



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

(21) International Application Number:
PCT/US2018/063722

(22) International Filing Date:
04 December 2018 (04.12.2018)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/594,243 04 December 2017 (04.12.2017) US
62/608,631 21 December 2017 (21.12.2017) US
16/207,644 03 December 2018 (03.12.2018) US

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

(72) Inventors: **XU, Jianwu**; 8 Pond View Lane, Titusville,
New Jersey 08560 (US). **ZONG, Bo**; 28 Hawthorne Drive,
West Windsor, New Jersey 08550 (US). **CHEN, Haifeng**;
11 Blackhawk Court, West Windsor, New Jersey 08550
(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, JO, 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).

(54) Title: EFFICIENT EVENT SEARCHING

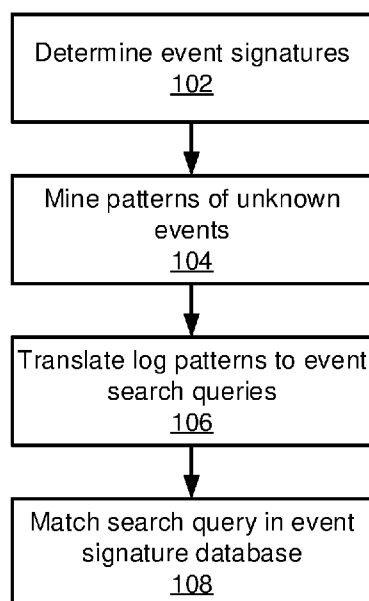


FIG. 1

(57) Abstract: Methods and systems for event detection and correc-
tion include determining a log pattern for a received event. The log pat-
tern is translated to an event search query. The event search query is
weighted according to discriminative dimensions using term- frequency
inverse-document-frequency. The event search query is matched to one
or more known events. A corrective action is automatically performed
based on a solution associated with the one or more known events.



Published:

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

EFFICIENT EVENT SEARCHING

RELATED APPLICATION INFORMATION

[0001] This application claims priority to U.S. Provisional Patent Application No. 62/594,243, filed on December 4, 2017, and to U.S. Patent Provisional Application No. 62,608,631, filed on December 21, 2017, and U.S. Utility Patent Application No. 16/207,644 filed on December 3, 2018 incorporated herein by reference herein its entirety.

BACKGROUND

Technical Field

[0002] The present invention relates to processing logged information and, more particularly, to efficiently searching through large volumes of logged data.

Description of the Related Art

[0003] Computer systems record event information in the form of logs, with events such as system faults, failures, crashes, recovery, intrusion attempts, and normal operating events all producing log information. When an event occurs, logs are generated to indicate what happened before, during, and after the event. Computer forensics experts can then use the logs to investigate the causes of, and solutions to, problems in the system. However, such work can be very time-consuming, particularly in large, enterprise systems, where logs can be particularly voluminous.

[0004] Existing log pattern mining approaches take system logs as input and use data mining algorithms to discover patterns in the logs. Those patterns are then used as models

for anomaly detection and system performance prediction. However, such approaches do not generally provide system event search capabilities.

SUMMARY

[0005] A method for event detection and correction includes determining a log pattern for a received event. The log pattern is translated to an event search query. The event search query is weighted according to discriminative dimensions using term-frequency inverse-document-frequency. The event search query is matched to one or more known events. A corrective action is automatically performed based on a solution associated with the one or more known events.

[0006] A system for event detection and correction includes a pattern mining module configured to determine a log pattern for a received event. An event query module is configured to translate the log pattern to an event search query and to weight the event search query according to discriminative dimensions using term-frequency inverse-document-frequency. A search module includes a processor configured to match the event search query to one or more known events. A correction module is configured to automatically perform a corrective action based on a solution associated with the one or more known events.

[0007] 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

[0008] The disclosure will provide details in the following description of preferred embodiments with reference to the following figures wherein:

[0009] FIG. 1 is a block/flow diagram that shows a method/system for matching unknown events to known events in accordance with an embodiment of the present invention;

[0010] FIG. 2 is a block/flow diagram of a method/system for determining and storing event signatures for known events in accordance with an embodiment of the present invention;

[0011] FIG. 3 is a block/flow diagram of a method/system for mining patterns of known and unknown events in accordance with an embodiment of the present invention;

[0012] FIG. 4 is a block/flow diagram of a method/system for translating log patterns to event search queries in accordance with an embodiment of the present invention;

[0013] FIG. 5 is a block/flow diagram of a method/system for matching unknown events to known events and identifying and implementing a solution to an underlying problem in accordance with an embodiment of the present invention;

[0014] FIG. 6 is a block diagram of system for matching unknown events to known events and identifying and implementing a solution to an underlying problem in accordance with an embodiment of the present invention; and

[0015] FIG. 7 is a block diagram of a processing system in accordance with an embodiment of the present invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

[0016] Embodiments of the present invention provide efficient query and retrieval of log information based on log pattern analytics and system event search. The present embodiments provide a translation between the logic of log pattern mining and the processes of the system event search, with results being rendered in a user-friendly format. The present embodiments can thereby automatically detect, diagnose, and correct system faults, providing an improvement to the technical fields of log analysis and automated system management.

