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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 May 2018 (31.05.2018)

W I P O | PCT

(10) International Publication Number
WO 2018/097877 Al

(51) International Patent Classification:
G06F 11/36 (2006.01)

(21) International Application Number:
PCT/US2017/052388

(22) International Filing Date:
20 September 2017 (20.09.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/361,565 28 November 2016 (28. 11.2016) US

(71) Applicant: GOOGLE LLC [US/US]; 1600 Amphitheatre Parkway, Mountain View, CA 94043 (US).

(72) Inventor: VALLONE, Anthony; 1600 Amphitheatre Parkway, Mountain View, CA 94043 (US).

(74) Agent: KRUEGER, Brett, A.; Honigman Miller Schwartz and Cohn LLP, 350 East Michigan Avenue, Suite 300, Kalamazoo, MI 49007-3800 (US).

(81) Designated States *(unless otherwise indicated, for every kind cf 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 cf 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,

(54) Title: WINDOW DEVIATION ANALYZER

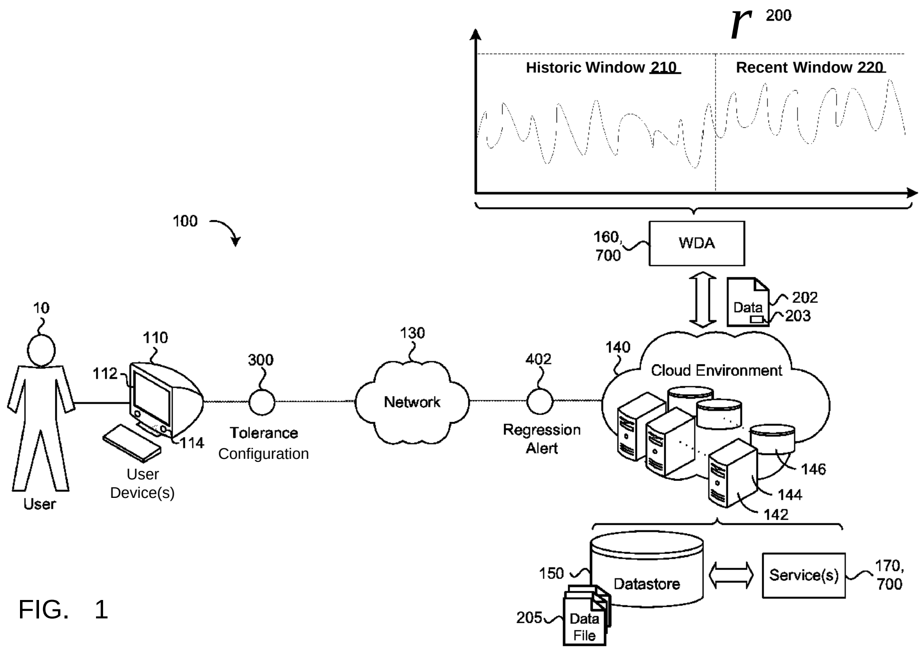


FIG. 1

(57) Abstract: A method (600) includes receiving samples of data (202) over a time range from a start time (204) to an end time (208), determining a mean/median of the data samples within a historic time window (210), and determining a mean/median of the data samples within a recent time window (220). The method also includes determining a data sample delta based on a difference between the mean/median of the data samples within the recent time window and the mean/median of the data samples within the historic time window. The method also includes determining whether the data sample delta exceeds a delta tolerance, and when the data sample delta exceeds the delta tolerance, determining a regression in performance. The method also includes transmitting a regression alert (402) to a user device (110). The user device is configured to notify a user (10) associated with the user device of the regression in performance.

[Continued on nextpage]

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))*

Window Deviation Analyzer

TECHNICAL FIELD

[0001] This disclosure relates to determining performance regressions for a remote system.

BACKGROUND

[0002] A distributed system (e.g., cloud environment) generally includes many loosely coupled computers, each of which typically includes a computing resource (e.g., one or more computer processors) and/or storage resources (e.g., memory, flash memory, and/or disks). A distributed storage system may overlay a storage abstraction (e.g., key/value store or file system) on the storage resources. Moreover, a server process running on one or more of the computers can export its capabilities to client processes running on other computers. The client processes result in a load on the computing resources and/or the storage resources. A distributed system typically warrants extensive monitoring and management to deliver expected performance and reliability.

SUMMARY

[0003] A distributed system may experience a regression in performance when the distributed system is still operational, but available capacity for the distributed system is degraded and/or reduced. Developers that create software applications or services for execution on the distributed system often lack tools for identifying and rectifying performance regressions at any stage during the development cycle. As a result, developers often identify performance regressions long after integrating offending portions of code, thereby leaving the developers with the onerous task of working backward to track performance regressions. This task is difficult and costly, and may be nearly impossible to track a point or cause of a performance regression. The present disclosure provides systems and methods for identifying performance regressions.

[0004] One aspect of the disclosure provides a method for determining a regression in performance. The method includes receiving, at data processing hardware, samples of data over a time range from a start time to an end time and determining, by the data processing hardware, a mean/median of the data samples within a historic time window extending from the start time to an intermediate time between the start time and the end time. The method also includes: determining, by the data processing hardware, a mean/median of the data samples within a recent time window extending from the intermediate time to the end time; determining, by the data processing hardware, a data sample delta based on a difference between the mean/median of the data samples within the recent time window and the mean/median of the data samples within the historic time window; and determining whether the data sample delta exceeds a delta tolerance. When the data sample delta exceeds the delta tolerance, the method includes determining, by the data processing hardware, a regression in performance and transmitting a regression alert from the data processing hardware to a user device in communication with the data processing hardware. The user device is configured to notify a user associated with the user device of the regression in performance in response to receiving the regression alert.

[0005] Implementations of the disclosure may include one or more of the following optional features. In some implementations, the method includes receiving a tolerance configuration at the data processing hardware from the user device. The tolerance configuration may identify a delta tolerance type and include the user-defined tolerance constraints for the identified tolerance type. The method may also include determining, by the data processing hardware, the delta tolerance for the identified tolerance type based upon the user-defined tolerance constraints and the mean/median of the data samples within the historic time window. The identified tolerance type may include a mean-based delta tolerance and the user-defined tolerance constraints may include at least one of a constant term, a mean coefficient, or a standard deviation coefficient. The identified tolerance type may also include a median-based delta tolerance and the user-defined tolerance constraints may include at least one of a constant term, a median coefficient, or a median absolute deviation coefficient. The received tolerance configuration may further indicate a directional bias for the data sample delta including one of an ignore increase bias or an ignore decrease bias.

[0006] In some examples, when the directional bias for the data sample delta includes the ignore increase bias and when the data sample delta exceeds the delta tolerance, the method includes determining, by the data processing hardware, whether the data sample delta is greater than zero or less than zero. When the data sample delta is greater than zero, the method may include determining, by the data processing hardware, no regression in performance exists. When the data sample delta is less than zero, the method may include determining the regression in performance and transmitting the regression alert from the data processing hardware to the user device.

[0007] When the directional bias for the data sample delta includes the ignore decrease bias and when the data sample delta exceeds the delta tolerance, the method may include determining, by the data processing hardware, whether the data sample delta is greater than zero or less than zero. When the data sample delta is greater than zero, the method may include determining the regression in performance and transmitting the regression alert from the data processing hardware to the user device. When the data sample delta is less than zero, the method may include determining, by the data processing hardware, no regression in performance exists. Each received data sample may include a duration of time for a service executing on the data processing hardware to retrieve a data file having a specified capacity from memory hardware in communication with the data processing hardware.

[0008] In some examples, the method includes receiving, at the data processing hardware, an update for a service executing on the data processing hardware from the user device and setting, by the data processing hardware, the intermediate time between the start time and the end time equal to a time of when the update for the service was received from the user device. The user device, in response to receiving the regression alert from the data processing hardware, may be configured to render regression notification for display upon a graphical user interface executing on the user device. The regression notification may notify the user associated with the user device of the regression in performance. The user device, in response to receiving the regression alert from the data processing hardware, may be configured to control one or more speakers of the user device to produce an audio signal associated with the alert signal, the audio signal notifying the user associated with the user device of the regression in performance. The historic time window and the recent time window may include different times, and may be substantially the same size.

[0009] Another aspect of the disclosure provides a system for determining a regression in performance. The system includes data processing hardware of a distributed system and memory hardware in communication with the data processing hardware. The memory hardware stores instructions that when executed on the data processing hardware cause the data processing hardware to perform operations. The operations include receiving samples of data over a time range from a start time to an end time, determining a mean/median of the data samples within a historic time window extending from the start time to an intermediate time between the start time and the end time, and determining a mean/median of the data samples within a recent time window extending from the intermediate time to the end time. The operations also include determining a data sample delta based on a difference between the mean/median of the data samples within the recent time window and the mean/median of the data samples within the historic time window and determining whether the data sample delta exceeds a delta tolerance. When the data sample delta exceeds the delta tolerance, the operations include determining a regression in performance and transmitting a regression alert to a user device in communication with the data processing hardware. The user device is configured to notify a user associated with the user device of the regression in performance in response to receiving the regression alert.

[0010] In some implementations, the operations include receiving a tolerance configuration from the user device and determining the delta tolerance for the identified tolerance type based upon the user-defined tolerance constraints and the mean/median of the data samples within the historic time window. The tolerance configuration may identify a delta tolerance type and include the user-defined tolerance constraints for the identified tolerance type. The identified tolerance type may include a mean-based delta tolerance and the user-defined tolerance constraints may include at least one of a constant term, a mean coefficient, or a standard deviation coefficient. The identified tolerance type may also include a median-based delta tolerance and the user-defined tolerance constraints may include at least one of a constant term, a median coefficient, or a median absolute deviation coefficient. The received tolerance configuration may also indicate a directional bias for the data sample delta comprising one of an ignore increase bias or an ignore decrease bias.

[0011] The operations may also include, when the directional bias for the data sample delta includes the ignore increase bias and when the data sample delta exceeds the delta tolerance, determining whether the data sample delta is greater than zero or less than zero. When the data sample delta is less than zero, the operations may include determining the regression in performance and transmitting the regression alert from the data processing hardware to the user device. When the directional bias for the data sample delta includes the ignore decrease bias and when the data sample delta exceeds the delta tolerance, the operations may include determining, by the data processing hardware, whether the data sample delta is greater than zero or less than zero. When the data sample delta is greater than zero, the operations may include determining the regression in performance and transmitting the regression alert from the data processing hardware to the user device. When the data sample delta is less than zero, the operations may include determining no regression in performance exists. Each received data sample may include a duration of time for a service executing on the data processing hardware to retrieve a data file having a specified capacity from memory hardware in communication with the data processing hardware.

[0012] In some implementations, the operations include receiving an update for a service executing on the data processing hardware from the user device and setting the intermediate time between the start time and the end time equal to a time of when the update for the service was received from the user device. In response to receiving the regression alert from the data processing hardware, the user device may be configured to render regression notification for display upon a graphical user interface executing on the user device. The regression notification may notify the user associated with the user device of the regression in performance. In response to receiving the regression alert from the data processing hardware, the user device may also be configured to control one or more speakers of the user device to produce an audio signal associated with the alert signal. The audio signal may notify the user associated with the user device of the regression in performance. The historic time window and the recent time window may include different sizes. Additionally or alternatively, the historic time window and the recent time window may include substantially the same size.

[0013] The details of one or more implementations of the disclosure are set forth in the accompanying drawings and the description below. Other aspects, features, and advantages will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

[0014] FIG. 1 is a schematic view of an example system for transmitting a regression alert in response to determining a regression in performance.

[0015] FIG. 2 is a schematic view of an example plot of data samples within a within a historic time window extending from a start time to an intermediate time, and a recent time window extending from the intermediate time to an end time.

[0016] FIGS. 3A and 3B show schematic views of a user device transmitting a tolerance configuration to a window deviation analyzer for determining a delta tolerance.

[0017] FIG. 4A shows a schematic view of a window deviation analyzer transmitting a regression alert to a user device when a data sample delta exceeds a delta tolerance.

