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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 September 2017 (28.09.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/165019 A1

(51) International Patent Classification:
G06F 11/34 (2006.01) *G06F 17/30* (2006.01)

(21) International Application Number:
PCT/US2017/017874

(22) International Filing Date:
15 February 2017 (15.02.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/312,035 23 March 2016 (23.03.2016) US
15/430,024 10 February 2017 (10.02.2017) US

(71) Applicant: **NEC LABORATORIES AMERICA, INC.** [US/US]; 4 Independence Way, Suite 200, Princeton, New Jersey 08540 (US).

(72) Inventors: **ZONG, Bo**; 6821 Ravens Crest Drive, Plainsboro, New Jersey 08536 (US). **XU, Jianwu**; 95 Oneill Court, Lawrenceville, New Jersey 08550 (US). **JIANG, Guofei**; 5 Danby Court, Princeton, New Jersey 08540 (US).

(74) Agent: **KOLODKA, Joseph**; NEC Laboratories America, Inc., 4 Independence Way, Suite 200, Princeton, New Jersey 08540 (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, DJ, 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, KH, KN, KP, KR, KW, 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).

Published:

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

(54) Title: INVARIANT MODELING AND DETECTION FOR HETEROGENEOUS LOGS

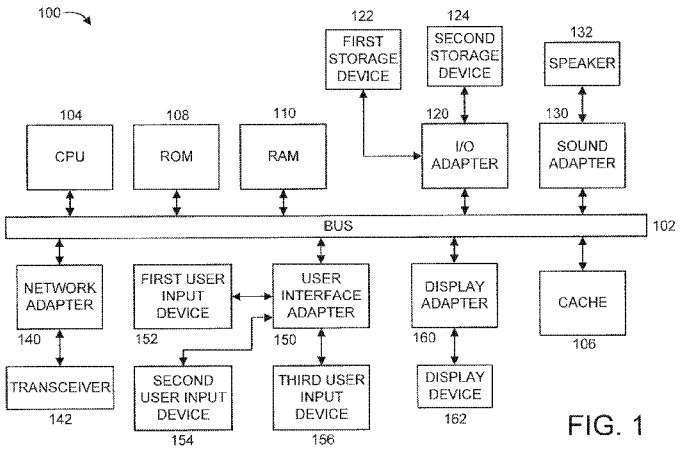


FIG. 1

(57) Abstract: A method is provided that is performed in a network having nodes that generate heterogeneous logs including performance logs and text logs. The method includes performing, during a heterogeneous log training stage, (i) a log-to-time sequence conversion process for transforming clustered ones of training logs, from among the heterogeneous logs, into a set of time sequences that are each formed as a plurality of data pairs of a first configuration and a second configuration based on cluster type, (ii) a time series generation process for synchronizing particular ones of the time sequences in the set based on a set of criteria to output a set of fused time series, and (iii) an invariant model generation process for building invariant models for each time series data pair in the set of fused time series. The method includes controlling an anomaly-initiating one of the plurality of nodes based on the invariant models.

WO 2017/165019 A1

INVARIANT MODELING AND DETECTION FOR HETEROGENEOUS LOGS

RELATED APPLICATION INFORMATION

[0001] This application claims priority to provisional application serial number 62/312,035 filed on March 23, 2016, incorporated herein by reference.

BACKGROUND

Technical Field

[0002] The present invention relates to data processing, and more particularly to invariant modeling and detection for heterogeneous logs.

Description of the Related Art

[0003] Information Technology (IT) systems include a large number of functional components, and these components have dependencies between each other. In such complex systems, heterogeneous log data is generated from individual components, where dependencies between components remain hidden. While invariant analysis has been widely adopted to discover hidden relations in time series data, it is difficult to apply existing tools over heterogeneous logs that are generated from multiple log sources. The key problem is the set of time series derived by logs from different sources are not synchronized. For example, (1) time periods covered by different time series are not aligned; and (2) different time series employ different sampling frequency. Therefore, there is a need for an approach for invariant modeling and detection for heterogeneous logs.

SUMMARY

[0004] These and other drawbacks and disadvantages of the prior art are addressed by the present invention.

[0005] According to an aspect of the present invention, a method is provided that is performed in a network having a plurality of nodes that generate heterogeneous logs including performance logs and text logs. The method includes performing, by a processor during a heterogeneous log training stage, (i) a log-to-time sequence conversion process for transforming clustered ones of training logs, from among the heterogeneous logs, into a set of time sequences that are each formed as a plurality of data pairs of a first configuration and a second configuration based on cluster type, (ii) a time series generation process for synchronizing particular ones of the time sequences in the set based on a set of criteria to output a set of fused time series, and (iii) an invariant model generation process for building invariant models for each time series data pair in the set of fused time series. The method further includes controlling, by the processor, an anomaly-initiating one of the plurality of nodes based on an output of the invariant models.

[0006] According to another aspect of the present invention, a computer program product is provided for invariant model formation for a network having a plurality of nodes that generate heterogeneous logs including performance logs and text logs. The computer program product includes a non-transitory computer readable storage medium having program instructions embodied therewith. The program instructions are executable by a computer to cause the computer to perform a method. The method includes performing, by a processor during a heterogeneous log training stage, (i) a log-

to-time sequence conversion process for transforming clustered ones of training logs, from among the heterogeneous logs, into a set of time sequences that are each formed as a plurality of data pairs of a first configuration and a second configuration based on cluster type, (ii) a time series generation process for synchronizing particular ones of the time sequences in the set based on a set of criteria to output a set of fused time series, and (iii) an invariant model generation process for building invariant models for each time series data pair in the set of fused time series. The method further includes controlling, by the processor, an anomaly-initiating one of the plurality of nodes based on an output of the invariant models.

[0007] According to yet another aspect of the present invention, a computer processing system is provided for invariant model formation for a network having a plurality of nodes that generate heterogeneous logs including performance logs and text logs. The computer processing includes a processor. The processor is configured to perform, during a heterogeneous log training stage, (i) a log-to-time sequence conversion process for transforming clustered ones of training logs, from among the heterogeneous logs, into a set of time sequences that are each formed as a plurality of data pairs of a first configuration and a second configuration based on cluster type, (ii) a time series generation process for synchronizing particular ones of the time sequences in the set based on a set of criteria to output a set of fused time series, and (iii) an invariant model generation process for building invariant models for each time series data pair in the set of fused time series. The processor is further configured to control an anomaly-initiating one of the plurality of nodes based on an output of the invariant models.

[0008] These and other features and advantages will become apparent from the following detailed description of illustrative embodiments thereof, which is to be read in connection with the accompanying drawings.

BRIEF DESCRIPTION OF DRAWINGS

[0009] The disclosure will provide details in the following description of preferred embodiments with reference to the following figures wherein:

[0010] FIG. 1 is a block diagram illustrating an exemplary processing system 100 to which the present principles may be applied, according to an embodiment of the present principles;

[0011] FIGs. 2-3 show exemplary heterogeneous logs 200 to which the present invention can be applied, in accordance with an embodiment of the present invention;

[0012] FIGs. 4-5 show an exemplary detected anomaly 401 from heterogeneous logs 400 to which the present invention can be applied, in accordance with an embodiment of the present invention;

[0013] FIG. 6 shows an exemplary system/method 600 for Invariant Model based Correlation Analysis over Heterogeneous Logs (IMCAHL), in accordance with an embodiment of the present invention;

[0014] FIG. 7 further shows the logs-to-time sequence conversion block 602 of FIG. 6, in accordance with an embodiment of the present invention;

[0015] FIG. 8 shows time sequences 800 for the logs in FIG. 2 that match the log schemas, in accordance with an embodiment of the present invention;

[0016] FIG. 9 further shows the time series generation block 603 of FIG. 6, in accordance with an embodiment of the present invention;

[0017] FIG. 10 shows the time series 1000 obtained from the time sequences in FIG. 8, in accordance with an embodiment of the present invention;

[0018] FIG. 11 further shows the invariant model generation block 604 of FIG. 6, in accordance with an embodiment of the present invention;

[0019] FIG. 12 shows an invariant model 1200 for the pair of log clusters shown in FIG. 10, in accordance with an embodiment of the present invention;

[0020] FIG. 13 further shows the logs-to-time sequence conversion block 606 of FIG. 6, in accordance with an embodiment of the present invention;

[0021] FIG. 14 further shows the time series generation block 607 of FIG. 6, in accordance with an embodiment of the present invention;

[0022] FIG. 15 further shows the time series generation block 608 of FIG. 6, in accordance with an embodiment of the present invention; and

[0023] FIG. 16 shows a block diagram of an exemplary environment 1600 to which the present invention can be applied, in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0024] The present invention is directed to invariant modeling and detection for heterogeneous logs.

[0025] The present invention provides an approach that fuses heterogeneous logs into synchronized time series data so that the following can be performed: invariant analysis; uncover hidden component dependencies; and enable outlier detection.

[0026] To perform invariant analysis over heterogeneous logs in, for example, IT systems and so forth, the present invention addresses the issue that log data is typically encoded in diverse formats with multiple data types. Therefore, the present invention provides a principled approach that integrates heterogeneous logs into a standard data structure for invariant analysis.

[0027] In an embodiment, the present invention provides a principled approach to discover (i) underlying invariants across time series extracted from heterogeneous text logs and system performance time series from multiple log sources, and (ii) detect any system anomalies based on the invariant analysis through machine learning methods. The present invention transforms heterogeneous logs into multi-dimensional time series, and performs fast and robust invariant analysis among the time series. In an embodiment, to address the time series synchronization problem in heterogeneous logs, the present invention first provides a time window generation method that creates a common set of sampling time points shared among all of the time series, and then applies a resampling procedure that fills reasonable values for the sampling time points. The correlation analysis mechanism is based on an invariant model with a fitness score as the parameter,

where both modeling and testing are performed by linear algorithms given a pair of time series.

