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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
07 May 2020 (07.05.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/091589 A1

(51) International Patent Classification:
G06F 11/34 (2006.01) *G06F 11/07* (2006.01)

(21) International Application Number:
PCT/MY2019/050073

(22) International Filing Date:
15 October 2019 (15.10.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
PI 2018001829 30 October 2018 (30.10.2018) MY

(71) Applicant: **MIMOS BERHAD** [MY/MY]; Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(72) Inventors: **DAUD, Nor Izyani Binti**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **HARON, Galoh Rashidah Binti**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY). **DIN, Dahlia Binti**; Mimos Berhad, Technology Park Malaysia, Kuala Lumpur, 57000 (MY).

(74) Agent: **SU, Siew Ling**; C/o Zaid Ibrahim & CO, Level 19, Menara Milenium, Jalan Damanlela, Pusat Bandar Damansara, Kuala Lumpur, 50490 (MY).

(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: METHOD AND SYSTEM FOR MONITORING USER ACCESS TO APPLICATION SERVER

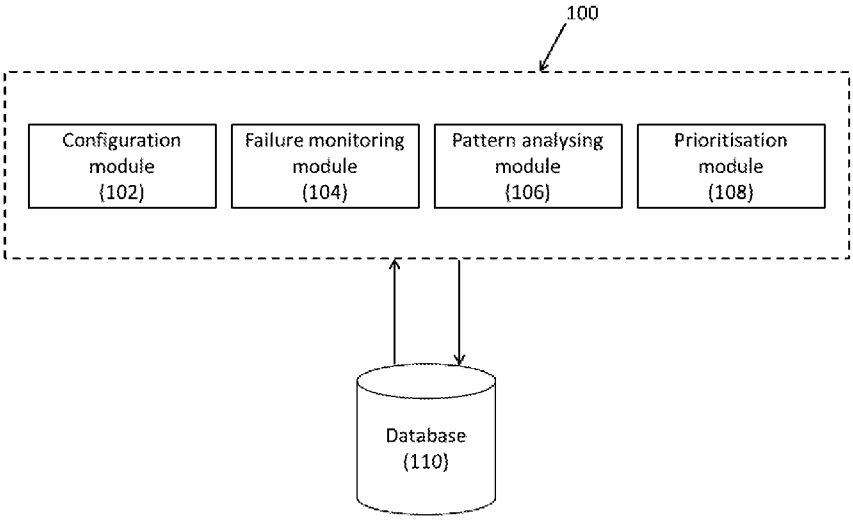

Figure 2

(57) Abstract: Method And System For Monitoring User Access To Application Server The invention provides a method and system (100) for monitoring access to an application server by one or several users. The method and system (100) involve monitoring failure tables associated to the users who access the application server, and then subjecting each failure record that is newly added to the failure tables to a pattern analysis. After completing the pattern analysis, a failure report may be generated for each of the new failure records. The failure reports may also be prioritised before being sent to the respective users or server administrator.

WO 2020/091589 A1

[Continued on next page]

Declarations under Rule 4.17:

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

Published:

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

— *in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE*

METHOD AND SYSTEM FOR MONITORING USER ACCESS TO APPLICATION SERVER

Field of Invention

The invention relates to application management. In particular, the invention relates to monitoring and analysis of a user failed attempt to access a target application server.

Background of the Invention

Various external users or clients may access a plurality of services through connection to servers providing these services through one or more networks, for instance a local area network or a wide area network (or internet). Nevertheless, the services may fail for a wide variety of reasons, such as code errors or bugs, user errors, bad or incorrect input data, user authentication failures, unavailable resources, etc. Such failures may result in data losses and service downtime. Further, it may also incur costs and time of recovery, as it is required to manually verify and identify the failures and their causes from a large amount of reports generated by a plurality of components or sub-modules running in providing the services.

To mitigate the above shortcomings, various methods and systems are available in the art for analysing a failure (or error) and also providing an alert when the failure arises. For instance, U.S. Patent No. 7,890,814 (B2) discloses an apparatus and a system for accessing error report information. The apparatus and system involve techniques and tools for analysing and interrelating failure data contained in error reports and thereby facilitating developers to more easily and quickly solve programming bugs. Numerous parameters may also be specified for selecting and searching the error reports. Several reliability metrics are provided to better track software reliability situations, and the reliability metrics also facilitate the tracking of the overall situations of failures that happen by providing metrics based on the error reports (e.g. failure occurrence trends, failure distributions across different languages).

U.S. Patent No. 9,189,308 (B2) also discloses a computer implemented method and a system for differentiating normal operation of an application programme from error conditions so as to predict, diagnose and recover from application failures. Access to resources by the application programme is monitored. Resource access events are also logged, wherein resource access patterns are established based on the logged resource access events using computer pattern recognition techniques. If subsequent access to resources by the application programme deviates from the established patterns, one or both of a user and a system administrator of the application programme is/are notified of a potential error condition based on the detected deviation. In addition, sequences of resource access events that deviate from the established resources access patterns are correlated with an error condition based upon a temporal proximity to the time of occurrence of the error to provide diagnostic information regarding the error.