[0018] FIG. 4B shows a schematic view of a window deviation analyzer determining no regression in performance exists when a data sample delta is associated with a directional bias.

[0019] FIG. 5 shows a schematic view of window deviation analyzer transmitting a regression alert to a user device in response to determining a regression in performance.

[0020] FIG. 6 is a flowchart of an example method for transmitting a regression alert in response to determining a regression in performance.

[0021] FIG. 7 is a schematic view of an example computing device executing a window deviation an analyzer.

[0022] Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

[0023] A window deviation analyzer may analyze data samples within a historic time window against data samples in a recent time window to detect performance regressions on a distributed system in the presence of noise. The window deviation analyzer may rely on tolerance constraints having values defined by a user to determine a tolerance, that when exceeded by a magnitude of change in the recent data samples from the historic data samples, causes the window deviation analyzer to detect a performance regression. The window deviation analyzer may transmit a regression alert to a user (e.g., developer) to inform the user of the detected performance regression. Accordingly, the window deviation analyzer may automatically detect a performance regression and notify the user so that the user can rectify the performance regression before fully integrating portions of code associated with the performance regression to services/applications executing on the distributed system,

[0024] Referring to FIGS. 1 and 2, in some implementations, a system 100 includes a user device 110 associated with a user 10, who may communicate, via a network 130, with a remote system 140. The remote system 140 may be a distributed system (e.g., cloud environment) having scalable/elastic resources 142. The resources 142 include computing resources 144 and/or storage resources 146. The remote system 140 may include a distributed storage system having a scalable/elastic non-transitory datastore 150 for storing data files 205 thereon. In some examples, the remote system 140 executes one or more services 170 that manage access to the datastore 150 and the data files 205 stored thereon. The service(s) 170 may execute on data processing hardware 700 (FIG. 7).

[0025] In some implementations, the data processing hardware 700 executes a window deviation analyzer (WDA) 160 that analyzes samples of data 202 within a historic time window 210 on a plot 200 against samples of the data 202 within a recent time window 220 on the plot 200. The WDA 160 may receive a tolerance configuration 300 from the user device 110 for use by the WDA 160 in determining a delta tolerance. The delta tolerance may include a tolerance type 302 (FIGS. 3A and 3B) corresponding to one of a mean-based delta tolerance **TMean** or a median-based delta tolerance **TMedian,** or some other metric-based delta tolerance.

[0026] In some implementations, the WDA 160 determines a regression in performance on the remote system 140 when a mean or median of the data samples 202 within the recent time window 220 deviate from a corresponding mean or median of the data samples 202 within the historic time window 210 by a magnitude that exceeds the corresponding mean-based delta tolerance **TMean** or median-based delta tolerance **TMedian.** Accordingly, the delta tolerance may define a threshold value that allows for some deviation between the samples of the data 202 within the historic and recent time windows 210, 220, respectively. FIG. 1 shows the WDA 160 transmitting a regression alert 402 to the user device 110 when the WDA 160 determines the regression in performance. The user device 110 is configured to notify the user 10 of the regression in performance in response to receiving the regression alert 402. In some examples, the user device 110 renders a regression notification 450 (FIG. 4A) for display on a graphical user interface (GUI) 112 executing on the user device 110 that notifies the user 10 of the regression in performance. Additionally or alternatively, the user device 110 may control one or more speakers 114 to produce an audio signal 452 (FIG. 4A) associated with the regression alert 402 to notify the user 10 of the regression in performance.

[0027] FIG. 2 shows the plot 200 of data samples 202 received by the WDA 160 over a time range from a start time 204 to an end time 208. The vertical y-axis denotes a specified metric for the data 202 and the horizontal x-axis denotes time increasing from left to right. Accordingly, each data sample 202 includes a corresponding timestamp 203 (FIG. 1) and a corresponding value of the specified metric for the data 202. In some examples, the specified metric is associated with one or more performance metrics measured by the WDA 160 (and/or service 170) while executing a performance test on the remote system 140. For instance, each data sample may be associated with a conglomerate of different performance metrics that, when taken alone, are not comparable with one another. In one example, the specified metric for each data sample 202 corresponds to a duration of time for the service 170 to retrieve a 100 Megabytes (MB) data file 205 from the datastore 150. In another example, the specified metric for each data sample 202 corresponds to a duration of time for the service 170 to execute a process (e.g., algorithm). In some implementations, the end time 208 corresponds to a current time.

[0028] The historic time window 210 extends from the start time 204 to an intermediate time 206 between the start time 204 and the end time 208. The recent time window 220 extends from the intermediate time 206 to the end time 208. Accordingly, each data sample 202 is time ordered and located within one of the historic time window 210 or the recent time window 220 such that each data sample 202 within the historic time window 210 is associated with an earlier time (e.g., smaller timestamp 203) than the data samples 202 within the recent time window 220. In some examples, the historic time window 210 represents a time range that benchmarks a set of data samples 202 deemed normal. In these examples, the recent time window 220 may or may not commence immediately successive to when the historic time window 210 ends. The plot 200 of FIG. 2 shows the historic time window 210 spanning a longer time range than the recent time window 220. However, the recent time window 220 may increase while a corresponding performance test continues to execute on the WDA 160 or service 170. Accordingly, the historic time window 210 and the recent time window 220 may include substantially the same or different sizes.

[0029] Referring to FIGS. 1 and 2, in some implementations, the WDA 160 determines a historic mean of the data samples 202 within the historic time window 210 and determines a recent mean of the data samples 202 within the recent time window 220. Thereafter, the WDA 160 determines a data sample delta based on a difference between the recent mean of the data samples 202 within the recent time window 220 and the historic mean of the data samples 202 within the historic time window 210, and then determines if an absolute value of the data sample delta exceeds the mean-based delta tolerance T_{Mean} based in part by the tolerance configuration 300 received from the user device 110. When an absolute value of the data sample delta exceeds the T_{Mean}, the WDA 160 may determine the regression in performance and transmit the regression alert 402 to the user device 110 to notify the user 10 of the regression in performance. The WDA 160 may calculate a mean-based regression in performance in accordance with the following equation:

$$RIP = \left(abs(R_{Mean} - H_{Mean})\right) > T_{Mean} \qquad (1A)$$

where RIP is a regression in performance, R_{Mean} is a recent mean of the data samples 202 within the recent time window 220, and H_{Mean} is a mean of the data samples 202 within the historic time window 210. The user device 110, or user 10 associated with the user device 110, may take appropriate action to resolve the regression in performance on the remote system 140.

[0030] In additional implementations, the WDA 160 determines a historic median of the data samples 202 within the historic time window 210 and determines a recent median of the data samples 202 within the recent time window 220. In these implementations, the WDA 160 determines the data sample delta based on a difference between the recent median of the data samples 202 within the recent time window 220 and the historic median of the data samples 202 within the historic time window 210, and then determines if the data sample delta exceeds the median-based delta tolerance T_{Median} based in part by the tolerance configuration 300 received from the user device 110. In some scenarios, the WDA 160 determines the regression in performance when an absolute value of the data sample delta exceeds the **TMedian,** and transmits the regression alert 402 to the user device 110 to notify the user 10 of the regression in performance. The WDA 160 may calculate a median-based regression in performance in accordance with the following equation:

$$RIP = \left(abs\left(R_{Median} - H_{Median}\right)\right) > T_{Median} \qquad \text{(IB)}$$

[0031] where **RMedian** is the recent median of the data samples 202 within the recent time window 220 and **HMedian** is the median of the data samples 202 within the historic time window 210. While the WDA 160 may determine the data sample delta based on the mean/median of the data samples 202 within the historic and recent time windows 210, 220, respectively, the WDA 160 may determine other metrics such as, but not limited to, a standard deviation and/or some percentile of the specified metric associated with the data samples 202 for use in determining whether or not deviations exist between the two time windows 210 and 220.

[0032] Referring to FIGS 3A and 3B, in some implementations, the WDA 160 receives a tolerance configuration 300, 300a-b from the user device 110 that identifies the delta tolerance type 302 and one or more tolerance constraints 304 for use in determining the delta tolerance specified by the delta tolerance type 302. The tolerance constraints 304 may be defined by the user 10 associated with the user device 110 to customize how large of a deviation between the data samples 202 within the historic and recent time windows 210, 220, respectively, is required for the WDA 160 to determine the regression in performance. For instance, the tolerance constraints 304 may define a minimum tolerance, an allowable percentage of change from the data samples 202 in the historic time window 210, and/or account for noise within the historic time window 210. The WDA 160 may determine the delta tolerance (e.g., **TMean** or **TMedian)** for the identified tolerance type 302 based upon the one or more tolerance constraints 304 and the mean/median of the data samples 202 within the historic time window 210. As used herein, the term 'historic data samples 202' may be used to refer to the data samples 202 within the historic time window 210 of the plot 200 of FIGS. 1 and 2.

[0033] FIG. 3A shows the WDA 160 receiving the tolerance configuration 300a identifying the delta tolerance type 302 as the mean-based delta tolerance **TMean** and the user-defined tolerance constraints 304 including a constant term **CT**, a mean coefficient **CMean**, and a standard deviation coefficient **CstdDev**. When the WDA 160 receives the tolerance configuration 300a from the user device 110, the WDA 160 determines the historic mean **(HMean)** of the historic data samples 202, determines a historic standard deviation of the historic data samples 202, and determines the mean-based delta tolerance **TMean** in accordance with the following equation:

$$T'_{Mean} = C_T + (C_{Mean} \times H_{Mean}) + (C_{StdDev} \times {}^{\wedge}StdDev) \qquad \textbf{(2A)}$$

where **HstdDev** is a historic standard deviation of the historic data samples 202, i.e., the standard deviation of the data samples 202 within the historic time window 210 of plot 200.

[0034] The constant term **CT** is operative as a fixed term within **EQ**. 2A while the second term associated with the **CMean** and the third term associated with the **CstdDev** may vary with the values of the historic data samples 202. In some scenarios, each of the mean coefficient **CMean** and the standard deviation coefficient **CstdDev** are set to zero while the constant term **CT** is set to a non-zero integer. Here, the constant term **CT** defines a fixed tolerance for the mean-based delta tolerance **TMean**. The use of only the **CT** for determining the mean-based delta tolerance in **EQ**. 2A may be useful when the historic data samples 202 are relatively noise-free. On the other hand, the **CT** is associated with a minimum tolerance for the mean-based delta tolerance **TMean** of **EQ**. 2A when the mean coefficient **CMean** and/or the standard deviation coefficient **CstdDev** is also set to a non-zero integer. The value of **CT** may define the delta tolerance when the historic data samples 202 are associated with lower magnitudes. For example, if a typical metric for a latency operation is 10 milliseconds (ms), a 50-percent (50%) increase in this latency may not warrant a regression in performance. Accordingly, setting the constant term **CT** to 5 ms would prevent the WDA 160 from determining the regression in performance, regardless of what the other user-defined constraints 304 (e.g., **CMean** and/or **CstdDev)** are set to for use by **EQ**. 2A in calculating the mean-based delta tolerance **TMean**. FIG. 3A shows the **CT** set to five (5).

[0035] The mean coefficient **CMean** is operative to set a tolerance based on a percent deviation from the historic mean **HMean** of the data samples 202 within the historic time window 210. For instance, setting the mean coefficient **CMean** to a value equal to 0.05

requires the recent mean **RMean** of the data samples 202 within the recent time window 220 to deviate by more than five-percent (5%) from the historic mean **HMean** of the data samples 202 within the historic time window 210 before the WDA 160 may warrant a regression in performance, i.e., presuming values for **CT** and **CstdDev** are set to zero. In some scenarios, when the historic data samples 202 are relatively noise-free, the constant term **CT** and the standard deviation coefficient **CstdDev** may be set to zero, while the mean coefficient **CMean** may be set to a non-zero integer (e.g., 0.05). The **CMean** may be set to any percentage of acceptable change/deviation from the historic mean **HMean** of the data samples 202 within the historic time window 210. FIG. 3A shows the **CMean** set to a value equal to 0.05.