[0028] Referring now in detail to the figures in which like numerals represent the same or similar elements and initially to FIG. 1, a block diagram illustrating an exemplary processing system 100 to which the present principles may be applied, according to an embodiment of the present principles, is shown. The processing system 100 includes at least one processor (CPU) 104 operatively coupled to other components via a system bus 102. A cache 106, a Read Only Memory (ROM) 108, a Random Access Memory (RAM) 110, an input/output (I/O) adapter 120, a sound adapter 130, a network adapter 140, a user interface adapter 150, and a display adapter 160, are operatively coupled to the system bus 102.

[0029] A first storage device 122 and a second storage device 124 are operatively coupled to system bus 102 by the I/O adapter 120. The storage devices 122 and 124 can be any of a disk storage device (e.g., a magnetic or optical disk storage device), a solid state magnetic device, and so forth. The storage devices 122 and 124 can be the same type of storage device or different types of storage devices.

[0030] A speaker 132 is operatively coupled to system bus 102 by the sound adapter 130. A transceiver 142 is operatively coupled to system bus 102 by network adapter 140. A display device 162 is operatively coupled to system bus 102 by display adapter 160.

[0031] A first user input device 152, a second user input device 154, and a third user input device 156 are operatively coupled to system bus 102 by user interface adapter 150. The user input devices 152, 154, and 156 can be any of a keyboard, a mouse, a keypad, an image capture device, a motion sensing device, a microphone, a device incorporating

the functionality of at least two of the preceding devices, and so forth. Of course, other types of input devices can also be used, while maintaining the spirit of the present principles. The user input devices 152, 154, and 156 can be the same type of user input device or different types of user input devices. The user input devices 152, 154, and 156 are used to input and output information to and from system 100.

[0032] Of course, the processing system 100 may also include other elements (not shown), as readily contemplated by one of skill in the art, as well as omit certain elements. For example, various other input devices and/or output devices can be included in processing system 100, depending upon the particular implementation of the same, as readily understood by one of ordinary skill in the art. For example, various types of wireless and/or wired input and/or output devices can be used. Moreover, additional processors, controllers, memories, and so forth, in various configurations can also be utilized as readily appreciated by one of ordinary skill in the art. These and other variations of the processing system 100 are readily contemplated by one of ordinary skill in the art given the teachings of the present principles provided herein.

[0033] FIGs. 2-3 show exemplary heterogeneous logs 200 to which the present invention can be applied, in accordance with an embodiment of the present invention. The heterogeneous logs 200 include heterogeneous text logs 210 and heterogeneous performance logs 220 (FIG. 2), as well as respective plots 210A and 220A (FIG. 3) of the heterogeneous text logs 210 and heterogeneous performance logs 220.

[0034] FIGs. 4-5 show an exemplary detected anomaly 401 from heterogeneous logs 400 to which the present invention can be applied, in accordance with an embodiment of the present invention. The heterogeneous logs 400 include heterogeneous text logs 410

and heterogeneous performance logs 420 (FIG. 4), as well as respective plots 410A and 420A (FIG. 5) of the heterogeneous text logs 410 and heterogeneous performance logs 420.

[0035] FIG. 6 shows an exemplary system/method 600 for Invariant Model based Correlation Analysis over Heterogeneous Logs (IMCAHL), in accordance with an embodiment of the present invention.

[0036] The system/method 600 includes a heterogeneous log collection for training block 601 and a heterogeneous log collection for testing block 605, and a log management applications block 609.

[0037] Relating to the heterogeneous log collection for training block 601, the system/method 600 includes a logs-to-time sequence conversion block 602, a time series generation block 603, and an invariant model generation block 604.

[0038] Relating to the heterogeneous log collection for testing block 605, the system/method 600 includes a logs-to-time sequence conversion block 606, a time series generation block 607, and an invariant model checking block 608.

[0039] The heterogeneous log collection for training block 601 takes heterogeneous logs from arbitrary/unknown systems or applications. The heterogeneous logs can be obtained from one source (single source from single IT server), or can be obtained from multiple sources (multiple log sources from multiple IT servers). A log message includes a time stamp and the text content with one or multiple fields.

[0040] The logs to time sequence conversion block 601 transforms original training text logs into a set of time sequence data.

[0041] The time series generation block 603 synchronizes the set of time sequences output by 602 and outputs time series for the input time sequences.

[0042] The invariant model generation block 604 analyzes the set of time series output by 603, and builds invariant models for each pair of time series.

[0043] The heterogeneous log collection for testing block 605 takes heterogeneous logs collected from the same system in block 601 for invariant model testing. A log message includes a time stamp and the text content with one or multiple fields. The testing data may come in one batch as a log file, or come in a stream process.

[0044] The logs to time sequence conversion block 606 transforms original testing text logs into a set of time sequence data.

[0045] The time series generation block 607 synchronizes the set of time sequences output by block 606 and output time series for input time sequences.

[0046] The invariant model checking block 608 analyzes the set of time series data output by block 607 based on the corresponding invariant models output by block 604, and outputs anomalies on any time series data point violating the invariant model and the related log messages.

[0047] The log management application block 609 applies a set of management applications onto the heterogeneous logs from block 601 based on the invariant models output by block 603, or onto the heterogeneous logs from block 604 based on the invariant model checking output by block 606. For example, invariant models output by block 603 can be applied to analyze hidden dependency within a target system, and anomalies output by block 606 can be used to detect unexpected system workload or behavior changes. Moreover, based on the detection of an anomaly using an invariant

model, an anomaly-initiating one of a plurality of nodes (e.g., a computer in a cluster of computers, and so forth) can be controlled. In an embodiment, the control can involve powering down a root cause computer processing device at the anomaly-initiating one of the plurality of nodes to mitigate an error propagation therefrom. In an embodiment, the control can involve terminating a root cause process executing on a computer processing device at the anomaly-initiating one of the plurality of nodes to mitigate an error propagation therefrom.

[0048] FIG. 7 further shows the logs-to-time sequence conversion block 602 of FIG. 6, in accordance with an embodiment of the present invention.

[0049] The logs-to-time sequence conversation block 602 includes a log schema recognition block 602A and a per-cluster time sequence generation block 602B.

[0050] Regarding the log scheme recognition block 602A, a set of log schemas matching the training logs can be provided by users directly, or generated automatically by a pattern recognition procedure on all the heterogeneous logs as follows in block 602A1-602A3:

Block 602A1: tokenization, similarity, clustering;

Block 602A2: alignment, log schema discovery/recognition; and

Block 603A3: classification as log or performance cluster.

[0051] At block 602A1 (tokenization; similarity; clustering), taking arbitrary heterogeneous logs (from step 601 of FIG. 6), a tokenization process is performed so as to generate semantically meaningful tokens from logs. After tokenization, a similarity measurement on heterogeneous logs is applied. This similarity measurement leverages both the log layout information and log content information, and it is specially tailored to

arbitrary heterogeneous logs. Once the similarities among logs are obtained, a log clustering algorithm can be applied so as to generate and output log clusters. IMCAHL allows users to plug in their favorite clustering algorithms.

[0052] At block 602A2 (alignment; log schema discovery/recognition), once the logs are clustered, the logs are also aligned within each cluster. The log alignment is designed to preserve the unknown layouts of heterogeneous logs so as to help log schema recognition in the following steps. Once the logs are aligned, log schema discovery is conducted so as to find the most representative layouts and log fields.

[0053] The following steps show how we perform log field recognition. First, fields such as time stamps, Internet Protocol (IP) addresses, and universal resource locators (URLs) are recognized based on prior knowledge about their syntax structures. Second, fields which are highly stable in the logs are recognized as general constant fields in log schemas. Third, the rest fields are recognized as general variable fields, including number fields, hybrid string fields, and string fields.

[0054] At block 602A3 (classification as log or performance cluster), we classify log clusters as text log clusters and performance log clusters. A cluster is a performance log cluster, if its log schema contains three fields. The first field is a constant field indicating performance metric names, the second field is time stamp field, and the third field is number field. If a cluster is not a performance log cluster, then it is a text log cluster. For example, log messages about CPU usage are usually grouped into a performance log cluster, and one such message could be "CPU_usage, 2015/5/17 01:30:20, 60.72".

[0055] Regarding the per-cluster time sequence generation block 602B, within one cluster, logs share a common log schema and are taken as same type of logs. We generate time sequences for each log cluster as follows per block 602B1 and 602B2:

602B1: performance log cluster time sequence generation; and

602B2: text log cluster time sequence generation.

[0056] At block 602B1, for a performance log cluster, we generate its time sequence as follows. First, we order log messages in the cluster. Second, we extract values in the time stamp and the number fields, and build a tuple (X, Y) for each log message, where X is the value in its time stamp field and Y is the value in its number field. Assume we have k log messages. After this step, we obtain a time sequence $s = <(X_1, Y_2), \ldots, (X_k, Y_k)>$, where $X_1 < X_2 < \ldots < X_k$.

[0057] At block 602B2, for a text log cluster, we generate its time sequence as follows. First, we order log messages in the cluster. Second, we extract values in the time stamp field, and build a tuple $(X, 1)$ for each log message, where X is the value in its time stamp field and 1 indicates such kind of logs occur once at time X. Assume we have k log messages. After this step, we obtain a time sequence $s = <(X_1, 1), \ldots, (X_k, 1)>$, where $X_1 < X_2 < \ldots < X_k$.