[0017] Referring now in detail to the figures in which like numerals represent the same or similar elements and initially to FIG. 1, a method for searching event logs is illustratively depicted in accordance with one embodiment of the present invention. Block 102 determines a set of event signatures based on training data, in a manner that will be described in fuller detail below. The training data is made up of previously recorded events and may include events from multiple different sets of logs that represent detailed, system-level information. The logs may share a format or may, alternatively, be heterogeneous. The logs may include text-based log messages and time series data. The signatures include vector representations of the events in a multi-dimensional space, where similar events are positioned closer to one another in the space than dissimilar events.

[0018] Block 104 mines patterns in a set of unknown events. For example, during runtime, new sets of logs are constantly being generated and processed. Block 104 identifies patterns within these new events using log analytics. The patterns that are found can include, for example, sequential patterns, periodic patterns, and correlation patterns.

[0019] Block 106 translates the output of the log analysis to an event search query. Block 108 performs an event search, matching the event search query to the database of signatures. The event search weights the importance of different dimensions in the vector representation of the logs according to, e.g., a term frequency, inverse document frequency (TF-IDF) scheme. Block 108 determines the similarity between events in the multi-dimensional space using any appropriate metric, such as the cosine similarity. In general, only some of the dimensions in the vector representation of the logs will be truly discriminative. The weighting identifies the useful dimensions by putting higher weights on those dimensions that are more discriminative.

[0020] The operation of block 108 includes a first step of term frequency normalization. Given log patterns that belong to a given pattern category, per-pattern counts are normalized by the total number of counts for the category. For example, if there are four patterns (P1, P2, P3, P4), and a particular event matches the patterns with counts of 1, 2, 3, and 4, respectively, then the vector representation of the event can be $\langle 1, 2, 3, 4 \rangle$. After normalization, the vector representation becomes $\langle 0.1, 0.2, 0.3, 0.4 \rangle$.

[0021] Inverse document frequency weights are then calculated based on the vector representation of the known events. If N is the number of known events, then x_i is the vector representation of a system event I , and $x_i(j)$ is the value of x_i in its j^{th} dimension. The weight of the j^{th} dimension is determined as:

$$idf(j) = \frac{N}{|\{i | x_i(j) > 0\}|}$$

These weights are then multiplied against the vector representation of the event, with the values of the vector being weighted as $x'_i(j) = x_i(j)idf(j)$. The TF-IDF weighted vector representation of each known event is stored and is used for system event search queries.

[0022] Once one or more similar events have been determined by block 108, the results are represented as objects that include, for example, a relevance field, a data label field, and an “other” field. The relevance field provides a key/value tuple, where the key is an identifier of a known event and the value is the similarity between the search query and the known event, with a value of zero representing the minimum similarity and a value of one representing the maximum similarity. The data label field indicates the event identifier of the unknown event that makes up the search query. The “other” field records the signature of the unknown event. All matched results are stored and can be used for, e.g., ranking, filtering, visualization, and other appropriate actions.

[0023] Referring now to FIG. 2, additional detail on the determination of event signatures 102 is shown. Block 202 collects system logs. It is contemplated that the system logs can be collected from any of a wide variety of computer systems and sensors, including software application self-report, operating system logging, hardware sensor monitor logs, and logs received from external sources, but it should be understood that any appropriate logging system may be used to generate the logs. Block 204 mines patterns of known events from the collected logs, in the manner described below, using log analytics.

[0024] Block 206 translates the log patterns that have been discovered by block 204 into search queries suitable for event searching and block 208 uses the search queries to generate event signatures. The event signatures can be a vector representation of the events in an appropriate space. Block 210 stores the generated event signatures in a signature database to be used for subsequent event search queries. The signatures can be stored as objects having keys that are pattern identifiers and values that are the signature

values. These objects are stored in an appropriate database, with event signatures being indexed for high-efficiency searches.