[0036] In some implementations, the user-defined constraints 304 include the standard deviation coefficient **CstdDev** set to a non-zero integer to add an amount of tolerance associated with noise about the historic mean **HMean** of the data samples 202 within the historic time window 210. Generally, the standard deviation coefficient **CstdDev** is set to a non-zero integer when the constant term **CT** and/or the mean coefficient **CMean** is also set to a non-zero integer. Unless the historic data samples 202 received by the WDA 160 are guaranteed to have noise, using only the standard deviation coefficient **CstdDev** in **EQ**. 2A is not recommended because the **HstdDev** may approach zero, and therefore, become too sensitive to minor changes/deviations in the recent data samples 202 from the historic data samples 202. As twice the standard deviation is a typical metric for use in defining a margin of error in a normal distribution, the standard deviation coefficient **CstdDev** may be set to a recommended value equal to two (2.0). Thus, about 95-percent (95%) of the data samples 202 within the historic time window 210 will fall in a range equal to the **HMean** plus/minus 2.0 multiplied by the **HstdDev**. FIG. 3A shows the **CstdDev** set to 2.0. In other examples, the standard deviation coefficient **CstdDev** may include a value within the range from 1.0 to 2.0.

[0037] FIG. 3B shows the WDA 160 receiving the tolerance configuration 300b identifying the delta tolerance type 302 as the median-based delta tolerance **TMedian** and the user-defined tolerance constraints 304 including at least one of the constant term **CT**, a median coefficient **CMedian**, or a median absolute deviation coefficient **CMAD**. When the WDA 160 receives the tolerance configuration 300b from the user device 110, the WDA

160 determines the historic median (**HMedian**) of the historic data samples 202, determines a median absolute deviation of the historic data samples 202, and determines the mean-based delta tolerance **TMean** in accordance with the following equation:

$$^{\wedge}Median = C_T + (C_M edian \text{ x } H_{MEDIAN}) + (C_{MAD} \ ^{X} \ H_{MAD}) \qquad \textbf{(2B)}$$

where **HMAD** is the median absolute deviation of the historic data samples 202, i.e., the median absolute deviation of the data samples 202 within the historic time window 210 of plot 200.

[0038] As with **EQ.** 2A, the constant term **C T** is operative as a fixed term within **EQ.** 2B while the second term associated with the **CMedian** and the third term associated with the **CMAD** may vary with the values of the historic data samples 202. In some scenarios, the constant term **C T** is operative as a fixed tolerance for the median-based delta tolerance **TMedian** when the user-defined tolerance constraints 304 include the median coefficient **CMedian** and the median absolute deviation coefficient **CMAD** set to zero, while the constant term **C T** is set to a non-zero integer. In other scenarios, the **C T** is associated with a minimum tolerance for the median-based delta tolerance **TMedian** of **EQ.** 2B when the user-defined tolerance constraints 304 include the median coefficient **CMedian** and/or the mean absolute deviation coefficient **CMAD** also set to a non-zero integer. FIG. 3B shows the constant term **C T** set to zero (0).

[0039] The median coefficient **CMedian** is operative to set a tolerance based on a percent deviation from the historic median **HMedian** of the data samples 202 within the historic time window 210. For instance, setting the mean coefficient **CMedian** to a value equal to 0.03 requires the recent median **RMedian** of the data samples 202 within the recent time window 220 to deviate by more than five-percent (3%) from the historic median **HMedian** of the data samples 202 within the historic time window 210 before the WDA 160 may warrant a regression in performance, i.e., presuming values for **C T** and **CMAD** are set to zero. In some scenarios, when the historic data samples 202 are relatively noise-free, the user-defined tolerance constraints 304 include the constant term **C T** and the median absolute deviation coefficient **CMAD** set to zero and the median coefficient **CMedian** set to a non-zero integer (e.g., 0.03). As with the **CMean** of **EQ.** 2A, the user 10 may set, i.e., via the user-defined tolerance constraints 304 of the tolerance configuration 300, the **CMedian** to any percentage of acceptable change from the historic median **HMedian** of the data samples 202 within the historic time window 210. FIG. 3B shows the **CMedian** set to a value equal to 0.03.

[0040] In some implementations, the user-defined constraints 304 include the median absolute deviation coefficient **CMAD** set to a non-zero integer to add an amount of tolerance associated with noise about the historic median **HMedian** of the data samples 202 within the historic time window 210. As with standard deviation, a median absolute deviation similarly provides a measure of variability for a distribution of data 202. However, the median absolute deviation focuses on median values and typically provides a more robust measurement of variability compared to standard deviation. Generally, the median absolute deviation coefficient **CMAD** is set to a non-zero integer when the constant term **C T** and/or the median coefficient **CMedian** is also set to a non-zero integer to serve as a margin of error about the historic median **HMedian.** Unless the historic data samples 202 received by the WDA 160 are guaranteed to have noise, use of the median absolute deviation coefficient **CMAD** in **EQ**. 2B by itself is not recommended because the **HMAD** may approach zero, and therefore, become too sensitive to minor changes/deviations in the recent data samples 202. As half of all data samples 202 within the historic time window 210 typically fall within acceptable noise levels, the median absolute deviation coefficient **CMAD** may be set to a recommended value equal to one (1.0). Thus, about 50-percent (50%) of the data samples 202 within the historic time window 210 will fall in a range equal to the **HMedian** plus/minus 1.0 multiplied by the **HMAD**. FIG. 3B shows the **CMAD** set to 1.0. Larger time windows are less affected by noise. Accordingly, in some examples, the **CMAD** is set to 0.0 when the time range for the recent time window 220 is large enough to not be affected by noise.

[0041] Referring to FIG. 4A, in some implementations, the WDA 160 transmits the regression alert 402 in response to determining the data sample delta exceeds the delta tolerance. For instance, when the received tolerance configuration 300 indicates the mean-based delta tolerance **TMean,** the WDA 160 determines a mean-based data sample delta based on a difference between the recent mean **RMean** of the data samples 202 within the recent time window 220 and the historic mean **HMean** of the data samples 202 within the historic time window 210. Thereafter, the WDA 160 uses **EQ**. 1A to determine whether the data sample delta exceeds the mean-based delta tolerance **TMean.** When the data sample delta (e.g., abs(R_{Mean} —H_{Mean})) exceeds the **TMean**, the WDA 160 determines the regression in performance and subsequently transmits the regression alert 402 to the user device 110. The regression alert 402 indicates the regression in performance determined by the WDA 160. In some examples, the regression alert 402 is an email or text message sent to the user device 110.

[0042] In some examples, in response to receiving the regression alert 402 from the user device 110, the WDA 160 provides a regression notification 450 to the user 10 associated with the user device 110. The regression notification 450 informs the user 10 of the regression in performance on the remote system 140 based on the data samples 202 returned from performance testing. In some implementations, the user device 110 provides the regression notification 450 by displaying the regression notification 450 on the GUI 112 executing on the user device 110. Additionally or alternatively, the user device 110 may control the one or more speakers 114 to produce the audio signal 452 associated with the regression notification 450. That is, the audio signal 452 includes a special regression notification audio signal 452 to inform the user 10 of the regression in performance on the remote system 140 with or without the user 10 having to visually view the regression notification 450 upon the GUI 112. Thus, the regression alert 402 received at the user device 110 may prompt the user device 110 to display the visual regression notification 450 on the GUI 112 and/or prompt the user device 110 to control the speakers 114 to output the regression notification audio signal 452.

[0043] In some scenarios, performance testing resulting in the mean/median of the recent data samples 202 increasing or decreasing from the mean/median of the historic data samples 202 is desirable while the other one of increasing or decreasing may be associated with a regression in performance. For instance, the WDA 160 may determine that no regression in performance exists when a data sample delta (e.g., **RMean - HMean)** is greater than zero and, thus, associated with an increase from the historic data samples 202. Conversely, the WDA 160 may determine the regression in performance when the data sample delta exceeds the delta tolerance, but is less than zero and, thus, associated with a decrease from the historic data samples 202.

[0044] Referring to FIG. 4B, in some implementations, the WDA 160 receives the tolerance configuration 300 from the user device 110 indicating a directional bias 306 for the data sample delta, in addition to the delta tolerance type 302 and the one or more user-defined tolerance constraints 304. The directional bias 306 specifies one of an Ignore_lncrease bias, an Ignore_Decrease bias, or no directional bias at all. The directional bias 306 may be defined by the user 10 to instruct the WDA 160 to determine no regression in performance exists when the mean/median of the recent data samples 202 deviates/changes from the mean/median of the historic data samples 202 in one of a positive (e.g., increasing) or negative (e.g., decreasing) direction.

[0045] The Ignore_Decrease bias instructs the WDA 160 to determine no regression in performance exists for recent data samples 202 decreasing from the historic data samples 202, even if the magnitude of deviation (e.g., data sample delta) exceeds the delta tolerance (e.g., **TMean** or **TMedian).** However, the WDA 160 will still determine the regression in performance (e.g., via **EQ**. 1A or **EQ**. IB) if the recent data samples 202 increase from the historic data samples 202 by a magnitude exceeding the delta tolerance when the directional bias 306 includes the Ignore_Decrease bias.

[0046] On the other hand, the Ignore_lncrease bias instructs the WDA 160 to determine no regression in performance exists for recent data samples 202 increasing from the historic data samples 202, even if the magnitude of deviation (e.g., data sample delta) exceeds the delta tolerance (e.g., **TMean** or **TMedian).** However, the WDA 160 will still determine the regression in performance (e.g., via **EQ**. 1A or **EQ**. IB) if the recent data samples 202 decrease from the historic data samples 202 by a magnitude exceeding the delta tolerance when the directional bias 306 includes the Ignore_lncrease bias. FIG. 4B shows the directional bias 306 specifying the Ignore_increase bias. Accordingly, when the WDA 160 determines that the data sample delta (e.g., **(RMean - HMean)** or **(RMedian - HMedian))** is greater than zero, the WDA 160 determines no regression in performance exists despite the data sample delta exceeding the delta tolerance (e.g., **TMean** or **TMedian).** Accordingly, the WDA 160 will not transmit a regression alert 402 to the user device 110. On the other hand, if the directional bias 306 instead specified "No Bias" or Ignore_Decrease, the same data sample delta including a value greater than zero and exceeding the delta tolerance would cause the WDA 160 to determine the regression in performance and transmit the regression alert 402 to the user device 110.

[0047] FIG. 5 shows a schematic view 500 of the service 170 executing a performance test by measuring a duration of time to retrieve a specified capacity of a data file 205 from the non-transitory datastore 150 and reporting data samples 202 to the WDA 160 each indicating the duration of time and a corresponding timestamp 203 associated with the data sample 202. For instance, the service 170 may provide a performance test request 502 that measures the duration of time to retrieve the data file 205 of the specified capacity from the datastore 150. In some examples, the specified capacity includes a 100 MB data file 205. In other configurations, the performance test is associated with a micro-benchmark of a process executing on the service 170 that measures a duration of time for the service 170 to execute the process or measures some other metric associated with executing the process. The service 170 may continuously execute the performance test.

[0048] The WDA 160 may plot (e.g., via plot 200 of FIGS. 1 and 2) the values for the duration of time and the timestamp 203 associated with each received data sample 202 over a time range from a start time 204 to an end time 208 associated with a current time. The time range may dynamically increase as the end time 208 increases during execution of the performance test. While executing the performance test, FIG. 5 shows the service 170 receiving a service update 504 from the user device 110 configured to update a version and/or software code on the service 170. Concurrently, the WDA 160 sets an intermediate time 206 equal to a timestamp of the service update 504 on the service 170. In some examples, the WDA 160 distributes the data samples 202 received before the service update 504 into a historic time window 210 (FIG. 2) and distributes the data samples 202 received after the service update 504 into a recent time window 220 (FIG. 2). Here, the historic time window 210 extends between the start time 204 and the intermediate time 206 and the recent time window 220 extends between the intermediate time 206 and the end time 208. The WDA 160 may determine a mean/median of the data samples 202 within the historic time window 210 and determine a mean/median of the data samples 202 within the recent time window 220.