Although numerous methods and systems have been provided in the art for analysing a failure and providing an alert when such failure arises, they however often involve a series of operations in identifying and verifying the failure before sending an alert to a user or system administrator. Further, the methods and systems available in the art are configured to operate using a predetermined set of parameters in analysing the failure, thereby rendering the failure analyses less flexible, in particular when the failure type varies from one to another.

Therefore, there exists a need to provide a method and system for analysing a failure more dynamically and also providing an automated report on such failure.

Summary of the Invention

One of the objects of the invention is to introduce a method and system provided with a dynamic configuration that may enable modifications to metrics involved in various analyses and operations, unlike the prior art that adopts static configuration or uses fixed parameters in monitoring and analysing operations.

Another object of the invention is to provide a method and system for monitoring user

access attempt to an application server and also capable of providing quick responses or alert notifications to a user and server administrator when a failed access attempt is detected. Quick responses or notifications can be provided, as only simple monitoring and analyses are performed, instead of complicated operations.

Yet another object of the invention is to introduce a method and system provided with a dynamic configuration that enables different metrics to be selected for generating various types of failure reports.

At least one of the preceding objects is met, in whole or in part, by the invention, in which one of the embodiments of the invention describes a method for monitoring a user access to an application server, comprising the steps of detecting a variation in a total number of failure records in a user-specific failure table, wherein each of the failure records is indicative of a historical failed attempt by the user to access the application server; subjecting a new failure record to a pattern analysis to generate a failure report, if there is an incremental change in the total number of failure records in the failure table; and prioritising the generated failure report based on an urgency priority level that defines a sequence for sending the report to the user or server administrator, where the urgency priority level is determined based on a prioritisation table.

Preferably, the pattern analysis may comprise at least one of extracting data associated with a plurality of metrics from the new failure record and comparing the extracted data to user historical behavioural data before generating the failure report.

In some preferred embodiments, the method may also comprise the step of selecting the metrics for the pattern analysis based on attributes of the failure record.

The method may further comprise the step of selecting the metrics to be included in the failure report and sent to the user or server administrator.

A further embodiment of the invention is a system for monitoring a user access to an

application server, comprising a failure monitoring module for detecting a variation in a total number of failure records in a user-specific failure table, wherein each of the failure records is indicative of a historical failed attempt by the user to access the application server; a pattern analysing module for receiving a new failure record from the failure monitoring module if there is an incremental change in the total number of failure records in the failure table, and subjecting the new record to a pattern analysis for generating a failure report; and a prioritisation module for determining an urgency priority level for the generated failure report by referring to a prioritisation table, and subsequently prioritising the generated failure report to be sent to the user based on the determined urgency priority level.

In further embodiments, the pattern analysing module may be configured to perform the pattern analysis by extracting data associated with a plurality of metrics from the new failure record, comparing the extracted data to user historical behavioural data before generating the failure report or both.

The system may further comprise a configuration module for selecting the metrics for the pattern analysis based on attributes of the failure record.

In some further embodiments, the configuration module may be further configured to select the metrics to be included in the failure report and sent to the user or server administrator.

One skilled in the art will readily appreciate that the invention is well adapted to carry out the aspects and obtain the ends and advantages mentioned, as well as those inherent therein. The embodiments described herein are not intended as limitations on the scope of the invention.

Brief Description of Drawings

For the purpose of facilitating an understanding of the invention, there is illustrated in the accompanying drawing the preferred embodiments from an inspection of which

when considered in connection with the following description, the invention, its construction and operation and many of its advantages would be readily understood and appreciated.

Figure 1 is a network architecture involving a system (100) for monitoring user access to an application server, according to one preferred embodiment of the invention.

Figure 2 is a general architecture of a system (100) for monitoring user access to an application server, according to another preferred embodiment of the invention.

Figure 3 is a general flow chart of a method for monitoring user access to an application server, according to still another preferred embodiment of the invention.

Figure 4 is a flow chart showing a first stage (202) of the method shown in Figure 3, according to an exemplary embodiment of the invention.

Figure 5 is a flow chart showing a second stage (204) of the method shown in Figure 3, according to an exemplary embodiment of the invention.

Figure 6 is a flow chart showing a third stage (206) of the method shown in Figure 3, according to an exemplary embodiment of the invention.

Figure 7 is a flow chart showing a fourth stage (208) of the method shown in Figure 3, according to an exemplary embodiment of the invention.

Detailed Description of the Invention

Hereinafter, the invention shall be described according to the preferred embodiments of the invention and by referring to the accompanying description and drawings. However, it is to be understood that limiting the description to the preferred embodiments of the invention and to the drawings is merely to facilitate discussion of the invention and it is envisioned that those skilled in the art may devise various

modifications without departing from the scope of the appended claim.

The invention provides a system and method for monitoring access by one or multiple users to an application server. In particular, the system and method enable a failed or unsuccessful attempt to gain access to the application server to be recorded and analysed, so that a failure report can be generated and sent to respective users, server administrator or both. The system and method may also enable dynamic configuration to customise or modify metrics relating to the failed attempts for numerous operations including pattern analyses and failure report generation.