[0058] FIG. 8 shows time sequences 800 for the logs in FIG. 2 that match the log schemas, in accordance with an embodiment of the present invention. That is, FIG. 8 shows an example of IMCAHL time sequence data for the logs in FIG. 2, in accordance with an embodiment of the present invention.

[0059] FIG. 9 further shows the time series generation block 603 of FIG. 6, in accordance with an embodiment of the present invention.

[0060] The time series generation block 603 includes a time window generation block 603A and a resampling block 603B.

[0061] For each log cluster/schema, we obtain a time sequence $s = <(X_1, Y_1), (X_2, Y_2), ..., (X_k, Y_k)>$ output from 602B (see FIG. 7), the following is time series generation procedure that fuses multiple time sequences into multiple time series that share identical sampling time and frequency. Given a user-define time window size w, we perform time series generation as follows.

[0062] Regarding the time window generation block 603A, take the time domain as a one-dimensional space, which starts at epoch time 0 (i.e., 1970/1/1 00:00:00) and goes into the infinite future. We partition time domain into time windows with identical size, where the duration of a time window is w.

[0063] Regarding the resampling block 603B, we denote a time window W as a time range $[t_s, t_e]$, where t_s is the starting time point of W and t_e is the end time point of W. Note that time point t_s is not included in W so that time windows are disjoint. Given a time sequence $s = <(X_1, Y_1), \ldots, (X_k, Y_k)>$, we identify a sequence of time windows $<W_1, W_2, \ldots, W_m>$ that fully covers time stamps $\{X_1, X_2, \ldots, X_k\}$.

[0064] The resampling block 603B can involve:

603B1: resampling a time sequence output from a performance log cluster; and

603B2: resampling a time sequence output from a text log cluster of log schema P.

[0065] At block 603B1 (for a time sequence output from a performance log cluster), we transform $s = <(X_1, Y_1), \ldots, (X_k, Y_k)>$ into time series $ts = <(X'_1, Y'_1), \ldots, (X'_m, Y'_m)>$. In ts, X'_i is the end time point of W_i, and Y'_i is obtained by performing linear interpolation at X'_i based on s.

[0066] At block 603B2 (for a time sequence output from a text log cluster of log schema P), we transform $s = <(X_1, Y_1), \ldots, (X_k, Y_k)>$ into time series $ts = <(X'_1, Y'_1), \ldots, (X'_m, Y'_m)>$. In ts, X'_i is the end time point of W_i, and Y'_i is the number of log messages that match log schema P within time window W_i.

[0067] FIG. 10 shows the time series 1000 obtained from the time sequences in FIG. 8, in accordance with an embodiment of the present invention.

[0068] FIG. 11 further shows the invariant model generation block 604 of FIG. 6, in accordance with an embodiment of the present invention.

[0069] The invariant model generation block 604 includes a merging time series block 604A and an invariant modeling block 604B.

[0070] For the set of time series output from block 603B of FIG. 9, the following is the invariant model generation procedure that produces invariant models for log cluster pairs.

[0071] Regarding merging time series block 604A, we collect the set of time series output from block 602, and merge them into a multi-dimensional time series.

[0072] Regarding the invariant modeling block, with the multi-dimensional time series, we utilize existing correlation analysis tools, such as SIAT (System Invariants Analysis Technology) to generate invariant models for log cluster pairs. In particular, in an embodiment, we filter out invariants whose fitness score is no more than 0.7.

[0073] FIG. 12 shows an invariant model 1200 for the pair of log clusters shown in FIG. 10: one is the text log cluster with schema P_1, and the other is the performance log cluster with schema P_2.

[0074] FIG. 13 further shows the logs-to-time sequence conversion block 606 of FIG. 6, in accordance with an embodiment of the present invention.

[0075] The logs-to-time sequence conversion block 606 includes a log schema selection block 606A and a per-message time sequence generation block 606B.

[0076] Regarding the log schema selection block 606A, from the set of log schemas generated from block 601, only the schemas with invariant models are selected for the rest of the testing procedure.

[0077] Regarding the per-message time sequence generation block 606B, for each log message i in the testing data, find the log schema P it matches (e.g., through a regular expression testing), and extract its time stamp X_i. If P is a text log schema, this block 606B outputs a tuple (X_i, 1) for this message; if P is a performance log schema, this block 606B outputs a tuple (X_i, Y_i) for this message, where Y_i is the value of the number field in this message.

[0078] FIG. 14 further shows the time series generation block 607 of FIG. 6, in accordance with an embodiment of the present invention.

[0079] For each log schema, we obtain a time sequence $s = <(X_1, Y_1), (X_2, Y_2), ..., (X_k, Y_k)>$ output from block 606B (see FIG. 13), the following is time series generation procedure that fuses multiple time sequences into multiple time series that share identical sampling time and frequency. Given a user-define time window size w, we perform time series generation as follows per blocks 1407A and 1407B.

[0080] The time series generation block 607 includes a time window generation block 607A and a resampling block 607B.

[0081] Regarding the time window generation block 607A, time windows are generated following the same approach in block 603A (see FIG. 9).

[0082] Regarding the sampling block 607B, the block is performed following the approach from block 603B in FIG. 9 over both time sequences for text log schemas and time sequences for performance schema. For each time sequence, this block 670B outputs its corresponding time series.

[0083] FIG. 15 further shows the time series generation block 608 of FIG. 6, in accordance with an embodiment of the present invention.

[0084] For a pair of log schemas with invariant models, the following is the invariant model testing procedure to decide if it violates correlation patterns learned from training data. An anomaly will be reported if such violation exists.

[0085] The time series generation block 608 includes a merging time series block 608A and an invariant model testing block 608B.

[0086] Regarding the merging time series block 608A, the set of time series output from block 607B (see FIG. 14) is collected and merged into a multi-dimensional time series.

[0087] Regarding the invariant model testing block 608B, with the multi-dimensional time series, we utilize existing correlation analysis tools, such as SIAT, to test if invariant models are broken for time series output by 801. When broken invariants are detected, anomalies are reported.

[0088] The following shows the three periodicity anomalies detected from the logs in FIG. 4 based on the invariant model learned from the logs in FIG. 2:

Invariant between P1 and P2 is broken, detected at time 2014/4/22 10:02:00.

[0089] FIG. 16 shows a block diagram of an exemplary environment 1600 to which the present invention can be applied, in accordance with an embodiment of the present

invention. The environment 1600 is representative of an invariant computer network to which the present invention can be applied. The elements shown relative to FIG. 2 are set forth for the sake of illustration. However, it is to be appreciated that the present invention can be applied to other network configurations as readily contemplated by one of ordinary skill in the art given the teachings of the present invention provided herein, while maintaining the spirit of the present invention.

[0090] The environment 200 at least includes a set of nodes, individually and collectively denoted by the figure reference numeral 210. Each of the nodes 210 can include one or more servers or other types of computer processing devices, individually and collectively denoted by the figure reference numeral 211. The computer processing devices 211 can include, for example, but are not limited to, machines (e.g., industrial machines, assembly line machines, robots, etc.) and so forth. For the sake of illustration, each of the nodes 210 is shown with a set of servers 211. Each of the nodes generates and/or otherwise provides time series data.

[0091] In an embodiment, the present invention performs invariant modeling and detection for heterogeneous logs, as described herein. Based on the ranks, a computer processing system can be controlled in order to mitigate errors stemming from propagation of an anomaly.

[0092] In the embodiment shown in FIG. 2, the elements thereof are interconnected by a network(s) 201. However, in other embodiments, other types of connections can also be used. Additionally, one or more elements in FIG. 2 may be implemented by a variety of devices, which include but are not limited to, Digital Signal Processing (DSP) circuits, programmable processors, Application Specific Integrated Circuits (ASICs), Field

Programmable Gate Arrays (FPGAs), Complex Programmable Logic Devices (CPLDs), and so forth. These and other variations of the elements of environment 200 are readily determined by one of ordinary skill in the art, given the teachings of the present invention provided herein, while maintaining the spirit of the present invention.

[0093] A description will now be given regarding specific competitive/commercial values of the solution achieved by the present invention.

[0094] The present invention significantly reduces the complexity of performing invariant analysis among heterogeneous logs, even when prior knowledge about the system might not be available. By integrating advanced text mining and time series analysis in a novel way, the present invention provides an automated method that converts heterogeneous logs into multiple time series and then fuses these time series into multi-dimensional time series by time window generation and resampling. The resulting multi-dimensional time series enables invariant analysis over heterogeneous logs, and allows efficient anomaly detection based invariant models.

[0095] Embodiments described herein may be entirely hardware, entirely software or including both hardware and software elements. In a preferred embodiment, the present invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

[0096] Embodiments may include a computer program product accessible from a computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. A computer-usable or computer readable medium may include any apparatus that stores, communicates, propagates, or transports the program for use by or in connection with the instruction

execution system, apparatus, or device. The medium can be magnetic, optical, electronic, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. The medium may include a computer-readable medium such as a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk and an optical disk, etc.

[0097] It is to be appreciated that the use of any of the following "/", "and/or", and "at least one of", for example, in the cases of "A/B", "A and/or B" and "at least one of A and B", is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of both options (A and B). As a further example, in the cases of "A, B, and/or C" and "at least one of A, B, and C", such phrasing is intended to encompass the selection of the first listed option (A) only, or the selection of the second listed option (B) only, or the selection of the third listed option (C) only, or the selection of the first and the second listed options (A and B) only, or the selection of the first and third listed options (A and C) only, or the selection of the second and third listed options (B and C) only, or the selection of all three options (A and B and C). This may be extended, as readily apparent by one of ordinary skill in this and related arts, for as many items listed.