[0049] In some implementations, the WDA 160 benchmarks the service update 504 on the service 170 by determining a data sample delta based on a difference between the mean/median of the data samples 202 within the recent time window 220 and the mean/median of the data samples 202 within the historic time window 210. Thereafter, the WDA 160 determines whether the data sample delta exceeds a delta tolerance (e.g., **TMean** or **TMedian)** based upon the mean/median of the historic data samples 202 and user-defined tolerance constraints 304 (FIGS. 3A and 3B). The determination of the delta tolerance is described above with reference to FIGS. 3A and 3B. When the data sample delta is greater than the delta tolerance, the WDA 160 determines a regression in performance and transmits the regression alert 402 to the user device 110. Accordingly, the WDA 160 determines that the service update 504 resulted in a degradation in performance to the service 170 based on a magnitude of change to the mean/median of the recent data samples 202 from the mean/median of the historic data samples 202. For instance, the duration of time for the service 170 to retrieve the data file 205 having the specified capacity from the datastore 150 after the service update 504 may be longer than the duration of time before the service update 504.

[0050] FIG. 6 provides an example arrangement of operations for a method 600 of determining a regression in performance on a system 140. At block 602, the method 600 includes receiving, at a window deviation analyzer (WDA) 160 executing on data processing hardware 700 (FIG. 7), samples of data 202 over a time range from a start time 204 to an end time 208, and at block 604, determining, by the WDA 160, a mean/median of the data samples 202 within a historic time window 210 extending from the start time 204 to an intermediate time 206 between the start time 204 and the end time 208. The method 600 also includes, at block 606, determining, by the WDA 160, a mean/median of the data samples 202 within a recent time window 220 extending between the intermediate time 206 and the end time 208, and at block 608, determining, by the WDA 160, a data sample delta (e.g., **(RMean - HMean)** or **(RMedian - HMedian))** based on a difference between the mean/median of the recent data samples 202 (e.g., within the recent time window 220) and the mean/median of the historic data samples 202 (e.g., within the historic time window 210).

[0051] At block 610, the method 600 includes determining, by the WDA 160, whether the data sample delta exceeds a delta tolerance, e.g., using EQ. 1A or EQ. IB. In some examples, the WDA 160 receives a tolerance configuration 300 (FIGS. 3A and 3B) indicating a tolerance type 302 and one or more tolerance constraints 304 associated with the tolerance type. The tolerance type 302 may include a mean-based delta tolerance **TMean** (FIG. 3A) based on a historic mean **(HMean)** of the data samples 202 within the historic time window 210 and the tolerance constraints 304 including at least one of a constant term **CT**, a mean coefficient **CMean,** or a standard deviation coefficient **CstdDev.** In other scenarios, the tolerance type 302 includes a median-based delta tolerance **TMedian** (FIG. 3B) based on a historic median **(HMedian)** of the data samples 202 within the historic time window 210 and the tolerance constraints 304 including at least one of the constant term **CT**, a median coefficient **CMedian,** or a median absolute deviation coefficient **CMAD.**

[0052] At block 612, the method 600 includes determining the regression in performance on the system 140 when the data sample delta exceeds the delta tolerance, the method 600. At block 614, the method 600 includes transmitting a regression alert 402 (FIG. 4A) from the data processing hardware 700 to a user device 110 associated with a user 10. The user device 110 is in communication with the data processing hardware 700. For instance, the data processing hardware 700 may transmit the regression alert 402 to the user device 110 over a network 130. The user device 110 is configured to notify the user 10 of the regression in performance in response to receiving the regression alert 402. For instance, the user device 110 may render a regression notification 450 (FIG. 4A) for display on a graphical user interface (GUI) 112 executing on the user device 110. Additionally or alternatively, the user device 110 may control one or more speakers 114 of the user device 110 to produce an audio signal 452 associated with the regression alert 402.

[0053] In some implementations, the method 600 includes receiving a tolerance configuration 300 at the data processing hardware 700 from the user device 110. The tolerance configuration 300 may identify a delta tolerance type 302 and include the user-defined tolerance constraints 304 for the identified tolerance type. The method 600 may also include determining, by the data processing hardware 700, the delta tolerance **TMedian** for the identified tolerance type based upon the user-defined tolerance constraints 304 and the mean/median of the data samples 202 within the historic time window 210. The identified tolerance type may include a mean-based delta tolerance **TMean** and the user-defined tolerance constraints 304 may include at least one of a constant term **CT**, a mean coefficient **CMean,** or a standard deviation coefficient **CstdDev.** Moreover, the method 600 may include calculating the mean-based delta tolerance T_{Mean} using EQ. 2A. The identified tolerance type may also include a median-based delta tolerance T_{Median} and the user-defined tolerance constraints 304 may include at least one of a constant term C_T, a median coefficient C_{Median}, or a median absolute deviation coefficient C_{MAD}. In such cases, the method 600 may include calculating the median-based delta tolerance T_{Median} using EQ. 2B. The received tolerance configuration 300 may further indicate a directional bias 306 for the data sample delta including one of an ignore increase bias or an ignore decrease bias.

[0054] In some examples, when the directional bias 306 for the data sample delta includes the ignore increase bias and when the data sample delta exceeds the delta tolerance, the method 600 includes determining, by the data processing hardware 700, whether the data sample delta is greater than zero or less than zero. When the data sample delta is greater than zero, the method 600 may include determining, by the data processing hardware 700, no regression in performance exists. When the data sample delta is less than zero, the method 600 may include determining the regression in performance and transmitting the regression alert 402 from the data processing hardware 700 to the user device 110.

[0055] When the directional bias 306 for the data sample delta includes the ignore decrease bias and when the data sample delta exceeds the delta tolerance, the method 600 may include determining, by the data processing hardware 700, whether the data sample delta is greater than zero or less than zero. When the data sample delta is greater than zero, the method 600 may include determining the regression in performance and transmitting the regression alert 402 from the data processing hardware 700 to the user device 110. When the data sample delta is less than zero, the method 600 may include determining, by the data processing hardware 700, no regression in performance exists. Each received data sample 202 may include a duration of time for a service executing on the data processing hardware 700 to retrieve a data file 205 having a specified capacity from memory hardware in communication with the data processing hardware 700.

[0056] In some examples, the method 600 includes receiving, at the data processing hardware 700, an update for a service executing on the data processing hardware 700 from the user device 110 and setting, by the data processing hardware 700, the intermediate time 206 between the start time 204 and the end time 208 equal to a time of when the update for the service was received from the user device 110. The user device 110, in response to receiving the regression alert 402 from the data processing hardware 700, may be configured to render regression notification 450 for display upon a graphical user interface 112 executing on the user device 110. The regression notification 450 may notify the user 10 associated with the user device 110 of the regression in performance. The user device 110, in response to receiving the regression alert 402 from the data processing hardware 700, may be configured to control one or more speakers 114 of the user device 110 to produce an audio signal 452 associated with the alert signal, the audio signal 452 notifying the user 10 associated with the user device 110 of the regression in performance. The historic time window 210 and the recent time window 220 may include different times, and may be substantially the same size.

[0057] FIG. 7 is a schematic view of an example of the data processing hardware (e.g., computing device) 700 that may be used to implement the systems and methods described in this document. The data processing hardware 700 is intended to represent various forms of digital computers, such as laptops, desktops, workstations, personal digital assistants, servers, blade servers, mainframes, and other appropriate computers. The components shown here, their connections and relationships, and their functions, are meant to be exemplary only, and are not meant to limit implementations of the inventions described and/or claimed in this document.

[0058] The data processing hardware 700 includes a processor 750, memory 720, a storage device 730, a high-speed interface/controller 740 connecting to the memory 720 and high-speed expansion ports 750, and a low speed interface/controller 760 connecting to a low speed bus 770 and storage device 730. Each of the components 710, 720, 730, 740, 750, and 760, are interconnected using various busses, and may be mounted on a common motherboard or in other manners as appropriate. The processor 710 can process instructions for execution within the computing device 700, including instructions stored in the memory 720 or on the storage device 730 to display graphical information for a GUI on an external input/output device, such as a display 780 coupled to a high speed interface 740. In other implementations, multiple processors and/or multiple buses may be used, as appropriate, along with multiple memories and types of memory. Also, multiple data processing hardware devices 700 may be connected, with each device providing portions of the necessary operations (e.g., as a server bank, a group of blade servers, or a multi-processor system).

[0059] The memory 720 (e.g., memory hardware) includes hardware that stores information non-transitorily within the data processing hardware 700. The memory 720 may be a computer-readable medium, a volatile memory unit(s), or non-volatile memory unit(s). The non-transitory memory 720 may be physical devices used to store programs (e.g., sequences of instructions) or data (e.g., program state information) on a temporary or permanent basis for use by the data processing hardware 700. Examples of non-volatile memory include, but are not limited to, flash memory and read-only memory (ROM) / programmable read-only memory (PROM) / erasable programmable read-only memory (EPROM) / electronically erasable programmable read-only memory (EEPROM) (e.g., typically used for firmware, such as boot programs) as well as disks or tapes. Examples of volatile memory include, but are not limited to, random access memory (RAM), dynamic random access memory (DRAM), static random access memory (SRAM), phase change memory (PCM).

[0060] The storage device 730 is capable of providing mass storage for the data processing hardware 700. In some implementations, the storage device 730 is a computer-readable medium. In various different implementations, the storage device 730 may be a floppy disk device, a hard disk device, an optical disk device, or a tape device, a flash memory or other similar solid state memory device, or an array of devices, including devices in a storage area network or other configurations. In additional implementations, a computer program product is tangibly embodied in an information carrier. The computer program product contains instructions that, when executed, perform one or more methods, such as those described above. The information carrier is a computer- or machine-readable medium, such as the memory 720, the storage device 730, or memory on processor 710.

[0061] The high speed controller 740 manages bandwidth-intensive operations for the computing device 700, while the low speed controller 760 manages lower bandwidth-intensive operations. Such allocation of duties is exemplary only. In some implementations, the high-speed controller 740 is coupled to the memory 720, the display

780 (e.g., through a graphics processor or accelerator), and to the high-speed expansion ports 750, which may accept various expansion cards (not shown). In some implementations, the low-speed controller 760 is coupled to the storage device 730 and low-speed expansion port 770. The low-speed expansion port 770, which may include various communication ports (e.g., USB, Bluetooth, Ethernet, wireless Ethernet), may be coupled to one or more input/output devices, such as a keyboard, a pointing device, a scanner, or a networking device, such as a switch or router, e.g., through a network adapter.

[0062] The data processing hardware 700 may be implemented in a number of different forms, as shown in the figure. For example, it may be implemented as a standard server or multiple times in a group of such servers, as a laptop computer, or as part of a rack server system.

[0063] In some implementations, the data processing hardware 700 implementing the window deviation analyzer (WDA) 160 and the service 170 is in communication with memory hardware in the memory 720 for implementing datastore 150 storing data files 205. The processor 710 executes the WDA 160 and the service 170. For example, the service 170 may execute a performance test on the datastore 150 and the WDA 160 may receive samples of data 202 each including a specified metric (e.g., a duration of time) associated with the performance test and a timestamp 203 associated with the corresponding data sample 202. The WDA 160 may distribute the data samples 202 into a historic time window 210 and a recent time window 220 based on the timestamps 203 and determine if the data samples in the recent time window 220 deviate from the data samples 202 in the historic time window 210. If the magnitude of deviation is large (e.g., greater than delta tolerance), then the WDA 160 may determine a regression in performance and transmit a regression alert 402 to a user device 110 in communication with the WDA 160.

[0064] A software application (i.e., a software resource 110s) may refer to computer software that causes a computing device to perform a task. In some examples, a software application may be referred to as an "application," an "app," or a "program." Example applications include, but are not limited to, mobile applications, system diagnostic applications, system management applications, system maintenance applications, word processing applications, spreadsheet applications, messaging applications, media streaming applications, social networking applications, and gaming applications.