According to a preferred embodiment of the invention, the system (100) for monitoring access to an application server by one or multiple users may be involved in a network architecture as illustrated in Figure 1. In particular, a user may make a login attempt to access the application server (or the application), as indicated by arrow 1 in Figure 1. It may be essential that the application server ensures or verifies that the user has a valid trust level before allowing or granting access to the application server to such user. As such, the application server may pass the information or details associated with the user to an adaptive system, as shown by arrow 2. As shown by arrow 3, the adaptive system may check against a database (110) where information relating to trust level is stored, so as to ensure that the user has a valid trust. Such information may then be transmitted to the application server, as illustrated by arrow 4, 5 and 6. As indicated by arrow 7 and 8 in Figure 1, the system (100) may also monitor user access to the application server concurrently, when the application server grants or does not grant an access to the user based on the trust level information received from the database (110).

Figure 2 shows a general architecture of the system (100) for monitoring access to an application server by one or multiple users, according to one preferred embodiment of the invention. The system (100) may also be configured in some of the embodiments to communicate with the database (110) shown in Figure 1 (or more databases) for storing part or all results produced by the system (100) or its components to the database (110). Further, the system (100) may be configured so that it concurrently

monitors access to the application server by multiple users. In particular, the system (100) may detect several failed attempts by multiple users concurrently within a time frame. The system (100) may also be able to analyse each of the failed attempts based on the respective user historical behavioural patterns (or in particular, the patterns that are obtained and generated based on the user login behaviours before the current login attempt), and also generate a failure report for each of the users. However, for ease of description, the exemplary embodiments described herein are limited to monitoring of access to a target application server by a single user, but it should be appreciated that the invention should not be limited therein or thereto.

As illustrated in Figure 2, the system (100) may comprise a failure monitoring module (104) that may be configured to detect a variation in a total number of failure records in a user-specific failure table. In particular, when a user fails to gain access to a target application server, such failed attempt may be identified, captured and recorded by the failure monitoring module (104). As the user failed attempt may be recorded as a new failure record by the failure monitoring module (104), it should be appreciated that the failure record may generally be referred to as data indicative of the user failed attempt to access the application server. It should also be appreciated that such failure record may relate to only a user and the failure record may comprise information associated with a plurality of metrics pertaining to the failed attempt. In more particular, each of these metrics may be related to an attribute of the failed attempt.

In some embodiments, the failure monitoring module (104) may also be configured to transfer the new failure record to a user-specific failure table. As mentioned above, the new failure record may relate to only a particular user and so, it may be transferred and added to the failure table associated with the particular user, in which the failure table may be stored in the database (110) or failure monitoring module (104) in some embodiments. Additionally, the failure table may be configured to be user-specific and so, there may be more than one failure table, each being associated with a particular user. It should also be appreciated that each failure table may comprise a plurality of failure records, each being indicative of a failed access attempt occurred before the new failure record is captured (or referred to as “historical failed attempt”).

Accordingly, addition of a new failure record to the failure table may result in an incremental change and so, the failure monitoring module (104) may be configured in certain embodiments to detect such variation in the total number of failure records stored or maintained in the failure table.

Additionally, as the failure monitoring module (104) may be required to monitor only the variation in the total number of failure records in the user-specific failure table, the simple monitoring process performed by the failure monitoring module (104) could be reduced to a background process. Therefore, the failure monitoring module (104) may be configured to perform the monitoring process continuously in the background.

In circumstances where the failure monitoring module (104) may detect several failed attempts made by multiple users to the application server, it should be appreciated that the failure monitoring module (104) may be capable of identifying the users from the failed attempts, before transferring these failure records to the respective user-specific failure tables. The failure monitoring module (104) may also be capable of monitoring variation in each of the failure records concurrently.

The system (100) may also comprise a pattern analysing module (106) for subjecting the new failure record to a pattern analysis before generating a failure report. In more specific, the pattern analysing module (106) may be in communication with the failure monitoring module (104), so that the pattern analysing module (106) may receive the new failure record from the failure monitoring module (104) when detecting that there is an incremental change in the total number of failure records stored in the failure table. Upon receipt of the new failure record, the pattern analysing module (106) may subject the new failure record to a pattern analysis in order to generate a failure report. The pattern analysing module (106) may also be configured to perform the pattern analysis that may include extracting data associated with one or a plurality of metrics from the new failure record. As mentioned above, the new failure record may be associated with an individual user and hence, it may enable the pattern analysing module (106) to identify and extract one or more metrics associated with attributes, such as a number of failed attempts in the incident time range, a number of failed

attempts before the incident time range, a number of successful attempts after the incident time range, etc. As mentioned in the preceding description, it should be noted that these metrics are extracted from the new failure record (or particularly the current failed attempt). The pattern analysis performed by the pattern analysing module (106) may also include obtaining the user behavioural patterns by comparing the extracted data (i.e. data extracted from the new failure record) to historical behavioural data of the same user that may be retrieved from the database (110). In certain embodiments, it may be preferred to configure the pattern analysing module (106) to perform the following operations in sequence: (i) identify a number of failed attempts during the incident time range, (ii) identify a number of failed attempts before the incident time range, (iii) identify a number of successful attempts after the incident time range, and eventually (iv) obtain the user behavioural patterns by comparing the extracted data to historical behavioural data of the same user that may be retrieved from the database (110). It should be appreciated that the data retrieved from operations (i), (ii) and (iii) are extracted from the new failure record (or the current failed attempt).

In certain embodiments, the pattern analysing module (106) may also be configured to generate a failure report after completing the pattern analysis. It should be appreciated that the failure report may comprise the pattern analysis associated with the user.