[0025] Referring now to FIG. 3, additional detail on the mining of patterns of known events in blocks 104 and 204 is shown. Block 302 identifies user requests for the generation of event signatures in a training process. In one exemplary embodiment, the training request can include such information as instructions for how to access input logs, details for accessing a signature database, identification of predetermined pattern types, and a requested action (e.g., “train”). Using the training request, block 304 performs log pattern mining to generate a set of log patterns that appear in the known events.

[0026] In some embodiments, blocks 104/204 can identify both text and time-series based log patterns using analytical techniques to discover a variety of text-based patterns such as, e.g., schema patterns, sequential patterns, and periodic patterns. Time-series based log patterns can be mined using analytical techniques to discover time-series based patterns such as correlation patterns and burst patterns. Thus, a set of log patterns with a variety of pattern identifiers that appear in the known events, as well as the occurrence frequency of the patterns in the known events, can be obtained.

[0027] In other embodiments, block 104 accepts user controls regarding how input logs for known events can be accessed, the details of the signature database, the types and categories of log patterns to extract, and the type of request.

[0028] Referring now to FIG. 4, additional detail on the translation of log patterns to event search queries in blocks 106 and 206 is shown. Block 402 collects the output of the log analysis in blocks 104/204. This output can include a number of objects, each indicating the occurrence of a specific log pattern. Block 404 extracts information from

the collected objects relating to the event search, including, for example, log pattern type, pattern identifier, and pattern-type-specific information. Block 406 then generates an event search query using this extracted information. This translation step decouples the logic of the log pattern analysis from the event searching system. In this manner, each can be updated independently from the other, making it so that the tools can be separately maintained.

[0029] Referring now to FIG. 5, a method for finding and implementing solutions to problems in a monitored computer system is shown. Block 502 determines events of interest. For example, the events of interest can relate to a system fault such as, e.g., a non-responsive application or hardware component, slow responses from an application or hardware component, incorrect responses from an application or hardware component, or a suspected network intrusion, or can relate to a high-risk condition that can lead to a system fault such as, e.g., an dangerous environmental condition, wear and tear on components, maintenance events, etc. The determination of events of interest can be performed by a human operator or can, alternatively, be performed automatically by recognizing events that are out of the ordinary.

[0030] Block 504 then performs a search of a known events database as described above to identify one or more matching events. These matching events are tagged in the manner above to describe what type of event they represent. Block 506 then identifies a solution corresponding to the matched events. This solution can be stored with the matching events in the events database or can, alternatively, be stored in an external database that can be searched for particular solution information.

[0031] Block 508 then automatically implements the solution to repair or mitigate the underlying problem. Exemplary corrective actions include changing a security setting for an application or hardware component, halting and/or restarting an application, halting and/or rebooting a hardware component, changing an environmental condition, changing a network interface's status or settings, etc. In this manner, embodiments of the present invention can automatically respond to conditions and implement predetermined solutions to existing or emerging problems. This provides a substantial improvement in automated system monitoring and administration by efficiently accessing solution information, making it possible to respond to developing conditions more rapidly than would be possible by a human operator.

[0032] 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.

[0033] 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 storage 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.

[0034] Each computer program may be tangibly stored in a machine-readable storage media or device (e.g., program memory or magnetic disk) readable by a general or special purpose programmable computer, for configuring and controlling operation of a computer when the storage media or device is read by the computer to perform the procedures described herein. The inventive system may also be considered to be embodied in a computer-readable storage medium, configured with a computer program, where the storage medium so configured causes a computer to operate in a specific and predefined manner to perform the functions described herein.

[0035] A data processing system suitable for storing and/or executing program code may include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code to reduce the number of times code is retrieved from bulk storage during execution. Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) may be coupled to the system either directly or through intervening I/O controllers.

[0036] Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modem and Ethernet cards are just a few of the currently available types of network adapters.

[0037] Referring now to FIG. 6, an event detection, matching, and correction system 600 is shown. The system 600 includes a hardware processor 602 and a memory 604. A network interface 606 provides communications to other systems via a wired or wireless communications medium and any appropriate protocol. The system 600 furthermore includes one or more functional modules that may, in some embodiments, be implemented as software that is stored in memory 604 and that is executed by hardware processor 602. In other embodiments, some or all of the functional modules can be implemented in the form of one or more discrete hardware components in the form of, e.g., application-specific integrated chips or field programmable gate arrays.