[0065] The memory hardware 1lOhm may be physical devices used to store programs (e.g., sequences of instructions) or data (e.g., program state information) on a temporary or permanent basis for use by a computing device 1lOhc. The non-transitory memory 1lOhm may be volatile and/or non-volatile addressable semiconductor memory. Examples of non-volatile memory include, but are not limited to, flash memory and read¬ only memory (ROM) / programmable read-only memory (PROM) / erasable programmable read-only memory (EPROM) / electronically erasable programmable read-only memory (EEPROM) (e.g., typically used for firmware, such as boot programs). Examples of volatile memory include, but are not limited to, random access memory (RAM), dynamic random access memory (DRAM), static random access memory (SRAM), phase change memory (PCM) as well as disks or tapes.

[0066] Various implementations of the systems and techniques described herein can be realized in digital electronic and/or optical circuitry, integrated circuitry, specially designed ASICs (application specific integrated circuits), computer hardware, firmware, software, and/or combinations thereof. These various implementations can include implementation in one or more computer programs that are executable and/or interpretable on a programmable system including at least one programmable processor, which may be special or general purpose, coupled to receive data and instructions from, and to transmit data and instructions to, a storage system, at least one input device, and at least one output device.

[0067] These computer programs (also known as programs, software, software applications or code) include machine instructions for a programmable processor, and can be implemented in a high-level procedural and/or object-oriented programming language, and/or in assembly/machine language. As used herein, the terms "machine-readable medium" and "computer-readable medium" refer to any computer program product, non-transitory computer readable medium, apparatus and/or device (e.g., magnetic discs, optical disks, memory, Programmable Logic Devices (PLDs)) used to provide machine instructions and/or data to a programmable processor, including a machine-readable medium that receives machine instructions as a machine-readable signal. The term "machine-readable signal" refers to any signal used to provide machine instructions and/or data to a programmable processor.

[0068] The processes and logic flows described in this specification can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by special purpose logic circuitry, e.g., an FPGA (field programmable gate array) or an ASIC (application specific integrated circuit). Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data. Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., EPROM, EEPROM, and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and CD ROM and DVD-ROM disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[0069] To provide for interaction with a user, one or more aspects of the disclosure can be implemented on a computer having a display device, e.g., a CRT (cathode ray tube), LCD (liquid crystal display) monitor, or touch screen for displaying information to the user and optionally a keyboard and a pointing device, e.g., a mouse or a trackball, by which the user can provide input to the computer. Other kinds of devices can be used to provide interaction with a user as well; for example, feedback provided to the user can be any form of sensory feedback, e.g., visual feedback, auditory feedback, or tactile feedback; and input from the user can be received in any form, including acoustic, speech, or tactile input. In addition, a computer can interact with a user by sending documents to and receiving documents from a device that is used by the user; for example, by sending web pages to a web browser on a user's client device in response to requests received from the web browser.

A number of implementations have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the disclosure. Accordingly, other implementations are within the scope of the following claims.

WHAT IS CLAIMED IS:

1. A method (600) comprising:

receiving, at data processing hardware (700), samples of data over a time range from a start time (204) to an end time (208);

determining, by the data processing hardware (700), a mean/median of the data samples (202) within a historic time window (210) extending from the start time (204) to an intermediate time (206) between the start time (204) and the end time (208);

determining, by the data processing hardware (700), a mean/median of the data samples (202) within a recent time window (220) extending from the intermediate time (206) to the end time (208);

determining, by the data processing hardware (700), a data sample delta based on a difference between the mean/median of the data samples (202) within the recent time window (220) and the mean/median of the data samples (202) within the historic time window (210);

determining whether the data sample delta exceeds a delta tolerance; and

when the data sample delta exceeds the delta tolerance:

determining, by the data processing hardware (700), a regression in performance; and

transmitting a regression alert (402) from the data processing hardware (700) to a user device (110) in communication with the data processing hardware (700), the user device (110) configured to notify a user (10) associated with the user device (110) of the regression in performance in response to receiving the regression alert (402).

2. The method (600) of claim 1, further comprising:

receiving a tolerance configuration (300) at the data processing hardware (700) from the user device (110), the tolerance configuration (300) identifying a delta tolerance type (302) and including the user-defined tolerance constraints (304) for the identified tolerance type (302); and

determining, by the data processing hardware (700), the delta tolerance for the identified tolerance type (302) based upon the user-defined tolerance constraints (304) and the mean/median of the data samples (202) within the historic time window (210).

3. The method (600) of claim 2, wherein the identified tolerance type (302) comprises a mean-based delta tolerance and the user-defined tolerance constraints (304) comprise at least one of a constant term, a mean coefficient, or a standard deviation coefficient.

4. The method (600) of claim 3, wherein the mean-based delta tolerance is calculated as follows:

$$^{\wedge}\boldsymbol{Mean} = C_T + (C_{Mean} \ \text{x} \ H_{Mean}) + (C_{stdDev} \ \text{x} \ H_{stdDev}),$$

wherein **TMean** is the mean-based delta tolerance, **C T** is the constant term, **CMean** is the mean coefficient, **HMean** is a mean of the historic data samples (202), **CstdDev** is a standard deviation coefficient, and **HstdDev** is a standard deviation of the historic data samples (202).

5. The method (600) of claim 2, wherein the identified tolerance type (302) comprises a median-based delta tolerance and the user-defined tolerance constraints (304) comprise at least one of a constant term, a median coefficient, or a median absolute deviation coefficient.

6. The method (600) of claim 5, wherein the median-based delta tolerance is calculated as follows:

$$\boldsymbol{TMe_{a}n} = C_T + (C_{Median} \ \text{x} \ H_{Median}) + (C_{MAD} \ \text{x} \ H_{MAD}),$$

wherein **TMedian** is the median-based delta tolerance, **C T** is the constant term, **CMedian** is the median coefficient, **HMedian** is a median of the historic data samples (202), **CMAD** is the median absolute deviation coefficient, and **HMAD** is a median absolute deviation of the historic data samples (202).

7. The method (600) of claim 2, wherein the received tolerance configuration (300) further indicates a directional bias (306) for the data sample delta comprising one of an ignore increase bias or an ignore decrease bias.

8. The method (600) of claim 7, further comprising, when the directional bias (306) for the data sample delta comprises the ignore increase bias and when the data sample delta exceeds the delta tolerance:

determining, by the data processing hardware (700), whether the data sample delta is greater than zero or less than zero;

when the data sample delta is greater than zero, determining, by the data processing hardware (700), no regression in performance exists; and

when the data sample delta is less than zero:

determining the regression in performance; and

transmitting the regression alert (402) from the data processing hardware (700) to the user device (110).

9. The method (600) of claim 7, further comprising, when the directional bias (306) for the data sample delta comprises the ignore decrease bias and when the data sample delta exceeds the delta tolerance:

determining, by the data processing hardware (700), whether the data sample delta is greater than zero or less than zero;

when the data sample delta is greater than zero:

determining the regression in performance; and

transmitting the regression alert (402) from the data processing hardware (700) to the user device (110); and

when the data sample delta is less than zero, determining, by the data processing hardware (700), no regression in performance exists.

10. The method (600) of any of claims 1-9, wherein each received data sample (202) comprises a duration of time for a service (170) executing on the data processing hardware (700) to retrieve a data file (205) having a specified capacity from memory hardware (720) in communication with the data processing hardware (700).

11. The method (600) of any of claims 1-10, further comprising:

receiving, at the data processing hardware (700), an update for a service (170) executing on the data processing hardware (700) from the user device (110); and

setting, by the data processing hardware (700), the intermediate time (206) between the start time (204) and the end time (208) equal to a time of when the update for the service (170) was received from the user device (110).

12. The method (600) of any of claims 1-11, wherein the user device (110), in response to receiving the regression alert (402) from the data processing hardware (700), is configured to render a regression notification (450) for display upon a graphical user interface (112) executing on the user device (110), the regression notification (450) notifying the user (10) associated with the user device (110) of the regression in performance.

13. The method (600) of any of claims 1-12, wherein the user device (110), in response to receiving the regression alert (402) from the data processing hardware (700), is configured to control one or more speakers (114) of the user device (110) to produce an audio signal (452) associated with the regression alert (402), the audio signal (452) notifying the user (10) associated with the user device (110) of the regression in performance.

14. The method (600) of any of claims 1-13, wherein the historic time window (210) and the recent time window (220) comprise different sizes.

15. The method (600) of any of claims 1-14, wherein the historic time window (210) and the recent time window (220) comprise substantially the same size.

16. A system (100) comprising:

data processing hardware (700) of a distributed system (140);

memory hardware (720) in communication with the data processing hardware (700), the memory hardware (720) storing instructions that when executed on the data processing hardware (700) cause the data processing hardware (700) to perform operations comprising:

receiving samples of data over a time range from a start time (204) to an end time (208);

determining a mean/median of the data samples (202) within a historic time window (210) extending from the start time (204) to an intermediate time (206) between the start time (204) and the end time (208);

determining a mean/median of the data samples (202) within a recent time window (220) extending from the intermediate time (206) to the end time (208);

determining a data sample delta based on a difference between the mean/median of the data samples (202) within the recent time window (220) and the mean/median of the data samples (202) within the historic time window (210);

determining whether the data sample delta exceeds a delta tolerance; and

when the data sample delta exceeds the delta tolerance:

determining a regression in performance; and

transmitting a regression alert (402) to a user device (110) in communication with the data processing hardware (700), the user device (110) configured to notify a user (10) associated with the user device (110) of the regression in performance in response to receiving the regression alert (402).

17. The system (100) of claim 16, wherein the operations further comprise:

receiving a tolerance configuration (300) from the user device (110), the tolerance configuration (300) identifying a delta tolerance type (302) and including the user-defined tolerance constraints (304) for the identified tolerance type (302); and

determining the delta tolerance for the identified tolerance type (302) based upon the user-defined tolerance constraints (304) and the mean/median of the data samples (202) within the historic time window (210).

18. The system (100) of claim 17, wherein the identified tolerance type (302) comprises a mean-based delta tolerance and the user-defined tolerance constraints (304)

comprise at least one of a constant term, a mean coefficient, or a standard deviation coefficient.

19. The system (100) of claim 18, wherein the mean-based delta tolerance is calculated as follows:

$$\wedge \mathit{Mean} = C_T + (C_{Mean} \text{ x } H_{Mean}) + (\mathit{CstdDev} \text{ x } \mathit{HstdDev}) \ ,$$

wherein **TMean** is the mean-based delta tolerance, **C T** is the constant term, **CMean** is the mean coefficient, **HMean** is a mean of the historic data samples (202), **CstdDev** is a standard deviation coefficient, and **HstdDev** is a standard deviation of the historic data samples (202).

20. The system (100) of claim 17, wherein the identified tolerance type (302) comprises a median-based delta tolerance and the user-defined tolerance constraints (304) comprise at least one of a constant term, a median coefficient, or a median absolute deviation coefficient.

21. The system (100) of claim 20, wherein the median-based delta tolerance is calculated as follows:

$$\wedge \mathit{Mean} = C_T + (C_{Median} \text{ x } H_{MEDIAN}) + (C_{MAD} \times H_{MAD}),$$

wherein **TMedian** is the median-based delta tolerance, **C T** is the constant term, **CMedian** is the median coefficient, **HMedian** is a median of the historic data samples (202), **CMAD** is the median absolute deviation coefficient, and **HMAD** is a median absolute deviation of the historic data samples (202).

22. The system (100) of claim 17, wherein the received tolerance configuration (300) further indicates a directional bias (306) for the data sample delta comprising one of an ignore increase bias or an ignore decrease bias.

23. The system (100) of claim 22, wherein the operations further comprise, when the directional bias (306) for the data sample delta comprises the ignore increase bias and when the data sample delta exceeds the delta tolerance:

determining whether the data sample delta is greater than zero or less than zero;

when the data sample delta is greater than zero, determining no regression in performance exists; and

when the data sample delta is less than zero:

determining the regression in performance; and

transmitting the regression alert (402) from the data processing hardware (700) to the user device (110).