In some preferred embodiments, the system (100) may also comprise a prioritisation module (108) for subjecting the failure report generated by the pattern analysing module (106) to prioritisation, before sending the failure report to the user, the server administrator or both. Preferably, the prioritisation module (108) may be configured to determine an urgency priority level for the failure report after receiving it from the pattern analysing module (106). Accordingly, the urgency priority level determined by the prioritisation module (108) may be used to determine a sequence or order for sending the failure report to the user, the server administrator or both. Such feature may particularly be beneficial when a plurality of failure reports is generated by the pattern analysing module (106). It may also help in ensuring that the failure reports may be sent to the users according to severity of the failure attempts (or an urgency level that may require the user's or the server administrator's attention).

Preferably, the prioritisation module (108) may be configured to determine the urgency priority level for a failure report based on the metrics selected or configured through a configuration module (102). More preferably, the prioritisation module (108) may be configured to determine the urgency priority level based on a prioritisation table. For example, the prioritisation table as illustrated in Table 1 may be established based on a relationship between number of failed attempts and urgency priority level. In some other exemplary embodiments, the prioritisation table may be established based on a relationship between the number of failed attempts and at least one variable selected from the group comprising identification numbers assigned to target applications or application servers of which the users are accessing (sp_id), number of failed attempts (i.e. the number of times that a user fails to gain access to the target application or application server), trust level (i.e. the level of trust required in order to gain access to the application or application server) and type of authentication method (i.e. authentication implemented to gain access to the target application or application server). However, it should be appreciated that other suitable variables may be used in building the prioritisation table.

Table 1

Number of failed attempts	sp_id	Urgency priority level
> 10	1	1
> 10	2	1
> 10	3	2
5 - 10	1	1
5 - 10	2	2
5 - 10	3	3
< 5	1	1
< 5	2	3
< 5	3	4

Accordingly, the prioritisation module (108) may be configured to transfer the failure report to the user, the server administrator or both, based on the urgency priority level. The prioritisation module (108) may also be configured in some of the embodiments to function in an automated setting, such that the failure report may automatically be

sent to a person of interest after the urgency priority level has been determined.

As mentioned in the foregoing, it may be preferred to enable dynamic configuration to the system (100), so that the system (100) may customise or modify one or a plurality of metrics required in operations performed by the pattern analysing module (106) or prioritisation module (108). To achieve such objective, it may be essential to provide the system (100) with a configuration module (102). As mentioned in the foregoing, a failed access attempt may comprise information associated with a plurality of metrics, and each of the metrics may be related to an attribute of the failed attempt. Examples of the metrics may include but are not limited to uuid, time_login, browser_info, browser_osname, trust_required, number_attempt and sp_id. Hence, by providing the configuration module (102) to the system (100), it may enable the server administrator to identify and select the metrics required in the pattern analysis or prioritisation.

The configuration module (102) may also enable personnel administrating the system (100) to identify and select the metrics to be included in the failure report and sent to the user or server administrator. If different metrics (or different groups of metrics) may be selected and sent to the user and server administrator, the pattern analysing module (106) may be configured to generate different failure reports. For example, the pattern analysing module (106) may generate a failure reasoning report to the user, providing possible reasons to the user regarding the failed attempts. In contrast, an error report may be generated by the pattern analysing module (106) to the server administrator, where issues pertaining to the user failure attempts are escalated to the server administrator.

In certain embodiments, the configuration module (102) may also allow modifications to the system configuration, so that the metrics may be changed in situations where it may be required to fine-tune, correct or reset an incorrect configuration. The objective to introduce a dynamic configuration to the system (100) may subsequently be achieved, which is unlike the prior art that adopts static configuration or uses fixed parameters in monitoring and analysing operations.

In further embodiments, it may be preferred to provide a method for monitoring a user access to an application server. More preferably, the method may be performed using the system described in the foregoing.

Figure 3 illustrates a general flow chart of the method for monitoring a user access to an application server, according to a further embodiment of the invention. The method may generally comprise several major stages, as illustrated in Figure 3. The first stage (202) or "configuration management" enables that metrics be dynamically configured for subsequent analyses and operations to provide a flexible operating environment. The second stage (204) or "failure monitoring" is where a failed access attempt by a user is captured and added to a failure table as a new failure record, whereas the third stage (206) or "pattern analysis" takes place when a pattern analysis is performed on the new failure record in order to generate a failure report. The fourth stage (208) or "automated reporting and alert notification" is where the failure report is sent to the user or server administrator based on an urgency priority level. An alert may also be generated in this stage for notifying the user and server administrator.

Figure 4 is a flow chart illustrating an exemplary first stage (202) in more detail. More preferably, the first stage (202) may be initiated before executing the operations in the second, third and fourth stages (204, 206, 208). In particular, a failed access attempt may comprise information associated with a plurality of metrics, and each metric may be related to an attribute of the failed attempt. For example, the metrics that could be extracted from the failed attempt may include but are not limited to uuid, time_login, browser_info, browser_osname, trust_required, number_attempt and sp_id. Hence, it may be essential that the metrics required for performing the third stage (206) or the fourth stage (208) be identified and selected in the first stage (202), as in Step 2022c. The metrics desired to be sent to the user or server administrator in the failure report may also be identified and selected in some embodiments, as illustrated in Steps 2022a and 2022b. In certain embodiments, it may be allowed in the first stage (202) to change the metrics before generating a failure report in the third stage (206). Such feature may be beneficial in situations where it is required to fine-tune, correct or reset an incorrect configuration. Hence, by introducing the first stage (202), it may enable

that metrics be dynamically configured or changed according to an operating environment, unlike the prior art that uses fixed parameters in monitoring and analysing processes.