[0038] The memory 604 stores a known event database that includes event signatures for a set of previously encountered events. The memory 604 also stores event logs 610. The event logs 610 are collected by one or more logging applications or hardware components that generate information in a continuous, periodic, or sporadic manner regarding monitored software or hardware components. This information can be generated locally to the system 600 or can, alternatively, be collected at other systems and then transmitted to the system 600 via the network interface 606.

[0039] An event signature module 612 generates event signatures from event logs 610 to be stored in the known event database 608. During runtime, the event signature module 612 generates event signatures for incoming event logs to be used to match against event signatures that are already in the known event database 608. A pattern mining module 614 performs analysis of the logs to determine log patterns that the event signature module uses to create the event signatures.

[0040] As new events arrive, event query module 616 takes the output of the pattern mining module 614 to generate search queries that represent the new patterns. Search module 618 then uses the search query to compare the new event to known events in the known event database 608. If the search module 618 finds a match (e.g., if a similarity score between the new event and a known event is above a threshold according to an appropriate similarity metric), then block 620 searches for a solution that matches the event. Correction module 622 then automatically performs a corrective action dictated by the solution module 620 to correct or mitigate a problem that caused the new event.

[0041] Referring now to FIG. 7, an exemplary processing system 700 is shown which may represent the transmitting device 100 or the receiving device 120. The processing system 700 includes at least one processor (CPU) 704 operatively coupled to other components via a system bus 702. A cache 706, a Read Only Memory (ROM) 708, a Random Access Memory (RAM) 710, an input/output (I/O) adapter 720, a sound adapter 730, a network adapter 740, a user interface adapter 750, and a display adapter 760, are operatively coupled to the system bus 702.

[0042] A first storage device 722 and a second storage device 724 are operatively coupled to system bus 702 by the I/O adapter 720. The storage devices 722 and 724 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 722 and 724 can be the same type of storage device or different types of storage devices.

[0043] A speaker 732 is operatively coupled to system bus 702 by the sound adapter 730. A transceiver 742 is operatively coupled to system bus 702 by network adapter 740. A display device 762 is operatively coupled to system bus 702 by display adapter 760.

[0044] A first user input device 752, a second user input device 754, and a third user input device 756 are operatively coupled to system bus 702 by user interface adapter 750. The user input devices 752, 754, and 756 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 752, 754, and 756 can be the same type of user input device or different types of user input devices. The user input devices 752, 754, and 756 are used to input and output information to and from system 700.

[0045] Of course, the processing system 700 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 700, 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 700 are readily contemplated by one of ordinary skill in the art given the teachings of the present principles provided herein.

[0046] The foregoing is to be understood as being in every respect illustrative and exemplary, but not restrictive, and the scope of the invention disclosed herein is not to be determined from the Detailed Description, but rather from the claims as interpreted according to the full breadth permitted by the patent laws. It is to be understood that the

embodiments shown and described herein are only illustrative of the present invention and that those skilled in the art may implement various modifications without departing from the scope and spirit of the invention. Those skilled in the art could implement various other feature combinations without departing from the scope and spirit of the invention. 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 for event detection and correction, comprising:
determining a log pattern for a received event;
translating the log pattern to an event search query;
weighting the event search query according to discriminative dimensions using term-frequency inverse-document-frequency (TF-IDF);
matching the event search query to one or more known events; and
automatically performing a corrective action based on a solution associated with the one or more known events.
2. The method of claim 1, wherein translating forms a vector representation of the log pattern as part of the event search query.
3. The method of claim 2, wherein matching the event search query to one or more known events comprises determining whether the vector representation of the log pattern is within a threshold value of a vector representation of a stored known event according to a similarity metric.
4. The method of claim 3, wherein the similarity metric is a cosine similarity that measures a similarity between the vector representation of the log pattern and the vector representation of the stored known event.

5. The method of claim 1, wherein translating the log pattern into an event query comprises extracting information from the log pattern and rendering the extracted information in the format of an event search query.

6. The method of claim 5, wherein the extracted information comprises log pattern type, a pattern identifier, and pattern-type-specific information.

7. The method of claim 5, wherein translating the log pattern into an event query decouples the determination of the log pattern from matching the event search query to one or more known events.

8. The method of claim 1, further comprising determining the solution associated with the one or more known events by searching a solutions database using the matched one or more known events.

9. The method of claim 1, wherein matching the event search query to the one or more known events generates an output that includes an identifier of the one or more known events, a similarity score that identifies a similarity between the event search query and the one or more known events, and an identifier for the received event.

10. The method of claim 1, wherein the corrective action includes one or more actions selected from the group consisting of changing a security setting for an application or hardware component, halting and/or restarting an application, halting

and/or rebooting a hardware component, changing an environmental condition, and changing a network interface's status or settings.