24. The system (100) of claim 22, further comprising, when the directional bias (306) for the data sample delta comprises the ignore decrease bias and when the data sample delta exceeds the delta tolerance:

determining whether the data sample delta is greater than zero or less than zero;

when the data sample delta is greater than zero:

determining the regression in performance; and

transmitting the regression alert (402) from the data processing hardware (700) to the user device (110); and

when the data sample delta is less than zero, determining no regression in performance exists.

25. The system (100) of any of claims 16-24, wherein each received data sample (202) comprises a duration of time for a service (170) executing on the data processing hardware (700) to retrieve a data file (205) having a specified capacity from memory hardware (720) in communication with the data processing hardware (700).

26. The system (100) of any of claims 16-25, wherein the operations further comprise:

receiving an update for a service (170) executing on the data processing hardware (700) from the user device (110); and

setting the intermediate time (206) between the start time (204) and the end time (208) equal to a time of when the update for the service (170) was received from the user device (110).

27. The system (100) of any of claims 16-26, wherein the user device (110), in response to receiving the regression alert (402) from the data processing hardware (700), is configured to render regression notification (450) for display upon a graphical user interface (112) executing on the user device (110), the regression notification (450) notifying the user (10) associated with the user device (110) of the regression in performance.

28. The system (100) of any of claims 16-27, wherein the user device (110), in response to receiving the regression alert (402) from the data processing hardware (700), is configured to control one or more speakers (114) of the user device (110) to produce an audio signal (452) associated with the regression alert (402), the audio signal (452) notifying the user (10) associated with the user device (110) of the regression in performance.

29. The system (100) of any of claims 16-28, wherein the historic time window (210) and the recent time window (220) comprise different sizes.

30. The system (100) of any of claims 16-29, wherein the historic time window (210) and the recent time window (220) comprise substantially the same size.

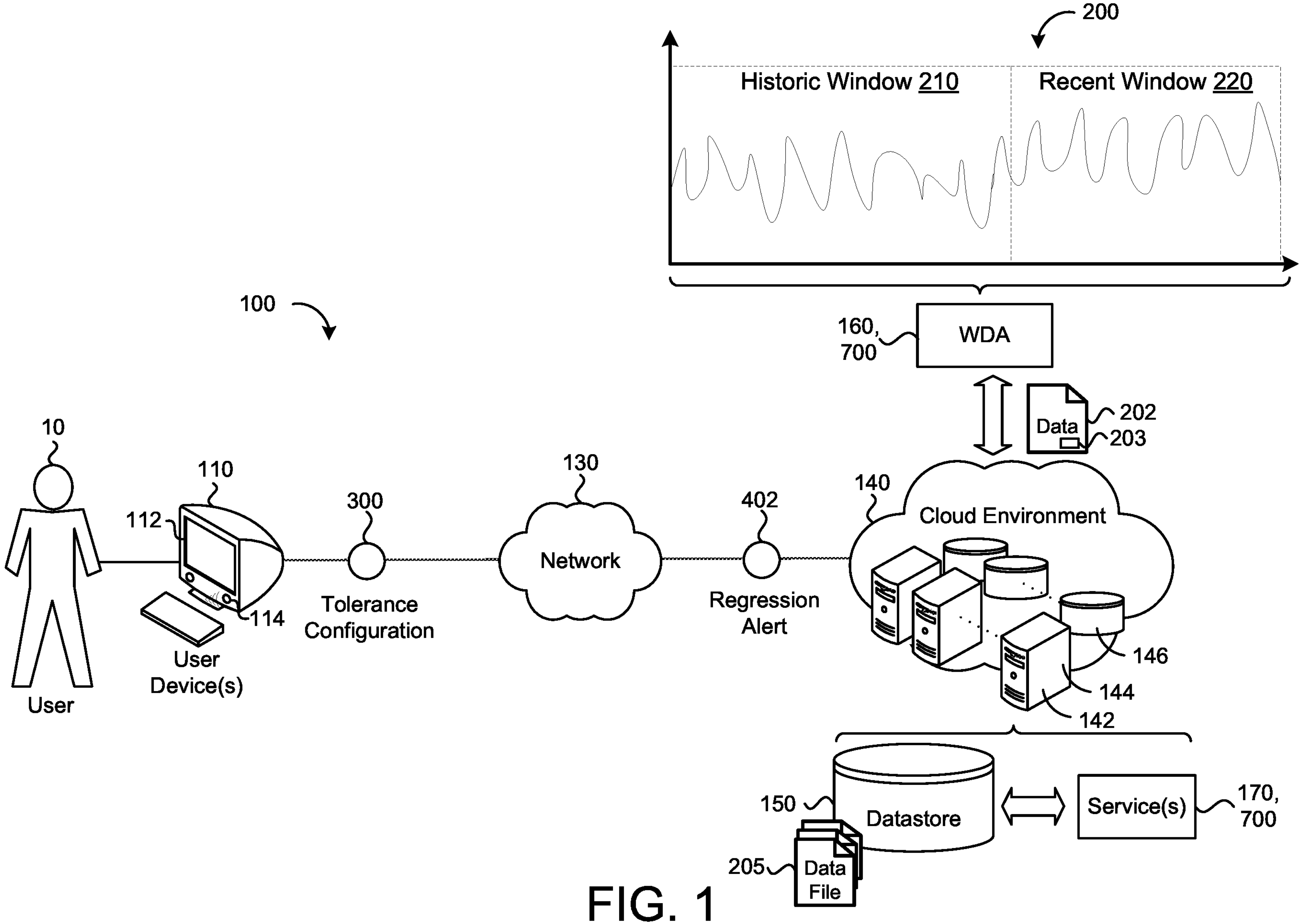

200
Historic Window 210
Recent Window 220
100
160,
700
WDA
Data
202
203
10
110
112
114
User
User
Device(s)
300
Tolerance
Configuration
130
Network
402
Regression
Alert
140
Cloud Environment
146
144
142
150
Datastore
Service(s)
170,
700
205
Data
File

FIG. 1

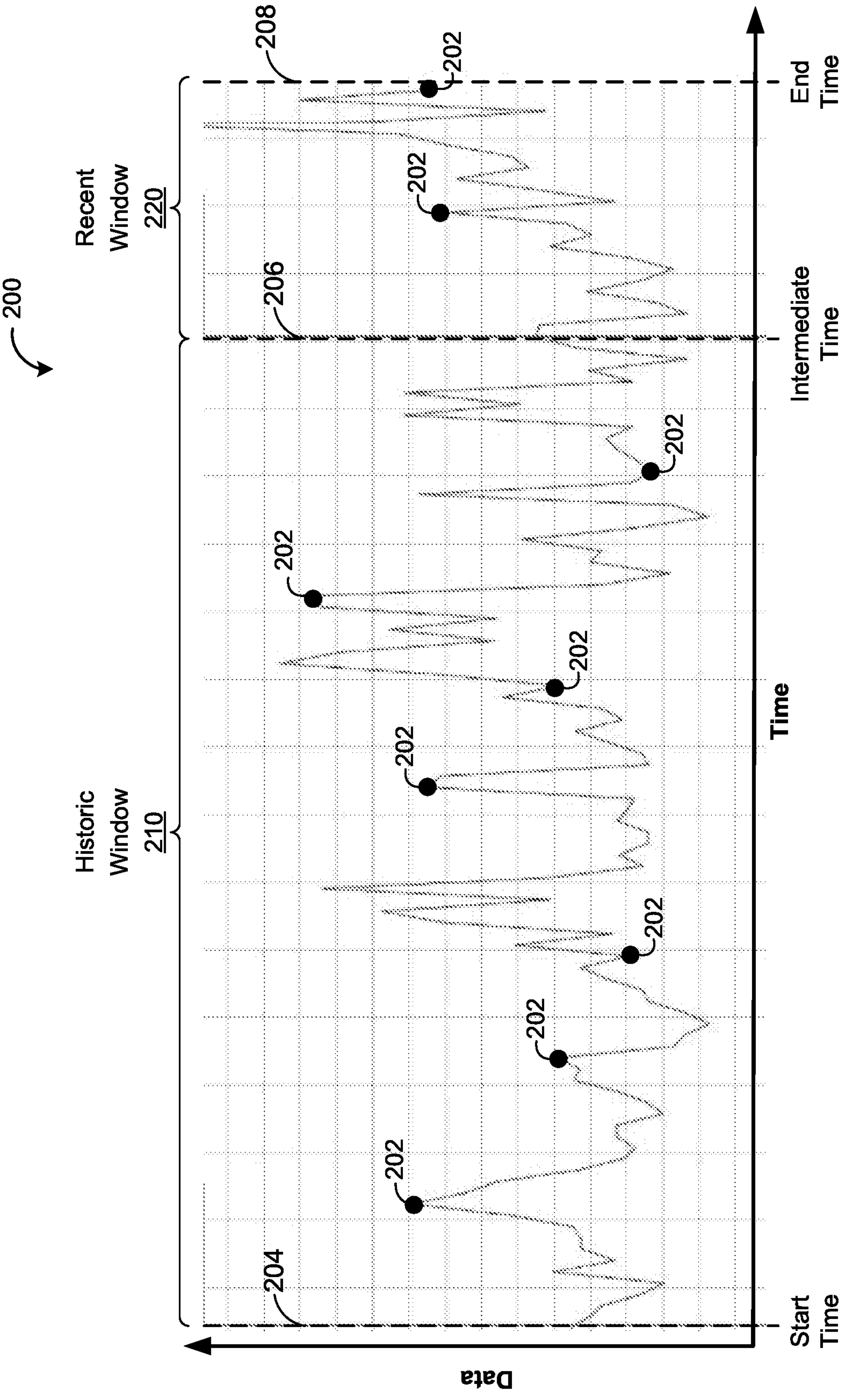

200
Recent Window 220
Historic Window 210
208
206
204
202
202
202
202
202
202
202
202
202
Data
Time
End Time
Intermediate Time
Start Time

FIG. 2

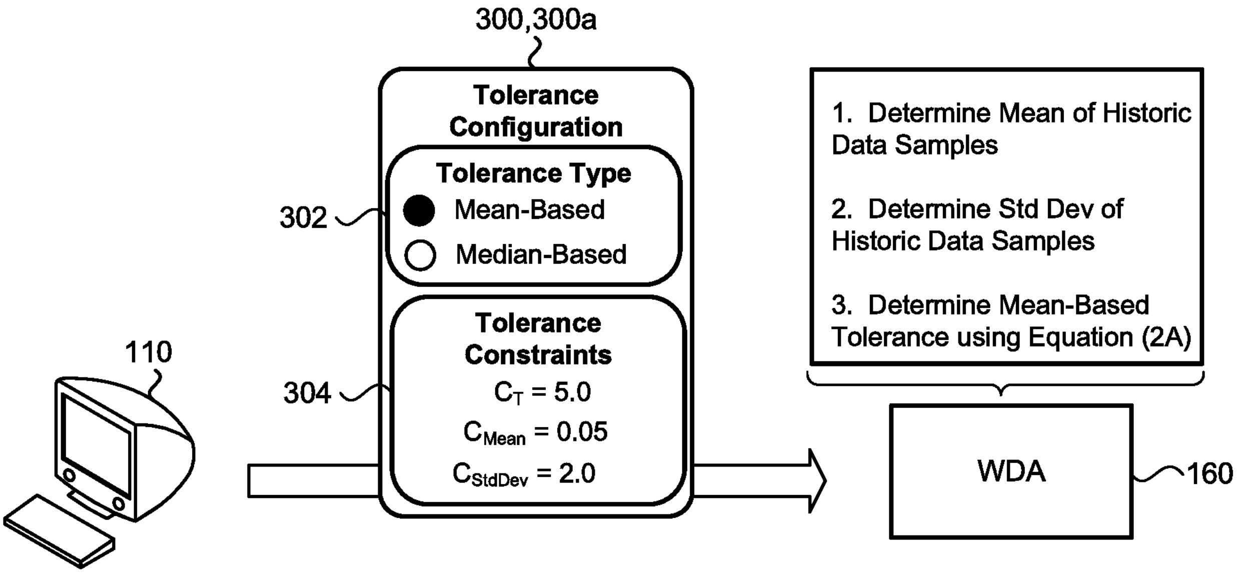

300,300a
Tolerance Configuration
Tolerance Type
● Mean-Based
○ Median-Based
302
Tolerance Constraints
C_T = 5.0
C_{Mean} = 0.05
C_{StdDev} = 2.0
304
110
1. Determine Mean of Historic Data Samples
2. Determine Std Dev of Historic Data Samples
3. Determine Mean-Based Tolerance using Equation (2A)
WDA
160