Figure 5 is a flow chart illustrating an exemplary second stage (204) in more detail. In particular, failed access attempts by various users to a target application server may be monitored in Step 2042. Instead of monitoring the user failed attempts at an adaptive web browser, the second stage (204) may preferably be performed to monitor a user-specific failure table associated with the application server. In specific, if a user fails to access the application server, such failed attempt may be detected, identified and added to the failure table associated with that user as a new failure record. Hence, when there is an incremental change in a total number of failure records in the failure table, it could be easily detected. In certain embodiments, such simple monitoring may be reduced to a background process and may be performed continuously.

After detecting in Step 2042 that there is an incremental change in the total number of failure records in the failure records, the new failure record may then be subjected to the third stage (206), as in Step 2044 of Figure 5.

Figure 6 is a flow chart illustrating an exemplary third stage (206) in more detail. The new failure record received from the second stage (204) may be subjected to a pattern analysis in Step 2062, so that a failure report may be generated. Preferably, the pattern analysis may be performed to extract data associated with one or a plurality of metrics from the new record. As described in the preceding description, the new failure record may be associated with a particular user and so, the pattern analysis may be performed to identify a number of failed attempts for that particular user within the incident time range. The pattern analysis may alternatively be performed to identify metrics, such as a number of failure attempts for the particular user before the incident time range and a number of successful attempts for the particular user after the incident time range. In addition, the pattern analysis may further be performed to obtain behavioural patterns of the particular user by comparing the extracted data to historical behavioural data of the same user (which may be retrieved from the database (110)). After completing the

pattern analysis in Step 2062, a failure report comprising the pattern analysis may be generated in Step 2064. The failure report may be subjected to the fourth stage (208) in Step 2066.

Figure 7 is a flow chart illustrating an exemplary fourth stage (208) in more detail, in which the failure report may be subjected to prioritisation. In particular, in Step 2082, the failure report obtained from the third stage (206) may be prioritised based on the urgency priority level that may define a sequence for sending the report to the user, the server administrator or both. In some embodiments, the urgency priority level may be determined based on the metrics selected or configured in the first stage (202). In other embodiments, the urgency priority level may be determined by using Table 1 (or a prioritisation table) that is built based on a relationship between the number of failed attempts and the urgency priority level.

After identifying the urgency priority level for a particular failure report in Step 2082, the failure report may be sent in Step 2084 to the user, the server administrator or both based on the urgency level. It should be appreciated that the user may receive a failure report different from that received by the server administrator in certain embodiments. Such difference may arise, as different metrics (or different groups of metrics) may be selected in the first stage (202) to be sent to the user and server administrator. As such, the user may receive a failure reasoning report (providing possible reasons to the user regarding the failed attempts), whereas the server administrator may receive an error report (escalating issues pertaining to the user failure attempts to the administrator).

In another further embodiment of the invention, the method described in the foregoing may be converted to a series of computer-executable program instructions stored on a non-transitory computer-readable storage medium. When the program instructions are executed by a processing module, it may cause the processing module to perform the steps shown in Figures 3, 4, 5, 6 and 7, thus allowing the monitoring of a user access to an application server to be performed in an automated setting.

The disclosure includes as contained in the appended claims, as well as that of the

foregoing description. Although this invention has been described in its preferred form with a degree of particularity, it is understood that the disclosure of the preferred form has been made only by way of example and that numerous changes in the details of construction and the combination and arrangements of parts may be resorted to without departing from the scope of the invention.

Claims

1. A method for monitoring a user access to an application server, comprising the steps of:

detecting a variation in a total number of failure records in a user-specific failure table, wherein each of the failure records is indicative of a historical failed attempt by the user to access the application server,

characterised in that the method further comprises:

subjecting a new failure record to a pattern analysis so as to generate a failure report, if there is an incremental change in the total number of failure records in the failure table, wherein the pattern analysis comprises at least one of extracting data associated with a plurality of metrics from the new failure record, and comparing the extracted data to user historical behavioural data before generating the failure report; and

prioritising the generated failure report based on an urgency priority level that defines a sequence for sending the failure report to the user or server administrator, wherein the urgency priority level is determined using a prioritisation table.

2. The method according to claim 1 further comprising the step of selecting the metrics for the pattern analysis based on attributes of the failure record.

3. The method according to claim 1 further comprising the step of selecting the metrics to be included in the failure report and sent to the user or server administrator.

4. The method according to claim 1, wherein the metrics extracted in the pattern analysis comprise a number of failed attempts in an incident time range, a number of failed attempts before the incident time range and a number of successful attempts after the incident time range.

5. The method according to claim 1, wherein the prioritisation table is built based on a relationship between urgency priority level and at least a variable selected from a group comprising identification numbers assigned to target applications or application

servers of which the users are accessing, number of failed attempts, trust level and type of authentication method.