11. A system for event detection and correction, comprising:

- a pattern mining module configured to determine a log pattern for a received event;
- an event query module configured to translate the log pattern to an event search query and to weight the event search query according to discriminative dimensions using term-frequency inverse-document-frequency (TF-IDF);
- a search module comprising a processor configured to match the event search query to one or more known events; and
- a correction module configured to automatically perform a corrective action based on a solution associated with the one or more known events.

12. The system of claim 11, wherein the event query module is further configured to form a vector representation of the log pattern as part of the event search query.

13. The system of claim 12, wherein the search module is further configured to determine whether the vector representation of the log pattern is within a threshold value of a vector representation of a stored known event according to a similarity metric.

14. The system of claim 13, wherein the similarity metric is a cosine similarity that measures a similarity between the vector representation of the log pattern and the vector representation of the stored known event.

15. The system of claim 11, wherein the event query module is further configured to extract information from the log pattern and rendering the extracted information in the format of an event search query.

16. The system of claim 15, wherein the extracted information comprises log pattern type, a pattern identifier, and pattern-type-specific information.

17. The system of claim 15, wherein the event query module is further configured to decouple operation of the pattern mining module from operation of the search module.

18. The system of claim 11, further comprising a solution module configured to determine the solution associated with the one or more known events by searching a solutions database using the matched one or more known events.

19. The system of claim 11, wherein the search module is further configured to generate an output that includes an identifier of the one or more known events, a similarity score that identifies a similarity between the event search query and the one or more known events, and an identifier for the received event.

20. The system of claim 11, wherein the correction module is configured to automatically perform a corrective action selected from the group consisting of changing a security setting for an application or hardware component, halting and/or restarting an application, halting and/or rebooting a hardware component, changing an environmental condition, and changing a network interface's status or settings.