FIG. 3A

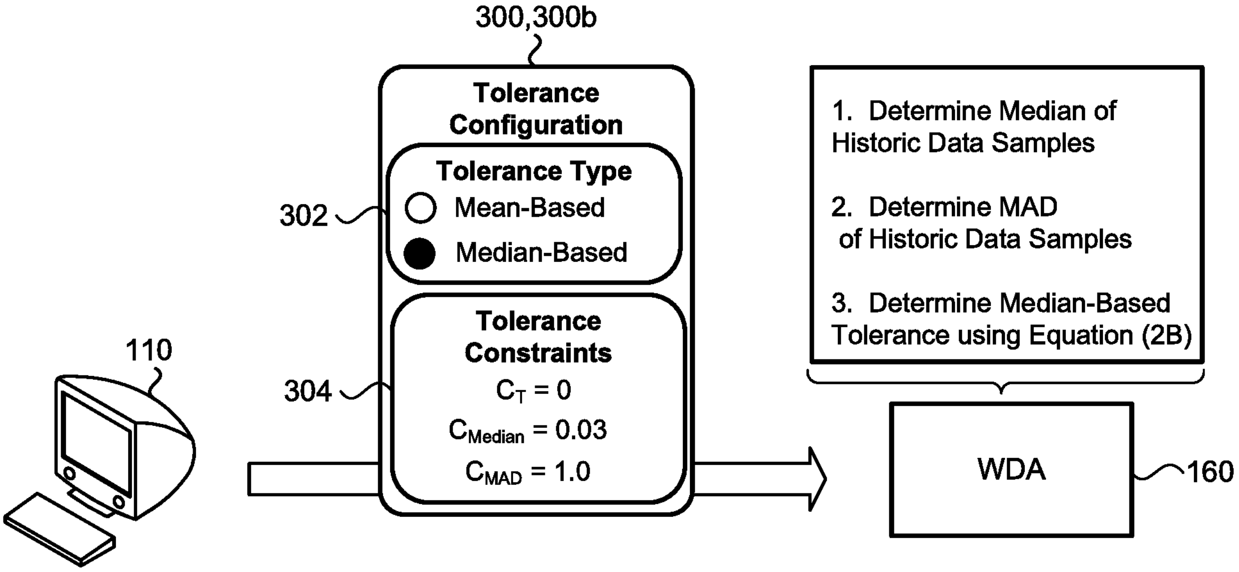
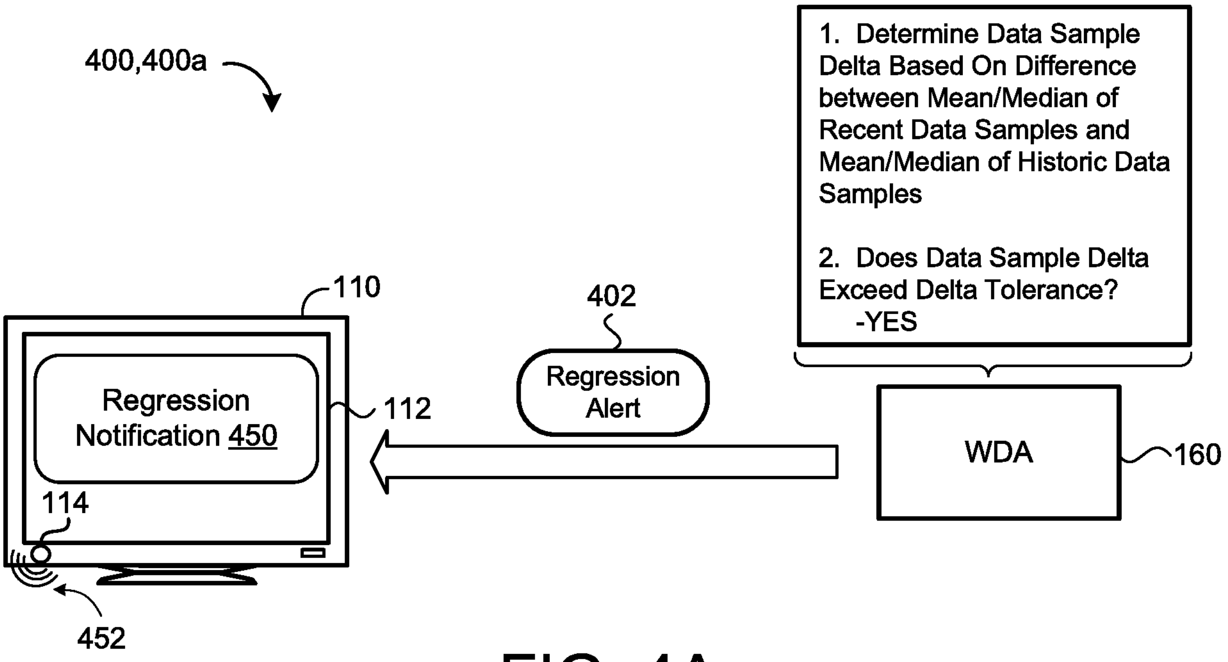

300,300b
Tolerance Configuration
Tolerance Type
○ Mean-Based
● Median-Based
302
Tolerance Constraints
C_T = 0
C_{Median} = 0.03
C_{MAD} = 1.0
304
110
1. Determine Median of Historic Data Samples
2. Determine MAD of Historic Data Samples
3. Determine Median-Based Tolerance using Equation (2B)
WDA
160

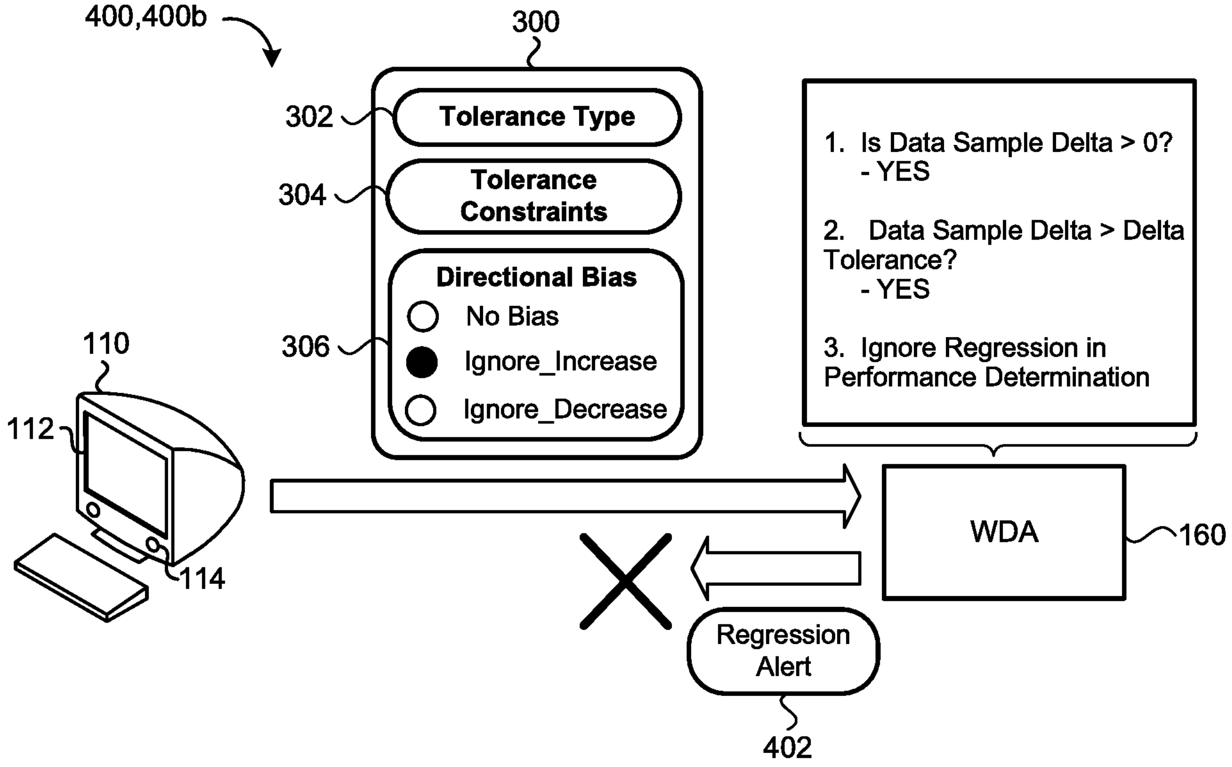
FIG. 3B



1. Determine Data Sample Delta Based On Difference between Mean/Median of Recent Data Samples and Mean/Median of Historic Data Samples