6. A system (100) for monitoring a user access to an application server, comprising:

a failure monitoring module (104) for detecting a variation in a total number of failure records in a user-specific failure table, wherein each of the failure records is indicative of a historical failed attempt by the user to access the application server,

characterised in that the system (100) further comprises:

a pattern analysing module (106) for receiving a new failure record from the failure monitoring module (104) if there is an incremental change in the total number of failure records in the failure table, and subsequently subjecting the new failure record to a pattern analysis for generating a failure report, wherein the pattern analysing module (106) is configured to perform the pattern analysis by extracting data associated with a plurality of metrics from the new failure record, comparing the extracted data to user historical behavioural data before generating the failure report, or both; and

a prioritisation module (108) for determining an urgency priority level for the generated failure report by referring to a prioritisation table, and then prioritising the generated failure report to be sent to the user based on the determined urgency priority level.

7. The system (100) according to claim 6 further comprising a configuration module (102) for selecting the metrics for the pattern analysis based on attributes of the failure record.

8. The system (100) according to claim 7, wherein the configuration module (102) is further configured to select the metrics to be included in the failure report and sent to the user or server administrator.

9. The system (100) according to claim 6, wherein the pattern analysing module (106) is configured to extract a number of failed attempts in an incident time range, a

number of failed attempts before the incident time range or a number of successful attempts after the incident time range.

10. The system (100) according to claim 6, wherein the prioritisation table is built based on a relationship between urgency priority level and at least a variable selected from a group comprising identification numbers assigned to target applications or application servers of which the users are accessing, number of failed attempts, trust level and type of authentication method.