[0098] Having described preferred embodiments of a system and method (which are intended to be illustrative and not limiting), it is noted that modifications and variations can be made by persons skilled in the art in light of the above teachings. It is therefore to be understood that changes may be made in the particular embodiments disclosed which are within the scope and spirit of the invention as outlined by the appended claims.

Having thus described aspects of the invention, with the details and particularity required by the patent laws, what is claimed and desired protected by Letters Patent is set forth in the appended claims.

WHAT IS CLAIMED IS:

1. A method performed in a network having a plurality of nodes that generate heterogeneous logs including performance logs and text logs, the method comprising:

performing, by a processor during a heterogeneous log training stage, (i) a log-to-time sequence conversion process for transforming clustered ones of training logs, from among the heterogeneous logs, into a set of time sequences that are each formed as a plurality of data pairs of a first configuration and a second configuration based on cluster type, (ii) a time series generation process for synchronizing particular ones of the time sequences in the set based on a set of criteria to output a set of fused time series, and (iii) an invariant model generation process for building invariant models for each time series data pair in the set of fused time series; and

controlling, by the processor, an anomaly-initiating one of the plurality of nodes based on an output of the invariant models.

2. The method of claim 1, wherein the log-to-time sequence conversion process comprises a log schema recognition process and a per-cluster time sequence generation process.

3. The method of claim 2, wherein the log schema recognition process comprises:

performing a tokenization process on the heterogeneous logs to generate tokens;

performing a log similarity process on the heterogeneous logs based on the tokens to identify log similarities amongst the heterogeneous logs; and

clustering the heterogeneous logs based on the log similarities.

4. The method of claim 2, wherein the per-cluster time sequence generation process comprises, for the performance logs, forming in the first configuration each of the plurality of data pairs to consist of a time stamp field value and a number field value.

5. The method of claim 2, wherein the per-cluster time sequence generation processes comprises, for the text logs, forming in the second configuration each of the plurality of data pairs to consist of a time stamp field value and a value indicating that a text log type occurs once at a time represented by the time stamp field value.

6. The method of claim 1, wherein the time series generation process comprises:

performing a time window generation process that partitions a time domain into a plurality of disjoint time windows of equal size and duration; and

resampling the time sequences in the set in accordance with the plurality of disjoint time windows.

7. The method of claim 6, wherein said resampling step comprises:

transforming the time sequences in the set output from a performance log cluster into transformed time sequences each having a plurality of transformed of data pairs that include a window end time point and a linear interpolated sequence-based value; and

transforming the time sequences in the set output from a text log cluster of a log schema into transformed time sequences each having a plurality of transformed of data pairs that include a window end time point and a number of log messages matching the log schema within a corresponding one of the plurality of time windows.

8. The method of claim 1, wherein the set of criteria, used by the time series generation process to determine the particular ones of the time series in the set to synchronize, comprises a common sampling time and a common frequency.

9. The method of claim 1, wherein the invariant model generation process comprises merging the fused time series in the set to form a multi-dimensional time series, and wherein the invariant models are built from the multi-dimensional time series.

10. The method of claim 1, further comprising repeating, by the processor during a heterogeneous log testing stage involving testing logs in place of the training logs, (i) the log-to-time sequence conversion process and (ii) the time series generation process, in order to test the invariant models.

11. The method of claim 1, further comprising performing, by a processor during a heterogeneous log testing stage, an invariant model testing process for testing the invariant models based on correlation mismatches in correlation patterns learned from the heterogeneous log training stage.

12. A computer program product for invariant model formation for a network having a plurality of nodes that generate heterogeneous logs including performance logs and text logs, the computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith, the program instructions executable by a computer to cause the computer to perform a method comprising:

performing, by a processor during a heterogeneous log training stage, (i) a log-to-time sequence conversion process for transforming clustered ones of training logs, from among the heterogeneous logs, into a set of time sequences that are each formed as a plurality of data pairs of a first configuration and a second configuration based on cluster type, (ii) a time series generation process for synchronizing particular ones of the time sequences in the set based on a set of criteria to output a set of fused time series, and (iii) an invariant model generation process for building invariant models for each time series data pair in the set of fused time series; and

controlling, by the processor, an anomaly-initiating one of the plurality of nodes based on an output of the invariant models.

13. The computer program product of claim 12, wherein the log-to-time sequence conversion process comprises a log schema recognition process and a per-cluster time sequence generation process.

14. The computer program product of claim 13, wherein the log schema recognition process comprises:

performing a tokenization process on the heterogeneous logs to generate tokens;

performing a log similarity process on the heterogeneous logs based on the tokens to identify log similarities amongst the heterogeneous logs; and

clustering the heterogeneous logs based on the log similarities.

15. The computer program product of claim 13, wherein the per-cluster time sequence generation process comprises, for the performance logs, forming in the first configuration each of the plurality of data pairs to consist of a time stamp field value and a number field value.

16. The computer program product of claim 13, wherein the per-cluster time sequence generation processes comprises, for the text logs, forming in the second configuration each of the plurality of data pairs to consist of a time stamp field value and a value indicating that a text log type occurs once at a time represented by the time stamp field value.

17. The computer program product of claim 12, wherein the time series generation process comprises:

performing a time window generation process that partitions a time domain into a plurality of disjoint time windows of equal size and duration; and

resampling the time sequences in the set in accordance with the plurality of disjoint time windows.

18. The computer program product of claim 17, wherein said resampling step comprises:

transforming the time sequences in the set output from a performance log cluster into transformed time sequences each having a plurality of transformed of data pairs that include a window end time point and a linear interpolated sequence-based value; and

transforming the time sequences in the set output from a text log cluster of a log schema into transformed time sequences each having a plurality of transformed of data pairs that include a window end time point and a number of log messages matching the log schema within a corresponding one of the plurality of time windows.

19. The computer program product of claim 12, wherein the set of criteria, used by the time series generation process to determine the particular ones of the time series in the set to synchronize, comprises a common sampling time and a common frequency.

20. A computer processing system for invariant model formation for a network having a plurality of nodes that generate heterogeneous logs including performance logs and text logs, the computer processing comprising:

a processor configured to:

perform, during a heterogeneous log training stage, (i) a log-to-time sequence conversion process for transforming clustered ones of training logs, from among the heterogeneous logs, into a set of time sequences that are each formed as a plurality of data pairs of a first configuration and a second configuration based on cluster type, (ii) a time series generation process for synchronizing particular ones of the time sequences in the set based on a set of criteria to output a set of fused time series, and (iii) an invariant model generation process for building invariant models for each time series data pair in the set of fused time series; and

control an anomaly-initiating one of the plurality of nodes based on an output of the invariant models.