2. Does Data Sample Delta Exceed Delta Tolerance?
   -YES

FIG. 4A



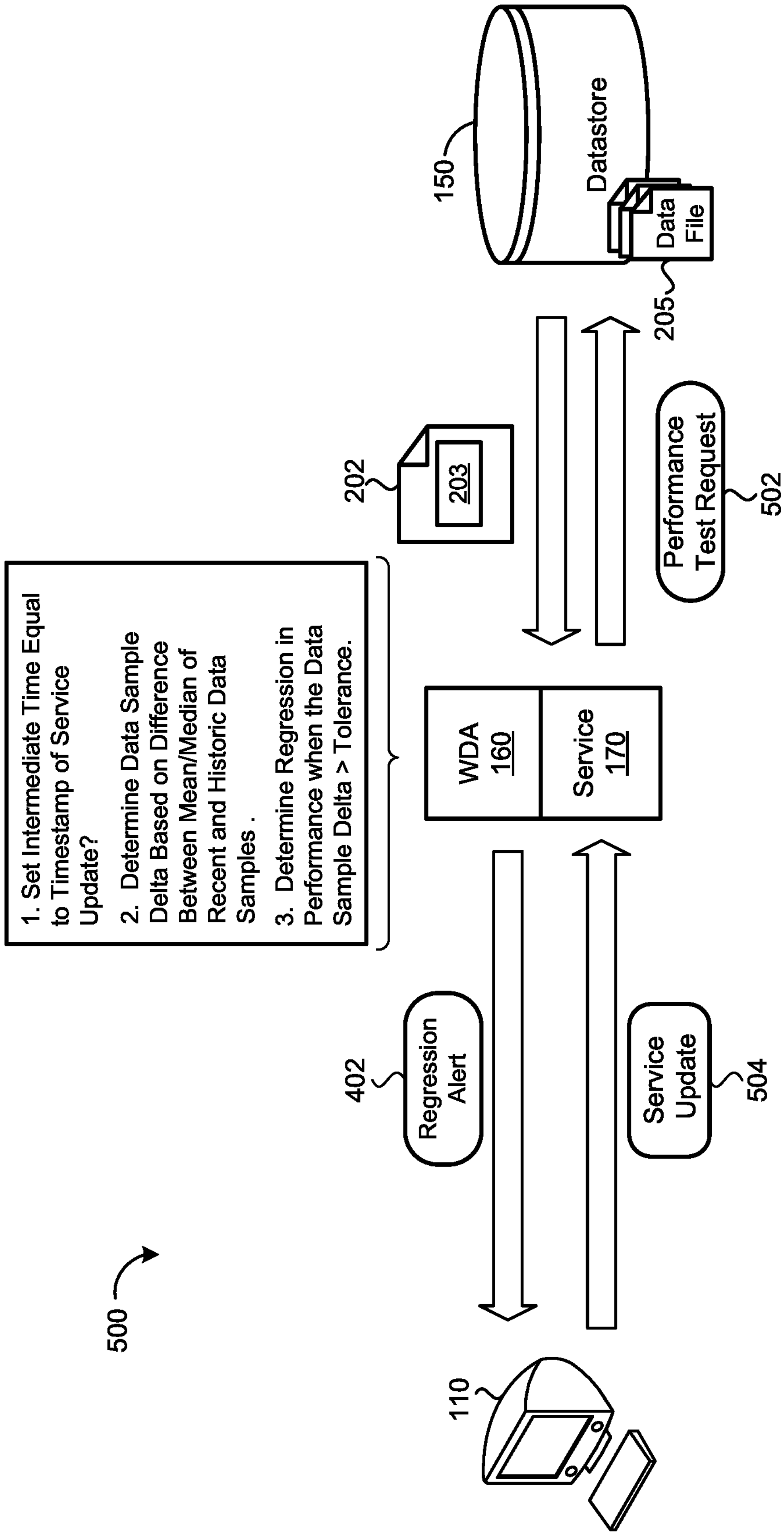
1. Is Data Sample Delta > 0?
   - YES

2. Data Sample Delta > Delta Tolerance?
   - YES

3. Ignore Regression in Performance Determination

Directional Bias
○ No Bias
● Ignore_Increase
○ Ignore_Decrease

FIG. 4B


1. Set Intermediate Time Equal to Timestamp of Service Update?

2. Determine Data Sample Delta Based on Difference Between Mean/Median of Recent and Historic Data Samples .

3. Determine Regression in Performance when the Data Sample Delta > Tolerance.

FIG. 5

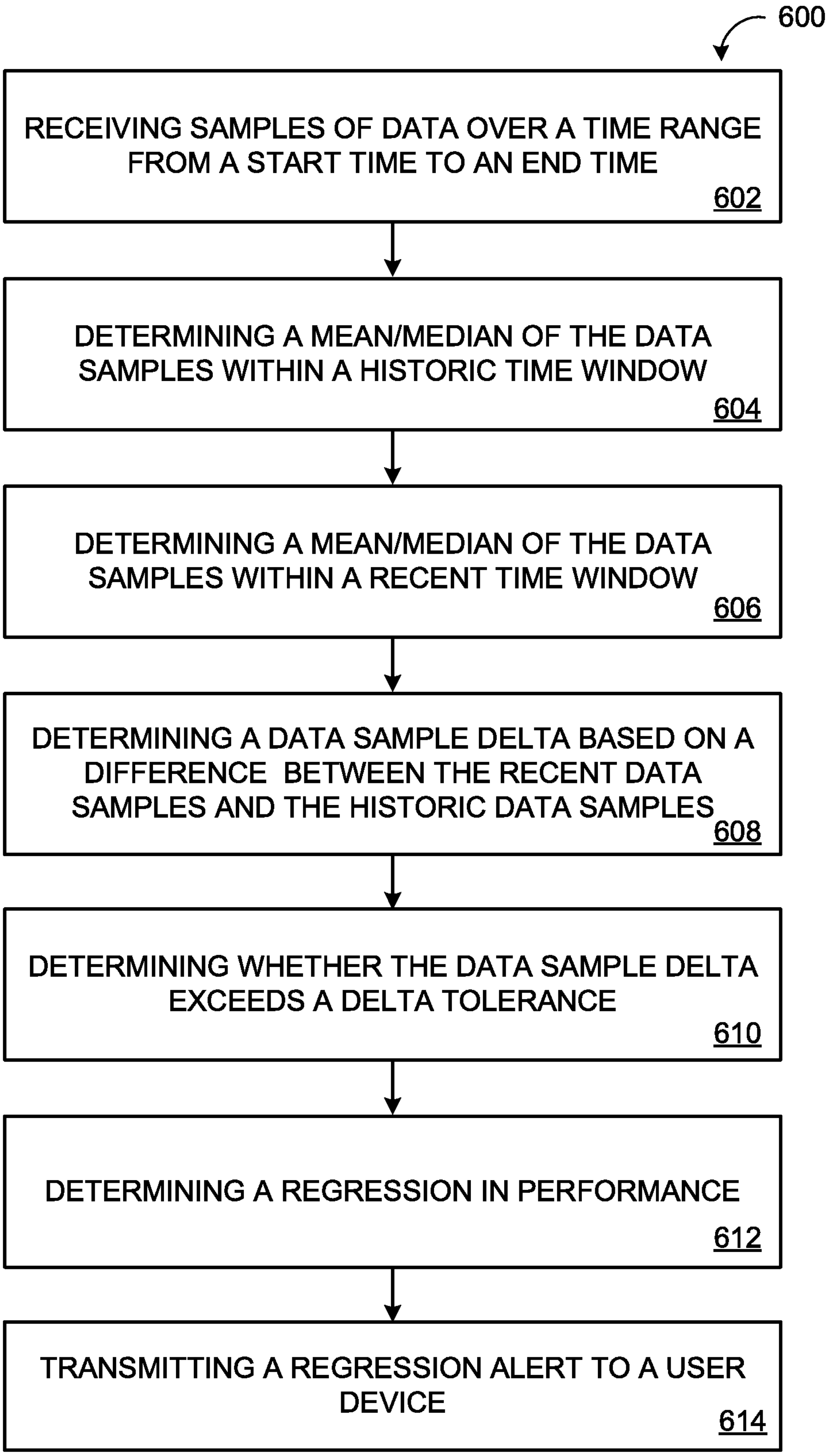

600
RECEIVING SAMPLES OF DATA OVER A TIME RANGE FROM A START TIME TO AN END TIME
602
DETERMINING A MEAN/MEDIAN OF THE DATA SAMPLES WITHIN A HISTORIC TIME WINDOW
604
DETERMINING A MEAN/MEDIAN OF THE DATA SAMPLES WITHIN A RECENT TIME WINDOW
606
DETERMINING A DATA SAMPLE DELTA BASED ON A DIFFERENCE BETWEEN THE RECENT DATA SAMPLES AND THE HISTORIC DATA SAMPLES
608
DETERMINING WHETHER THE DATA SAMPLE DELTA EXCEEDS A DELTA TOLERANCE
610
DETERMINING A REGRESSION IN PERFORMANCE
612
TRANSMITTING A REGRESSION ALERT TO A USER DEVICE
614

FIG. 6

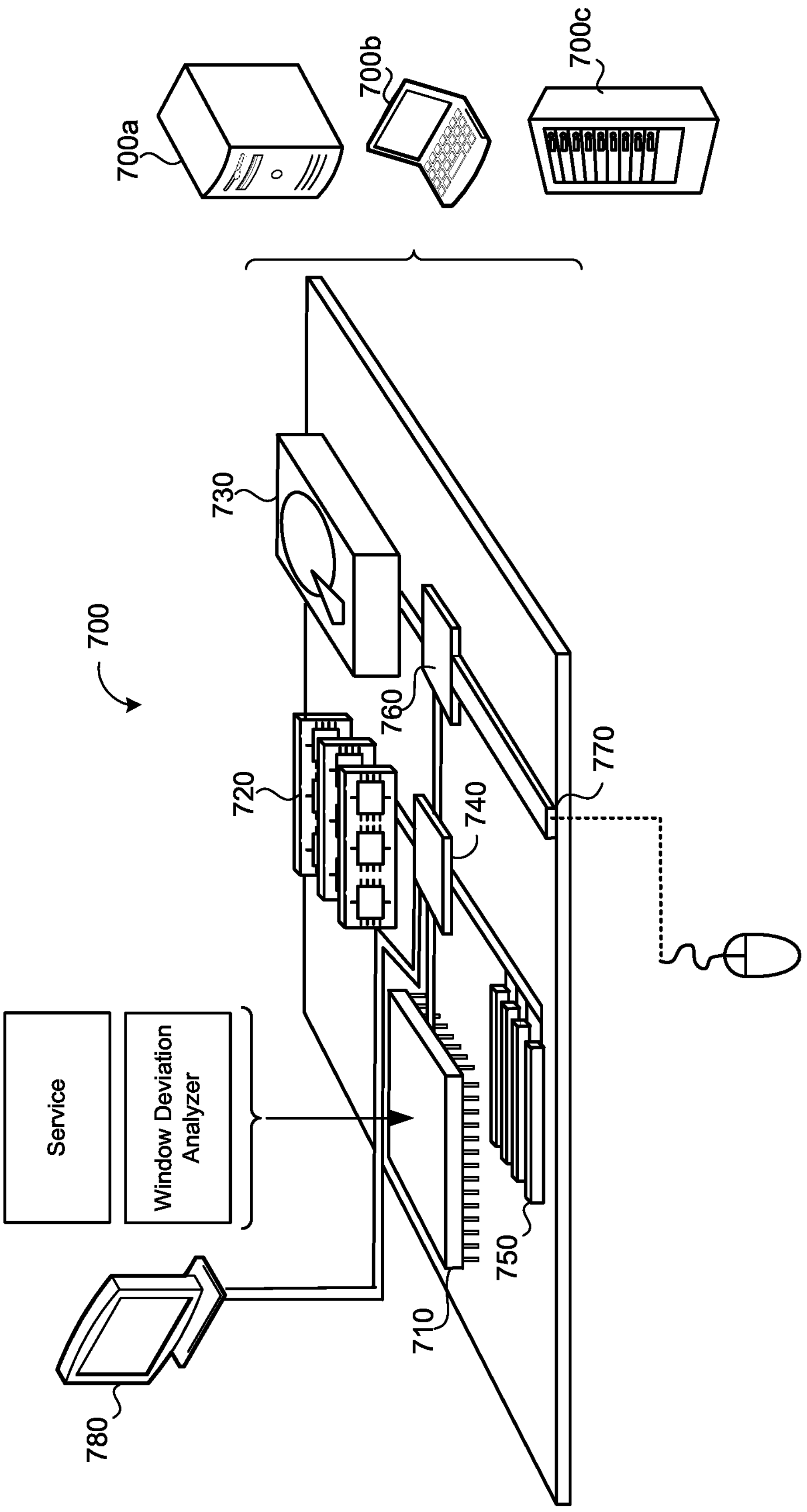

700a
700b
700c
700
730
760
770
720
740
750
710
780
Service
Window Deviation Analyzer

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2017/052388

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F11/36
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

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

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

EPO-Internal , COMPENDEX, INSPEC, IBM-TDB, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 9 176 759 BI (SAHASRANAMAN VIVEK [IN] ET AL) 3 November 2015 (2015-11-03) abstract; figures 1,3,4 column 1, line 35 - line 54 column 2, line 6 - column 4, line 58 column 6, line 43 - column 9, line 30 column 10, line 28 - column 11, line 56 column 14, line 35 - line 43 -----	1-30
A	US 2015/339207 AI (HOD R0NEN [IL]) 26 November 2015 (2015-11-26) abstract; figures 1,2 paragraph [0001] paragraph [0011] - paragraph [0012] paragraph [0017] - paragraph [0026] paragraph [0029] - paragraph [0041] ----- -/- ·	11,26

☒ 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" documentwhich may throw doubts on priority claim(s) orwhich 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 14 December 2017	Date of mailing of the international search report 21/12/2017
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Totir, Felix

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2007/061626 A1 (NELSON BRUCE L [US] ET AL) 15 March 2007 (2007-03-15) abstract; figures 1,3,5 paragraph [0002] - paragraph [0005] paragraph [0014] - paragraph [0029] -----	1-30
A	US 2005/120111 A1 (BAILEY PHILIP G [AU] ET AL) 2 June 2005 (2005-06-02) abstract; figures 1,4 paragraph [0004] paragraph [0024] - paragraph [0040] paragraph [0074] - paragraph [0088] -----	1-30

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2017/052388

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 9176759 Bl	03-11-2015	NONE	
US 2015339207 Al	26-11-2015	NONE	
us 2007061626 Al	15-03-2007	NONE	
us 2005120111 Al	02-06-2005	AU 2005301282 Al	11-05-2006
		CA 2583582 Al	11-05-2006
		EP 1812863 Al	01-08-2007
		US 2005120111 Al	02-06-2005
		Wo 2006049717 Al	11-05-2006