1/6

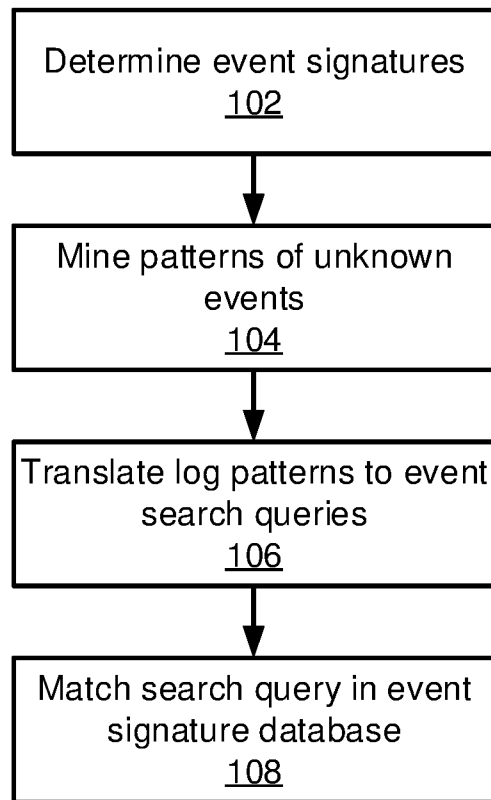


FIG. 1

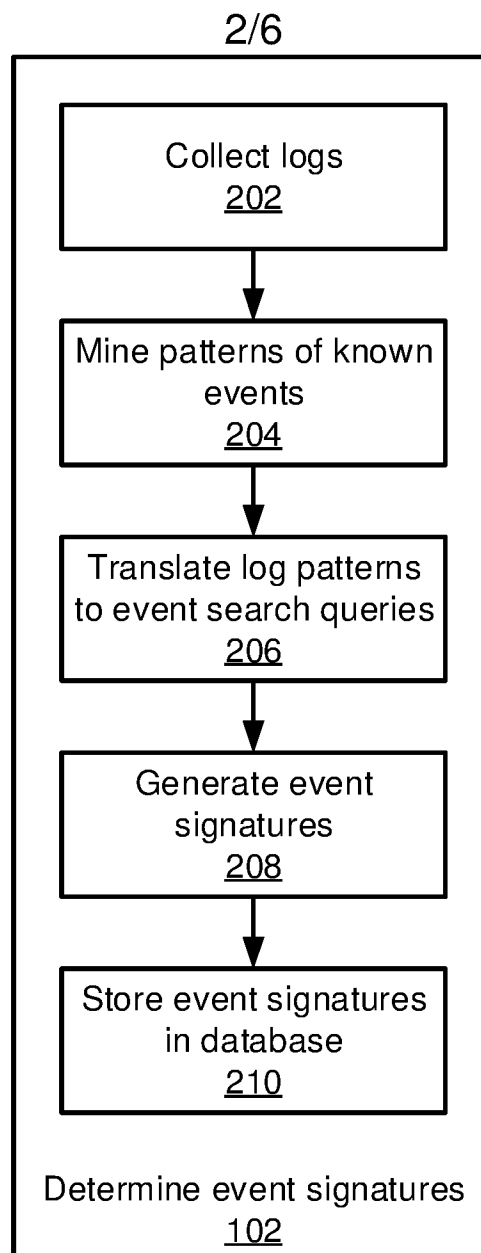


FIG. 2

3/6

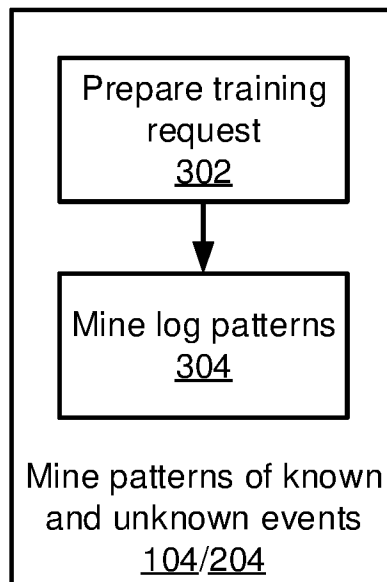


FIG. 3

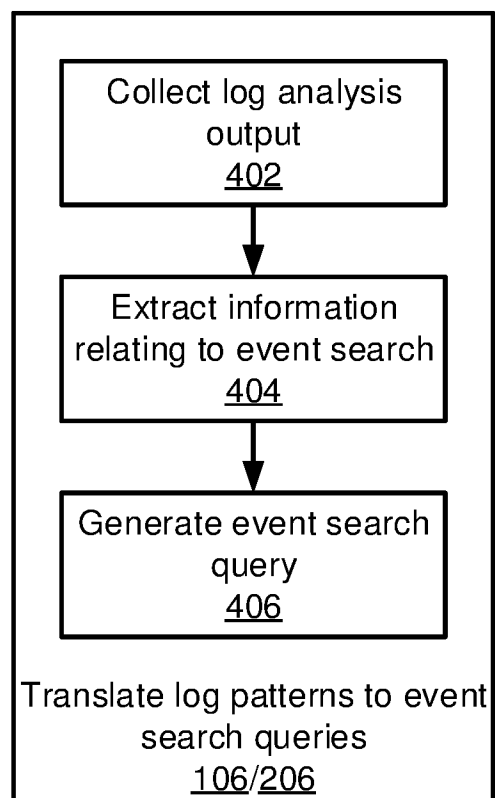


FIG. 4

4/6

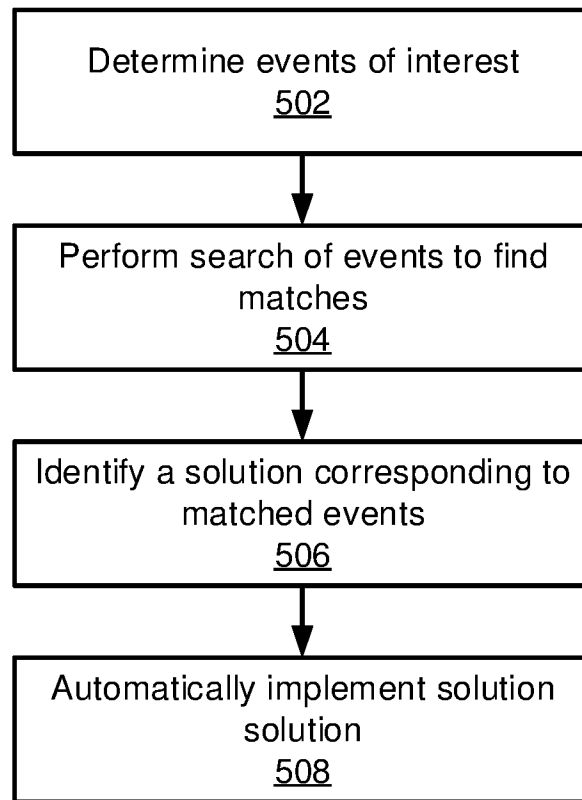


FIG. 5

5/6

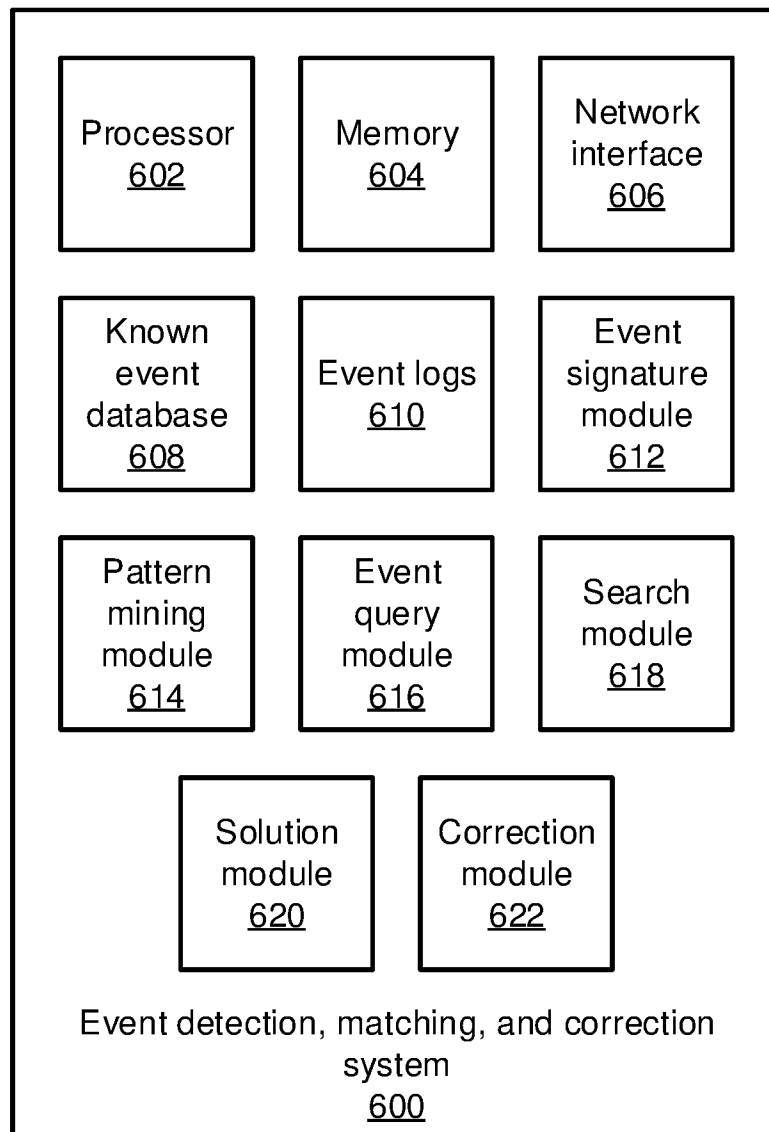


FIG. 6

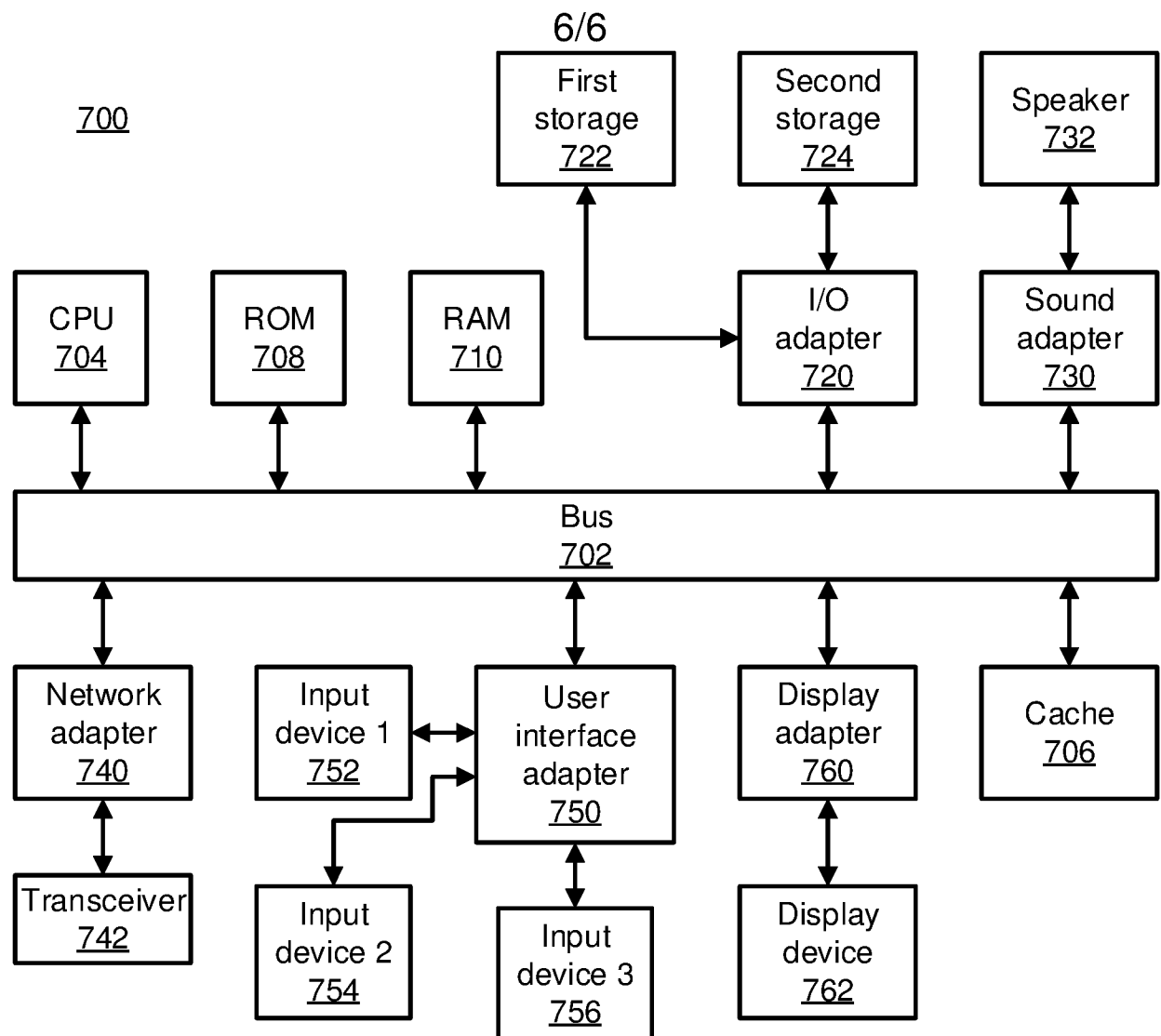


FIG. 7

A. CLASSIFICATION OF SUBJECT MATTER**G06F 11/34(2006.01)i**

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

B. FIELDS SEARCHEDMinimum documentation searched (classification system followed by classification symbols)
G06F 11/34; G06F 11/00; G06F 11/16; G06F 11/22; G06F 11/30; G06F 11/36; G06F 15/00Documentation 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 modelsElectronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & keywords: event, log pattern, search