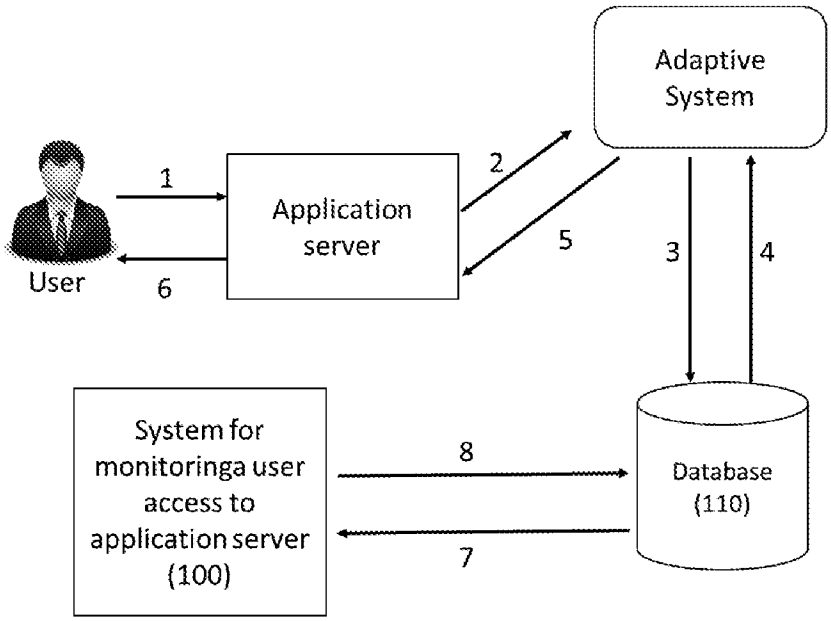

Figure 1

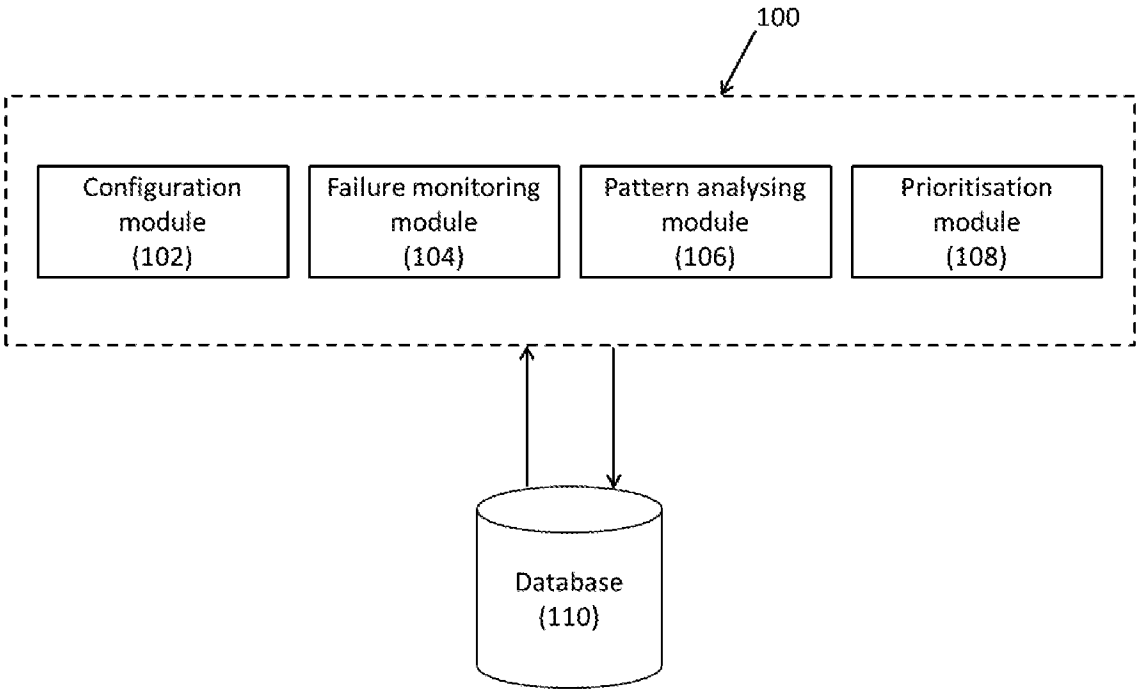

Figure 2

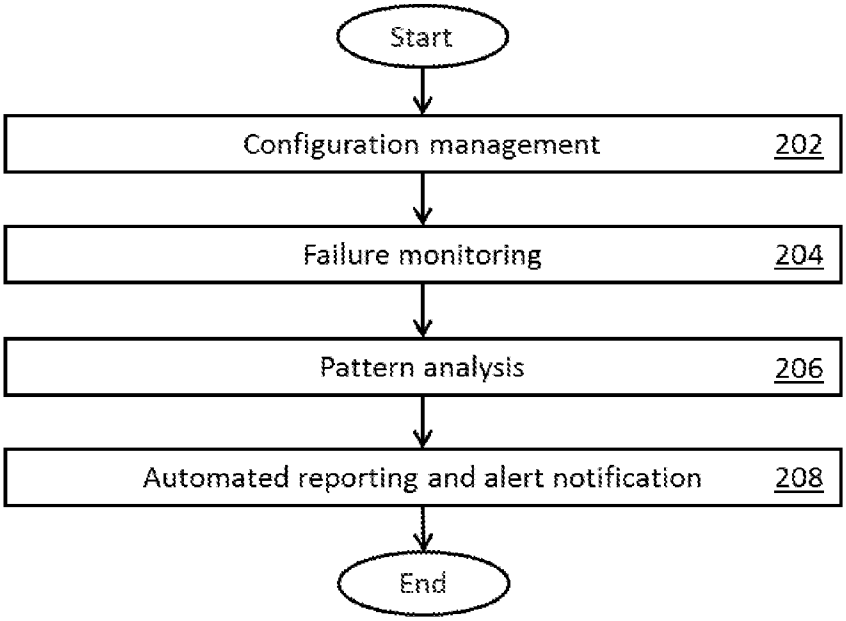
Start
Configuration management
202
Failure monitoring
204
Pattern analysis
206
Automated reporting and alert notification
208
End