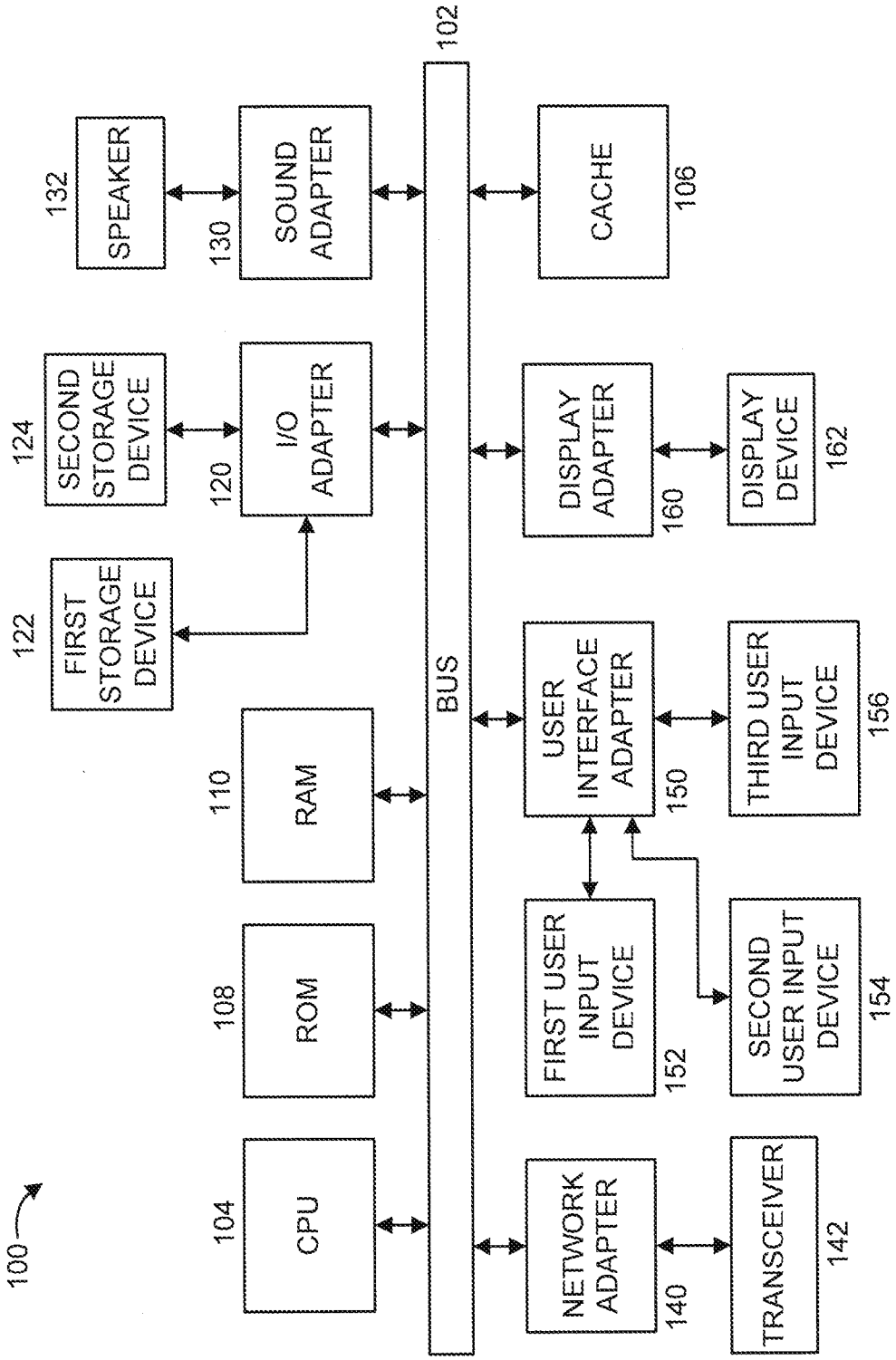

100
CPU
104
ROM
108
RAM
110
FIRST STORAGE DEVICE
122
SECOND STORAGE DEVICE
124
I/O ADAPTER
120
SPEAKER
132
SOUND ADAPTER
130
BUS
102
CACHE
106
DISPLAY ADAPTER
160
DISPLAY DEVICE
162
USER INTERFACE ADAPTER
150
THIRD USER INPUT DEVICE
156
FIRST USER INPUT DEVICE
152
SECOND USER INPUT DEVICE
154
NETWORK ADAPTER
140
TRANSCEIVER
142

FIG. 1

200

210

[Text logs]
Time: 2014/04/21 0:00:11, DB instance: mdb01, Info: full table join is performed
Time: 2014/04/21 0:00:23, DB instance: mdb02, Info: full table join is performed
Time: 2014/04/21 0:01:15, DB instance: mdb03, Info: full table join is performed
Time: 2014/04/21 0:01:16, DB instance: mdb02, Info: full table join is performed
Time: 2014/04/21 0:01:22, DB instance: mdb01, Info: full table join is performed
Time: 2014/04/21 0:04:22, DB instance: mdb02, Info: full table join is performed
Time: 2014/04/21 0:06:17, DB instance: mdb01, Info: full table join is performed
Time: 2014/04/21 0:06:21, DB instance: mdb03, Info: full table join is performed

220

[Computer Performance logs]
CPU_usage,2014/04/21 0:01:00,12.01
CPU_usage,2014/04/21 0:02:00,18.03
CPU_usage,2014/04/21 0:03:00,00.01
CPU_usage,2014/04/21 0:04:00,00.07
CPU_usage,2014/04/21 0:05:00,06.02
CPU_usage,2014/04/21 0:06:00,00.01
CPU_usage,2014/04/21 0:07:00,12.11
CPU_usage,2014/04/21 0:08:00,00.17

FIG. 2

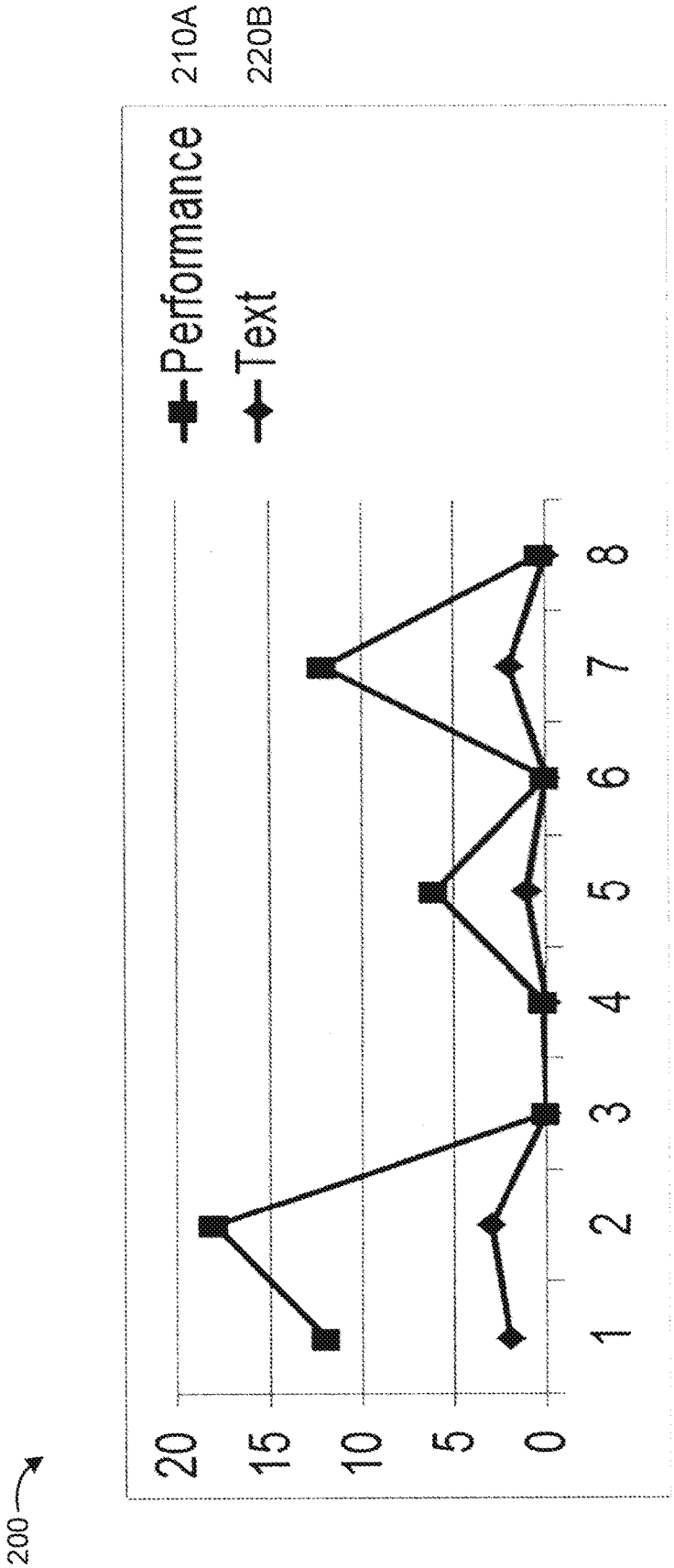
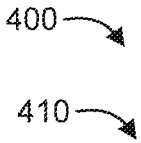


FIG. 3



[Text logs]
Time: 2014/04/22 10:00:11, DB instance: mdb01, Info: full table join is performed
Time: 2014/04/22 10:00:23, DB instance: mdb02, Info: full table join is performed
Time: 2014/04/22 10:01:15, DB instance: mdb03, Info: full table join is performed
Time: 2014/04/22 10:04:22, DB instance: mdb02, Info: full table join is performed
Time: 2014/04/22 10:06:17, DB instance: mdb01, Info: full table join is performed
Time: 2014/04/22 10:06:21, DB instance: mdb03, Info: full table join is performed

420

[Performance logs]
CPU_usage,2014/04/22 10:01:00,12.06
CPU_usage,2014/04/22 10:02:00,60.03
CPU_usage,2014/04/22 10:03:00,00.01
CPU_usage,2014/04/22 10:04:00,00.07
CPU_usage,2014/04/22 10:05:00,06.02
CPU_usage,2014/04/22 10:06:00,00.01
CPU_usage,2014/04/22 10:07:00,12.11
CPU_usage,2014/04/22 10:08:00,00.17