query, TF-IDF, match, corrective action**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 6643801 B1 (VINAY BHASKAR JAMMU et al.) 04 November 2003 See column 4, lines 8-11, 58-61; column 7, lines 60-65; and figure 1.	1-20
Y	US 2017-0116114 A1 (MICROSOFT TECHNOLOGY LICENSING, LLC) 27 April 2017 See paragraphs [0038], [0054], [0056], [0058], [0063], [0067], [0071], [0136], [0139], [0141], [0145]; table 1; and figures 3-4, 16-18.	1-20
Y	JP 2003-085003 A (MATSUSHITA ELECTRIC IND CO., LTD.) 20 March 2003 See paragraph [0023]; and figures 5, 12.	10, 20
A	KR 10-2014-0129676 A (INKA ENTWORKS INC. et al.) 07 November 2014 See paragraphs [0040]-[0045]; and figure 3.	1-20
A	JP 2014-078067 A (NEC CORP.) 01 May 2014 See paragraphs [0022]-[0066]; and figures 1-12.	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

18 March 2019 (18.03.2019)

Date of mailing of the international search report

18 March 2019 (18.03.2019)

Name and mailing address of the ISA/KR

International Application Division

Korean Intellectual Property Office

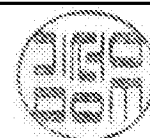
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea

Facsimile No. +82-42-481-8578

Authorized officer

KIM, Seong Woo

Telephone No. +82-42-481-3348



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2018/063722

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6643801 B1	04/11/2003	AU 2000-80320 A1 CA 2387927 A1 CA 2387927 C WO 01-31449 A1	08/05/2001 03/05/2001 10/10/2006 03/05/2001
US 2017-0116114 A1	27/04/2017	US 9710371 B2 WO 2017-074770 A1	18/07/2017 04/05/2017
JP 2003-085003 A	20/03/2003	None	
KR 10-2014-0129676 A	07/11/2014	None	
JP 2014-078067 A	01/05/2014	JP 06070040 B2	01/02/2017