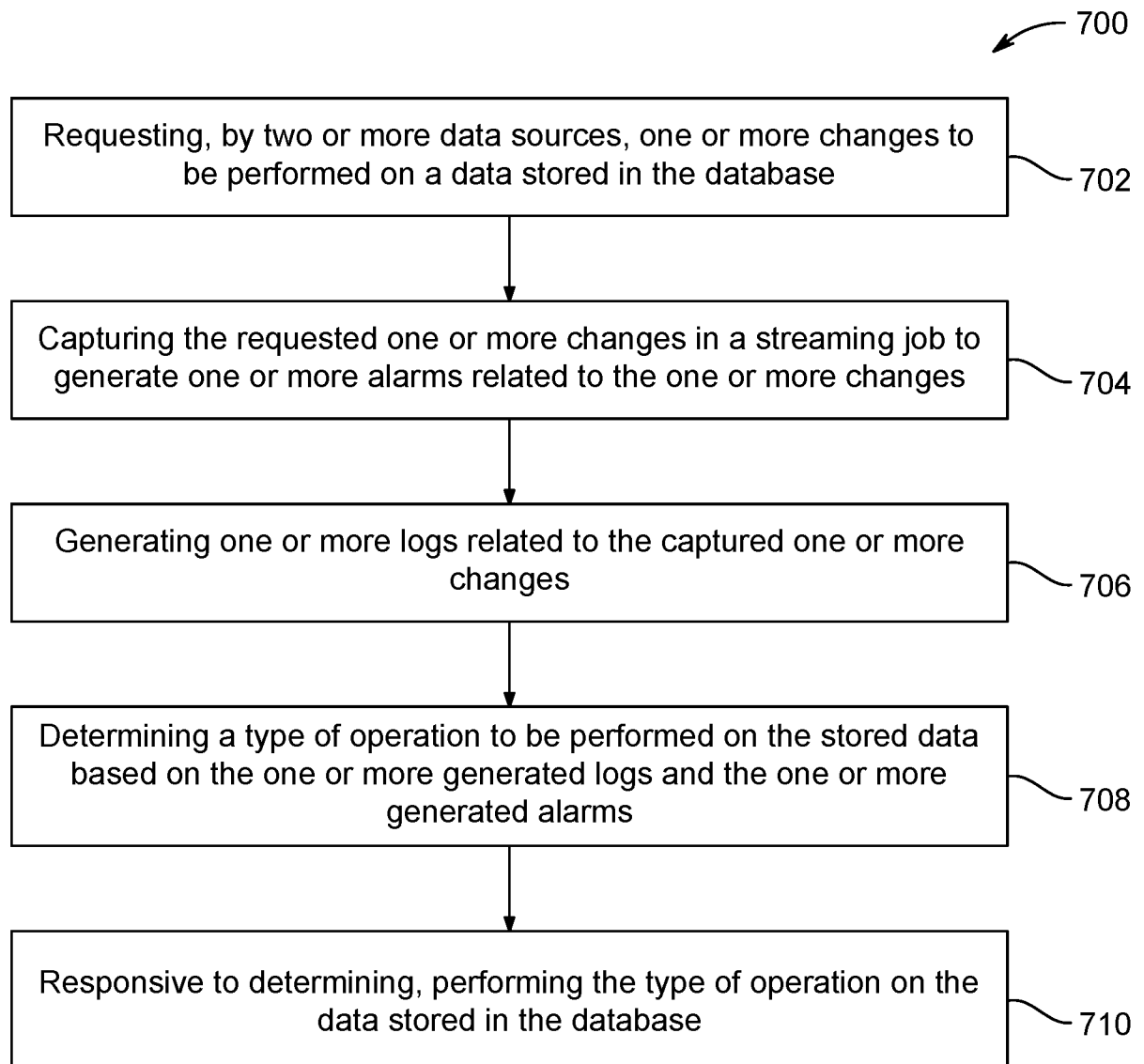


FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IN2024/050638

A. CLASSIFICATION OF SUBJECT MATTER
G06F16/23, G06F16/27 Version=2024.01

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

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic database consulted during the international search (name of database and, where practicable, search terms used)

PatSeer, IPO Internal Database

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US20230143636A1 (SALESFORCE INC) 11 May 2023 (11/05/2023) abstract, figures 1-9, paragraphs 0019-0020, 0028-0030, 0075-0076, 0081, 0082 -----	1-23
X	US20170344596A1 (GOOGLE LLC) 30 November 2017 (30/11/2017) abstract, figures 2-6, paragraphs 0010-0013, 0028, 0035-0039, 0052-0054 -----	1-23
A	US7673181B1 (CA INC) 02 March 2010 (02/03/2010) whole document	1-23

☐ 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

"D" document cited by the applicant in the international application

"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

"I" 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

06-09-2024

Date of mailing of the international search report

06-09-2024

Name and mailing address of the ISA/

Indian Patent Office
Plot No.32, Sector 14, Dwarka, New Delhi-110075
Facsimile No.

Authorized officer

Dinesh Chandra

Telephone No. +91-1125300200

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IN2024/050638

Citation	Pub.Date	Family	Pub.Date
US 20170344596 A1	30-11-2017	AU 2017269114 B1	26-09-2019
		CN 114691306 A	01-07-2022
		EP 3465433 A1	10-04-2019
		JP 6581315 B2	25-09-2019
		WO 2017204988 A1	30-11-2017
		CN 108701051 B	04-03-2022
		KR 102285920 B1	03-08-2021
		KR 102102269 B1	20-04-2020
		EP 4131000 A1	08-02-2023
		US 11354296 B2	07-06-2022
		US 20220276992 A1	01-09-2022