FIG. 4

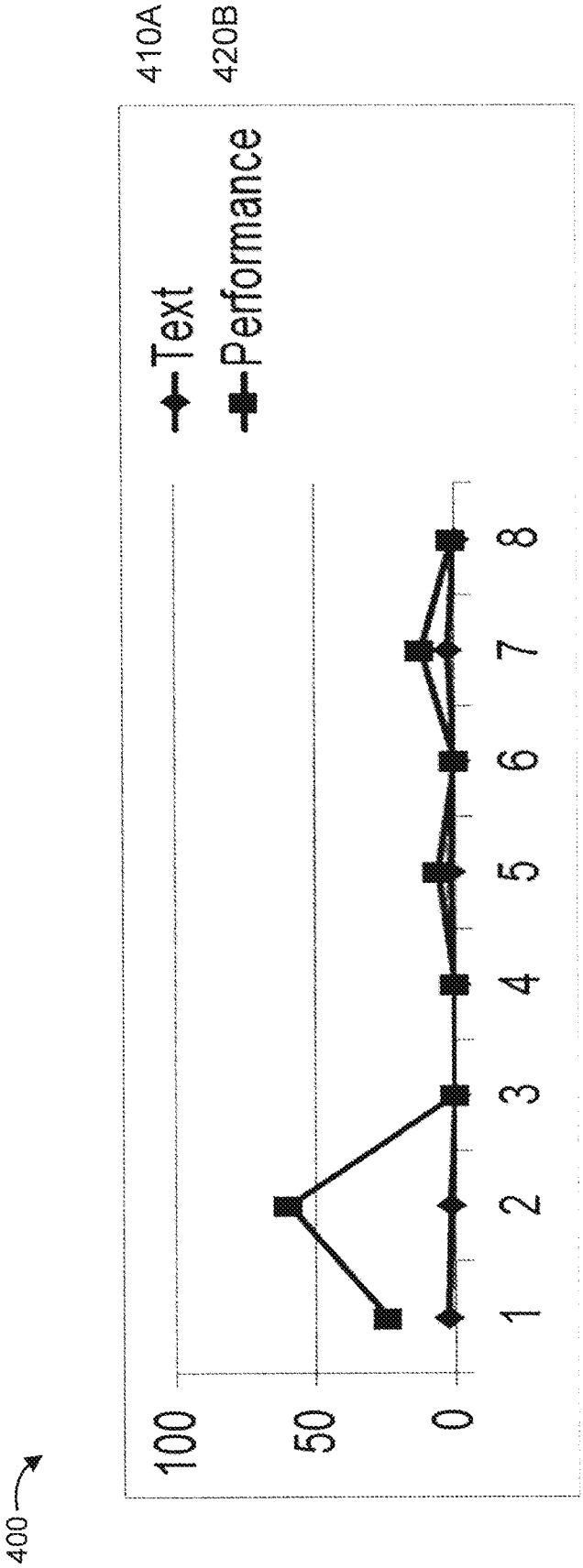


FIG. 5

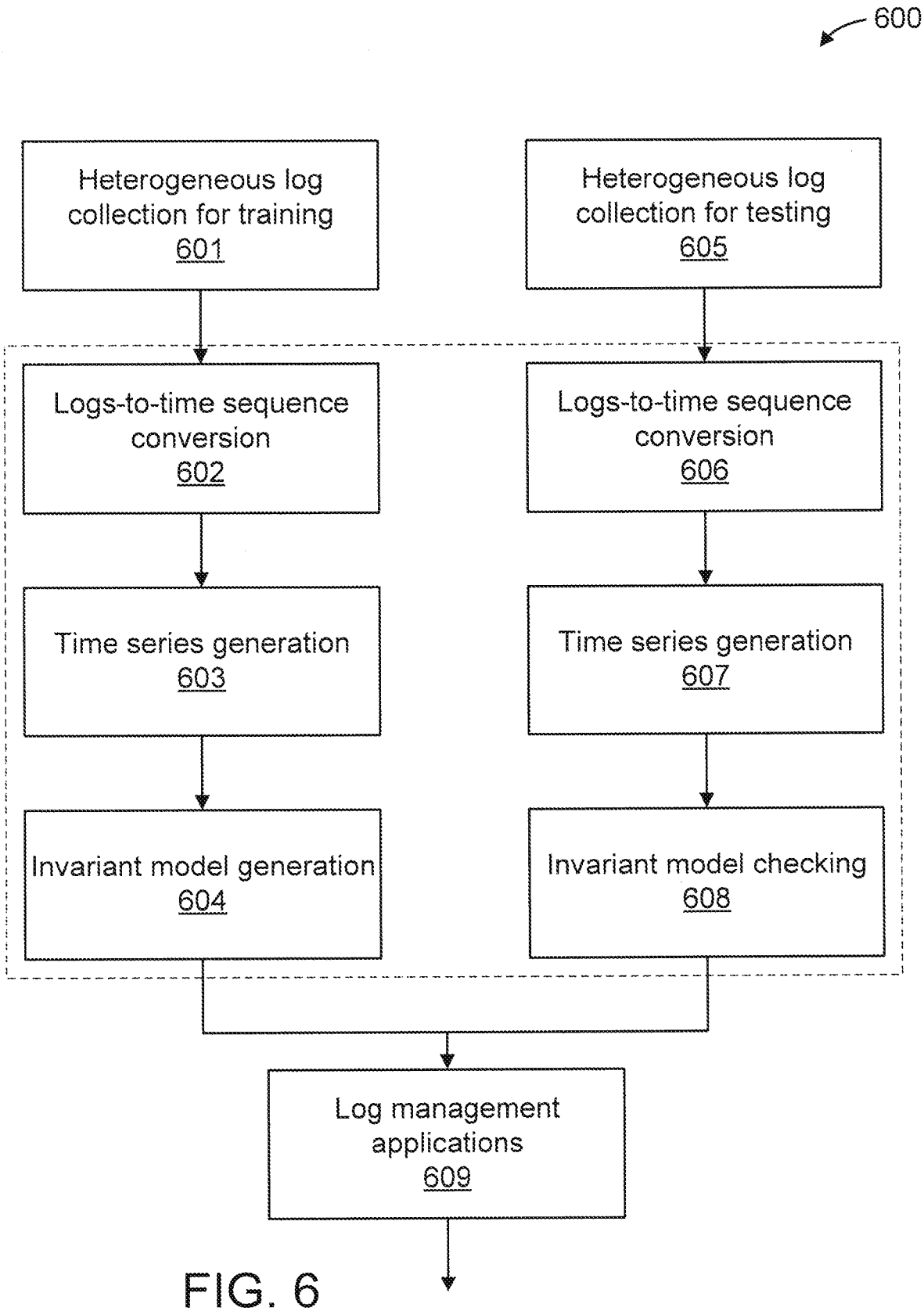

600
Heterogeneous log collection for training 601
Heterogeneous log collection for testing 605
Logs-to-time sequence conversion 602
Logs-to-time sequence conversion 606
Time series generation 603
Time series generation 607
Invariant model generation 604
Invariant model checking 608
Log management applications 609

FIG. 6

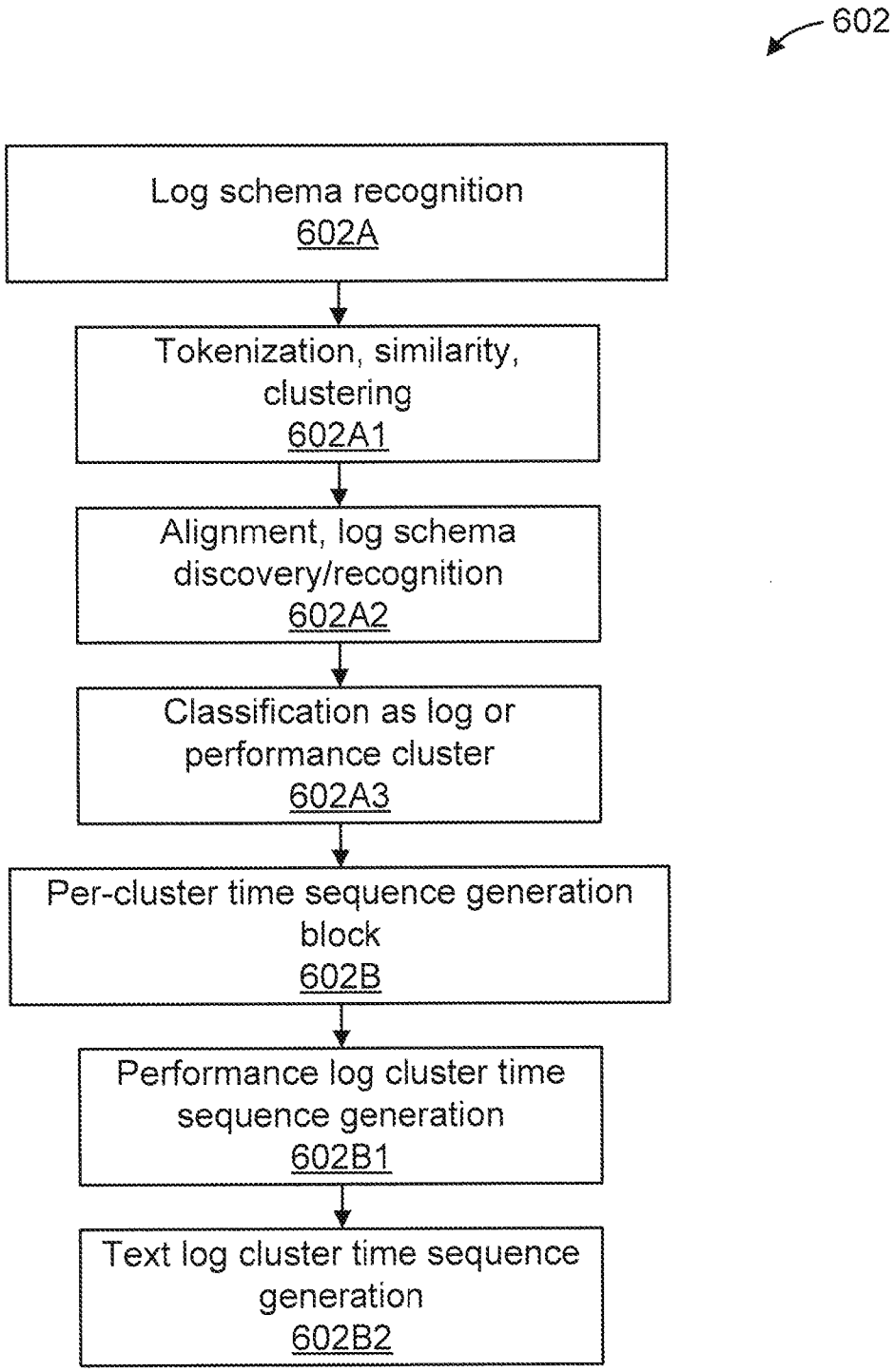

602
Log schema recognition
602A
Tokenization, similarity, clustering
602A1
Alignment, log schema discovery/recognition
602A2
Classification as log or performance cluster
602A3
Per-cluster time sequence generation block
602B
Performance log cluster time sequence generation
602B1
Text log cluster time sequence generation
602B2