Figure 3

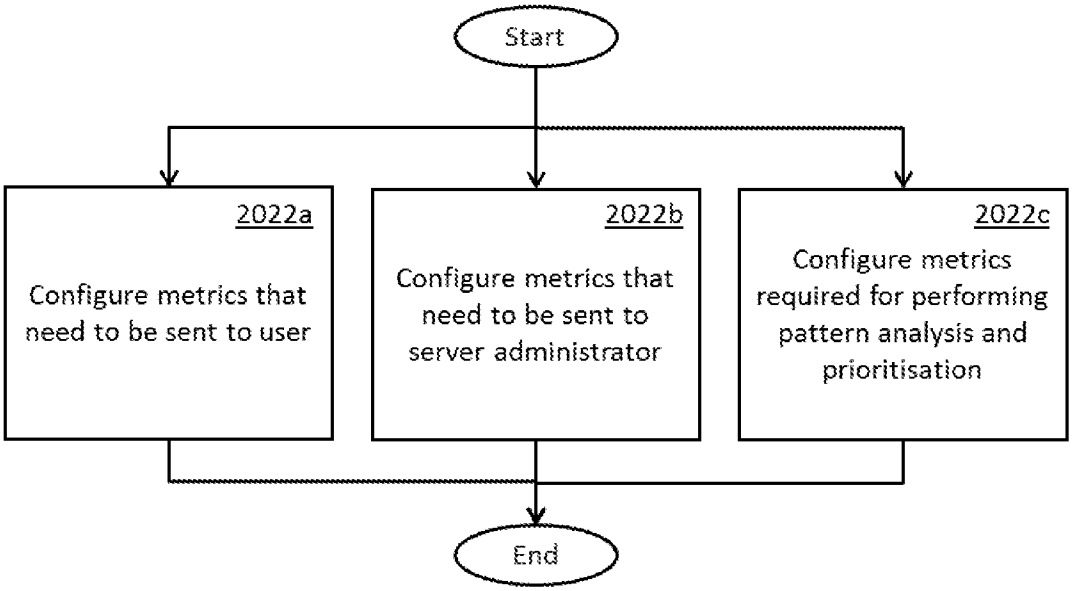

Figure 4

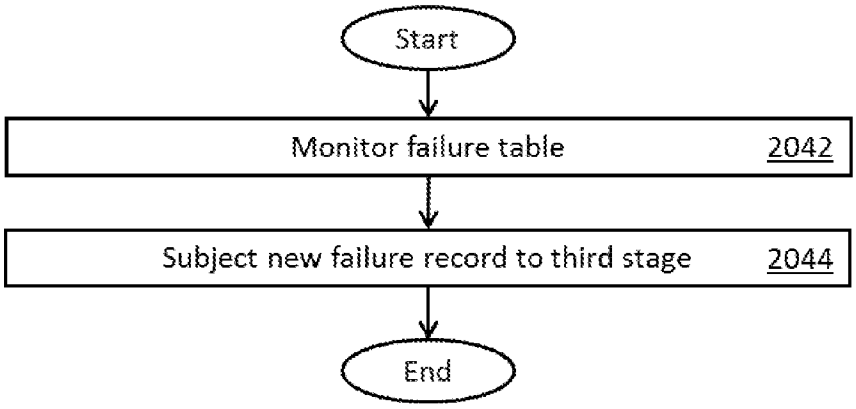

Figure 5

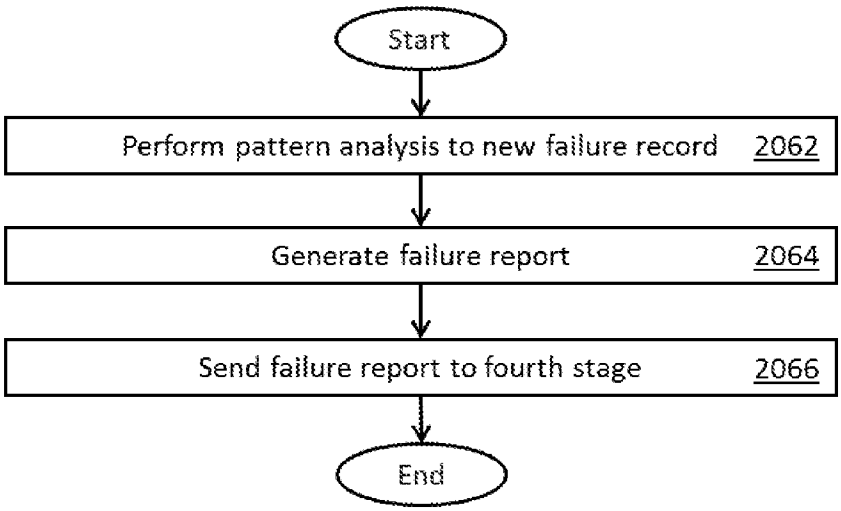

Figure 6

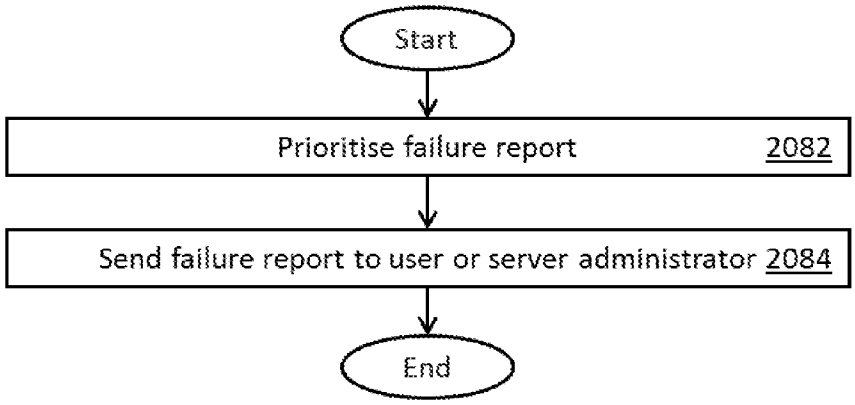

Figure 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/MY2019/050073

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/34(2006.01)i, G06F 11/07(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 11/07; G06F 11/22; G06F 11/30; G06F 17/30; G06F 9/445; H04L 12/26

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: user access, monitor, application server, failure record, failure report, pattern analysis, priority

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2012-0188879 A1 (YANGCHENG HUANG) 26 July 2012 See paragraphs [0006], [0068], [0088]-[0090]; claims 7, 17-18; and figure 6.	1-10
A	US 2018-0276063 A1 (NETSCOUT SYSTEMS, INC.) 27 September 2018 See paragraphs [0040]-[0094]; and figures 2-4.	1-10
A	US 2014-0096127 A1 (TIM G. KIMMET et al.) 03 April 2014 See paragraphs [0049]-[0065]; and figures 4-9.	1-10
A	US 2014-0081925 A1 (TILMANN HAEBERLE et al.) 20 March 2014 See paragraphs [0059]-[0065]; and figures 3-4.	1-10
A	KR 10-2018-0027161 A (KOREA INFORMATION SYSTEM CO., LTD.) 14 March 2018 See paragraphs [0075]-[0097]; and figures 6-7.	1-10

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

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"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
07 February 2020 (07.02.2020)	**07 February 2020 (07.02.2020)**

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
KWON, Sungho
Telephone No. +82-42-481-3547

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/MY2019/050073

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012-0188879 A1	26/07/2012	EP 2460313 A1	06/06/2012
		US 8873402 B2	28/10/2014
		WO 2011-012173 A1	03/02/2011
US 2018-0276063 A1	27/09/2018	EP 3379419 A2	26/09/2018
		EP 3379419 A3	05/12/2018
		US 10289473 B2	14/05/2019
US 2014-0096127 A1	03/04/2014	US 8949824 B2	03/02/2015
US 2014-0081925 A1	20/03/2014	US 8959063 B2	17/02/2015
KR 10-2018-0027161 A	14/03/2018	None	