FIG. 7

800
Text log schema P_1
(X_1 = 2014/04/21 0:00:11, Y_1 = 1)
(X_2 = 2014/04/21 0:00:23, Y_2 = 1)
(X_3 = 2014/04/21 0:01:15, Y_3 = 1)
(X_4 = 2014/04/21 0:01:16, Y_4 = 1)
(X_5 = 2014/04/21 0:01:22, Y_5 = 1)
(X_6 = 2014/04/21 0:04:22, Y_6 = 1)
(X_7 = 2014/04/21 0:06:17, Y_7 = 1)
(X_8 = 2014/04/21 0:06:21, Y_8 = 1)
Performance log schema P_2
(X_1 = 2014/04/21 0:01:00, Y_1 = 40.01)
(X_2 = 2014/04/21 0:02:00, Y_1 = 60.03)
(X_3 = 2014/04/21 0:03:00, Y_1 = 00.01)
(X_4 = 2014/04/21 0:04:00, Y_1 = 00.07)
(X_5 = 2014/04/21 0:05:00, Y_1 = 20.02)
(X_6 = 2014/04/21 0:06:00, Y_1 = 00.01)
(X_7 = 2014/04/21 0:07:00, Y_1 = 40.11)
(X_8 = 2014/04/21 0:08:00, Y_1 = 00.17)

FIG. 8

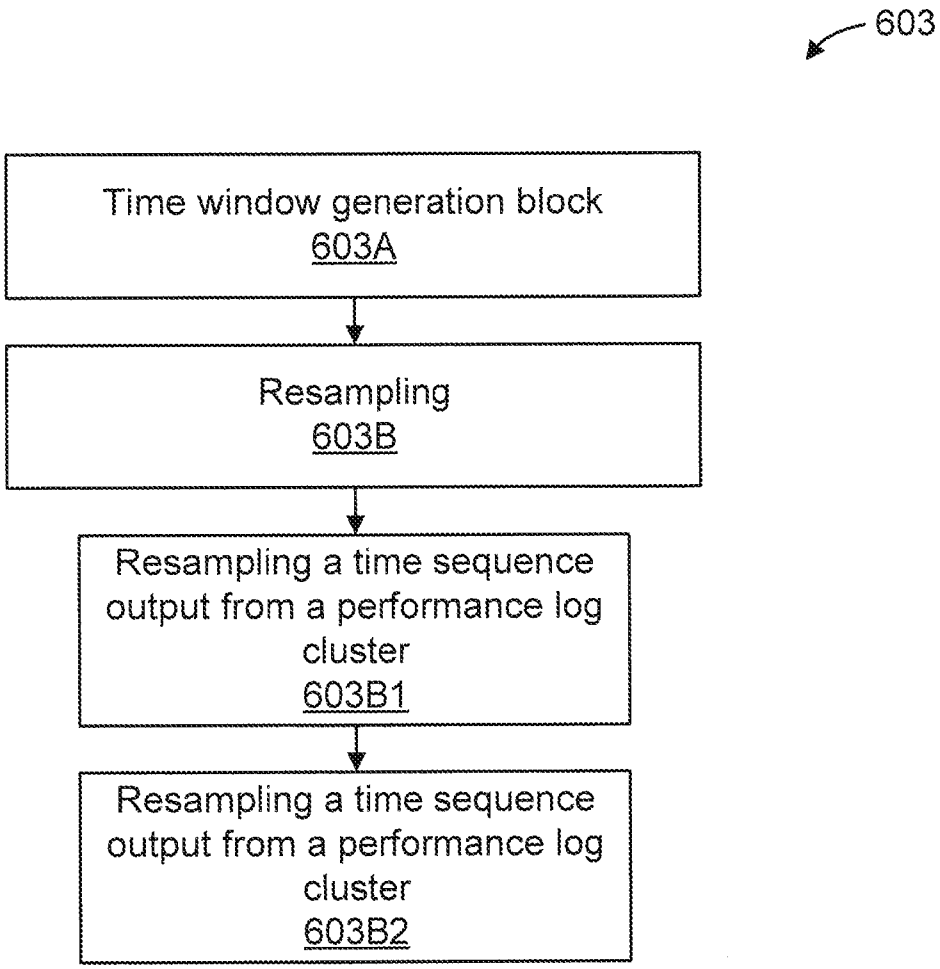
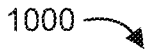

603
Time window generation block
603A
Resampling
603B
Resampling a time sequence output from a performance log cluster
603B1
Resampling a time sequence output from a performance log cluster
603B2

FIG. 9



Text log schema P_1

(X_1 = 2014/04/21 0:01:00, Y_1 = 2)
(X_2 = 2014/04/21 0:02:00, Y_2 = 3)
(X_3 = 2014/04/21 0:03:00, Y_3 = 0)
(X_4 = 2014/04/21 0:04:00, Y_4 = 0)
(X_5 = 2014/04/21 0:05:00, Y_5 = 1)
(X_6 = 2014/04/21 0:06:00, Y_6 = 0)
(X_7 = 2014/04/21 0:07:00, Y_7 = 2)
(X_8 = 2014/04/21 0:08:00, Y_8 = 0)

Performance log schema P_2

(X_1 = 2014/04/21 0:01:00, Y_1 = 40.01)
(X_2 = 2014/04/21 0:02:00, Y_1 = 60.03)
(X_3 = 2014/04/21 0:03:00, Y_1 = 00.01)
(X_4 = 2014/04/21 0:04:00, Y_1 = 00.07)
(X_5 = 2014/04/21 0:05:00, Y_1 = 20.02)
(X_6 = 2014/04/21 0:06:00, Y_1 = 00.01)
(X_7 = 2014/04/21 0:07:00, Y_1 = 40.11)
(X_8 = 2014/04/21 0:08:00, Y_1 = 00.17)

FIG. 10

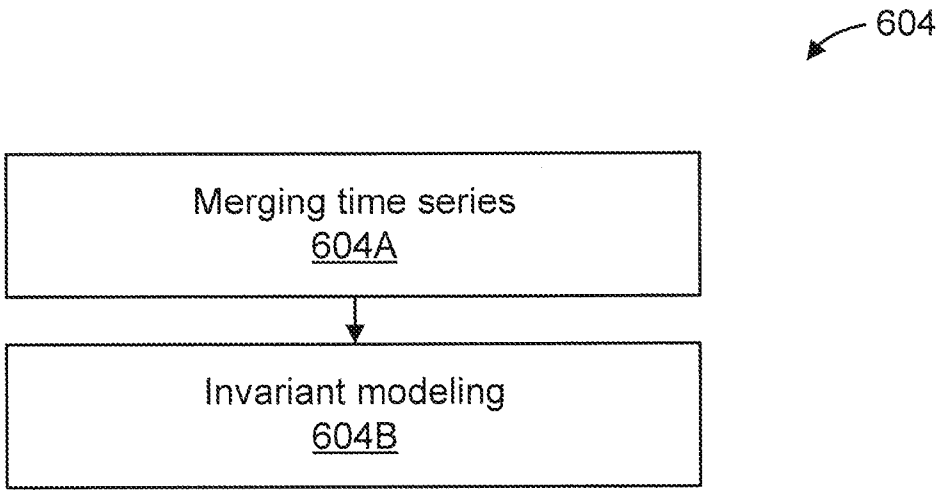


FIG. 11

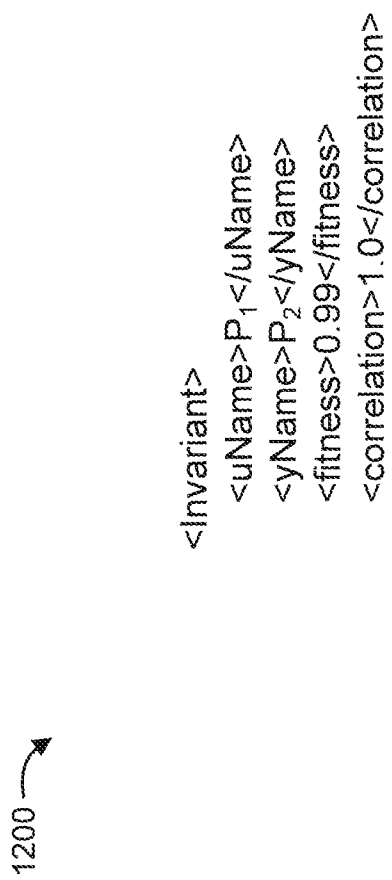

1200
<Invariant>
<uName>P_1</uName>
<yName>P_2</yName>
<fitness>0.99</fitness>
<correlation>1.0</correlation>
</Invariant>

FIG. 12

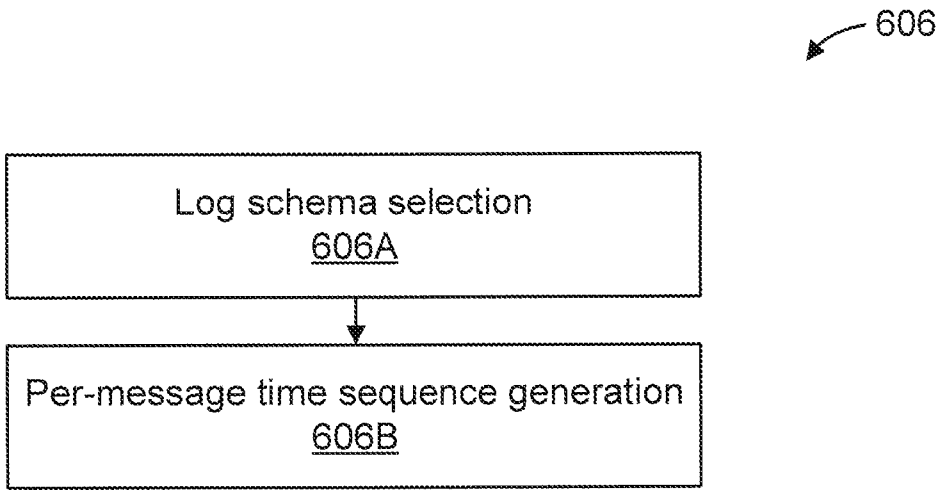

606
Log schema selection
606A
Per-message time sequence generation
606B

FIG. 13

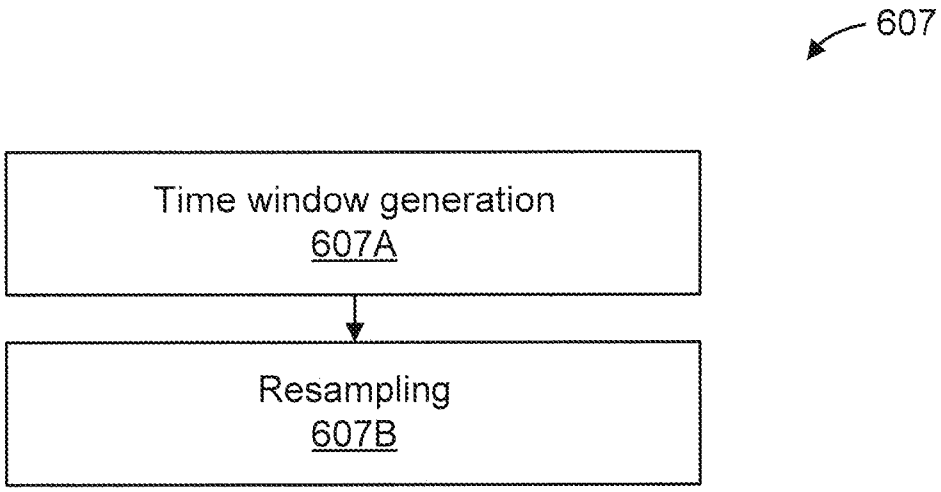


FIG. 14

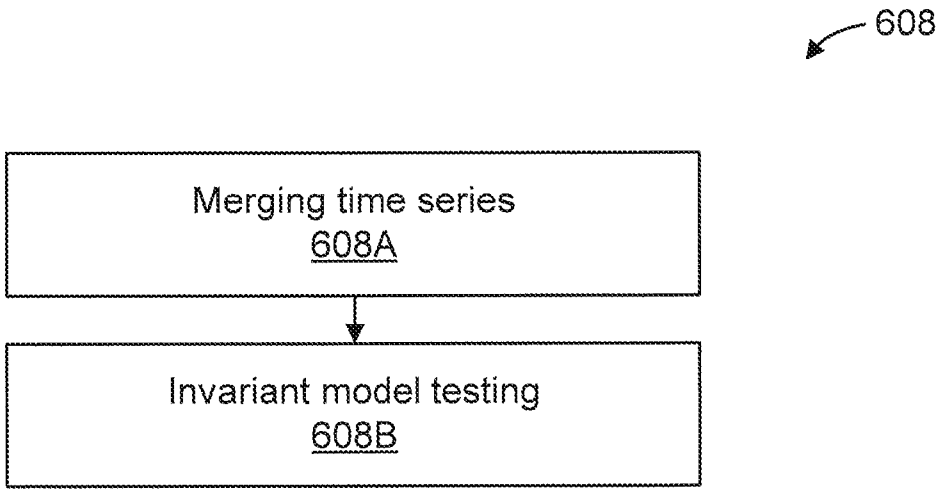


FIG. 15

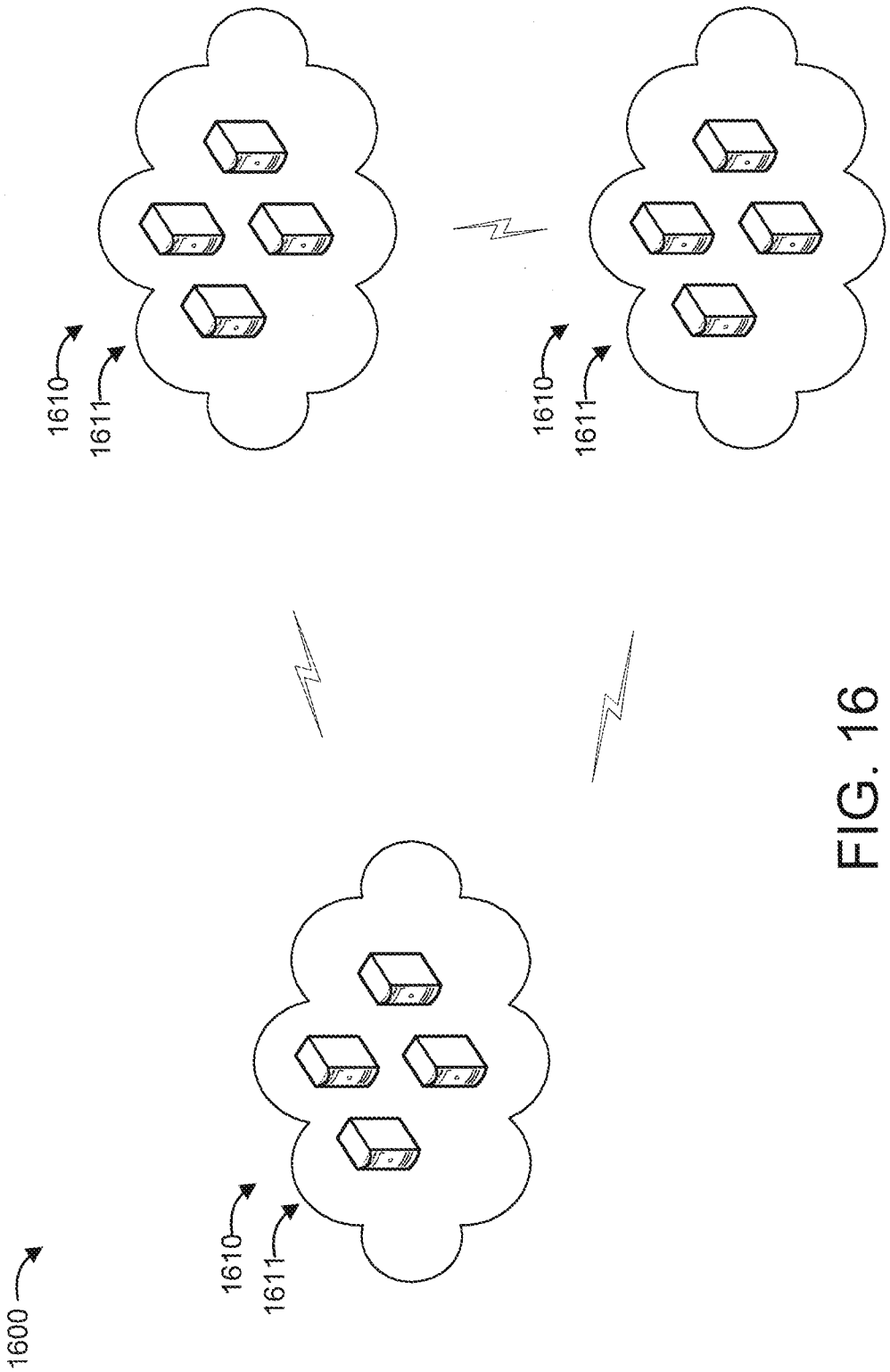


FIG. 16

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/017874

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/34(2006.01)i, G06F 17/30(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 11/34; G06F 11/30; G06F 17/30; G06F 21/55; G06N 5/02

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: heterogeneous logs, log-to-time sequence conversion, time series synchronization, invariant model, anomaly detection, 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.
Y	US 2015-0169681 A1 (MICROSOFT CORPORATION) 18 June 2015 See paragraphs [0019], [0024]-[0025], [0028]-[0029], [0032], and [0041]-[0042]; claim 15; and figures 1-2 and 5-6.	1-20
Y	US 2009-0292954 A1 (GUOFEI JIANG et al.) 26 November 2009 See paragraphs [0032]-[0036], [0039], [0041], [0045], [0050], and [0086]-[0087]; claim 10; and figure 1.	1-20
Y	US 2014-0096249 A1 (CATAPHORA, INC.) 03 April 2014 See paragraphs [0334], [0658], and [0758]-[0766]; claim 49; and figures 4, 17, and 20.	3,14
A	US 2012-0283991 A1 (ADAM J. OLINER et al.) 08 November 2012 See paragraphs [0002], [0009], [0037], [0039], [0058], [0067], [0079], [0086], [0123], and [0161].	1-20

☒ 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
22 May 2017 (22.05.2017)	**22 May 2017 (22.05.2017)**

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

International application No.
PCT/US2017/017874

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	FABIAN MRCHEN, "Time Series Knowledge Mining," Dissertation zur Erlangung des Doktorgrades der Naturwissenschaften, dem Fachbereich Mathematik und Informatik der Philipps-Universitat Marburg, 26 September 2006 Retrieved from <http://www.mybytes.de/papers/moerchen06tskm.pdf> See page 18, lines 23-25; page 22, line 27 - page 23, line 2; page 28, lines 11-12; page 40, lines 21-24; page 42, lines 33-34; and page 120, lines 3-6.	1-20

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2017/017874

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015-0169681 A1	18/06/2015	US 2012-0005220 A1	05/01/2012
		US 8977643 B2	10/03/2015
US 2009-0292954 A1	26/11/2009	EP 2286337 A2	23/02/2011
		EP 2286337 A4	05/11/2014
		JP 2011-521380 A	21/07/2011
		JP 5380528 B2	08/01/2014
		US 8098585 B2	17/01/2012
		WO 2009-142832 A2	26/11/2009
		WO 2009-142832 A3	14/01/2010
US 2014-0096249 A1	03/04/2014	US 2012-0137367 A1	31/05/2012
		US 8887286 B2	11/11/2014
US 2012-0283991 A1	08/11/2012	None	