Displaying Information Generated by a Monitoring Program

1. Field of the Invention

The invention relates to a method for monitoring computer events or operations by providing an analytical display. A software application can display positive, negative, and unknown status of computer operations or computer events in a view that provides visual confirmation to the user that one or more computer systems are being monitored. The display can indicate current performance with constantly updated metrics and thresholds.

2. The Prior Art

Various prior methods are known for displaying results of monitored databases. Current products typically only display negative results and do not display the metrics of positive results. Conventional products include those made by Argent, NetIQ, Microsoft, and HP. These products have a central focus on failures. Therefore, there is a need in the art for a visual monitoring system that displays positive results in addition to negative or unknown status information for computer events or computer operations.

3. Summary of the Invention

According to one aspect of the invention, a method for monitoring computer events or computer operations can indicate transaction speed, count, and process thresholds for various events on a computer display. This data can be conveyed in a visual manner that is intuitive for a business or technology owner of the systems that are monitored. To achieve this visual presentation of data, historical data can be stored and the system can display results that are constantly updated. The data can be presented in a logical manner with descriptions that are non technical.

4. Detailed Description of the Technology

Referring now to the drawings, in which like numerals represent like elements throughout the several figures, aspects of the present invention in the exemplary operating environment will be described. Figure 1A illustrates an exemplary architecture for a monitoring system 100. The monitoring system 100 can comprise a monitoring database 135 that communicates with a monitoring program referred to as Ranger Application 105. The monitoring program 105 can provide a graphical user interface 140. The graphical user interface 140 allows a user to monitor the status of operations of other programs running in a data store 110 and a web farm 130. Each data store 110 can comprise one or more different databases 120A, 120B.

The monitoring program or engine 105 can query data store 110 and web farm 130 and store the results in the monitoring database 135. Referring to the graphical user interface 140 of Figure 1A, it can comprise a window that displays several different tables of information. In the exemplary user interface 140 illustrated, there are seven tables: a first table 145; a second table 157; a third table 159a; a fourth table 150; a fifth table 199; a sixth table 193; and a seventh table 159b.

The fourth table 150 in the exemplary embodiment illustrated is entitled, “Ranger Admin.” The fourth table can comprise several different columns 155, 160, 165, 170, 175, 180, 185, 190, 195, and 197 of information. Each column can comprise several icons such as squares. Each square corresponds to an event or status of an operation that is being tracked by the monitoring program 105 that communicates with the monitoring database 135.

Each icon shaped as a square in the exemplary fourth table 150 can comprise a certain color to indicate a particular status. For example, each square icon in the fourth table can indicate positive or negative results of scan or query executed by the monitoring program 105. The graphical user interface 140 can include indicators of a positive or negative results as well as metrics that can be displayed with multiple thresholds.

For example, certain colors as well as additional visual cues such as flashing or changing of a color of each square icon in the fourth table 150 can have an associated meaning as set forth in Table 1 below:

Table 1 - Color Scale

Severity	**Color**
Critical	Blink Red
High	Red then Blink Red
Medium	Yellow then Red
Low	Yellow
Maintenance	Blue
Unknown response	Orange
Paused until {Time}	Gray

The invention is not limited to the color scale listed Table 1 above. Other color scales and metrics are not beyond the scope of the invention.

Referring to now to first and second tables 145 and 157, entitled, "Avg. Seconds Per QF," and "Process Status," respectively, these tables refer to single events but with more detail than the events listed in the fourth table 150. In other words, first table 145 provides several different rows of information that describe one event or operation tracked by the monitoring program 105 and stored in the monitoring database 135. Similarly, the second table 157 also provides several different rows of information that describe one event or operation tracked by the monitoring program 105 and stored in the monitoring database 135. The first and second tables 145 and 157 can be modified by the user to display more or less information about the event being tracked.

Further, additional tables can be created by the user and displayed in the graphical user interface 140. For example, if a user wants to display more information for a particular event that is only tracked in the fourth table 150 as a square icon, he or she could create a table similar to the first and second tables 145 and 157 that display additional information about the event of interest.

Referring to the third and seventh tables 159a, 159b entitled, "QF Version 6 Hour Window," and "LendingTree 3 Hour Window," the events and information tracked by these two tables are not provided by the monitoring database 135. These particular events are "hard coded" (programmed) and displayed on the same user interface 140 as the other tables 145, 157, and 150.

Similarly, the fifth and sixth tables 199 and 193 entitled, “Argent Console 7.X,” and “Argent Console 6.2,” display events and information tracked by third party software. This means that the graphical user interface 140 of the monitoring program 105 can display tables generated by the monitoring program 105 as well as tables generated by other software or hardware (or both) that are separate and independent from the monitoring program 105. The information displayed by the other hardware and software can have the same look and feel as the information tracked by monitoring program 105.

Referring now to Figure 1B, this figure illustrates an exemplary high-level process flow diagram for the system. In step 201, the monitoring application 105 queries the monitoring database 135 to find the last time an event was checked and if it is due yet based on the interval time. Next, in step 203, events that are due to be checked are then submitted by the monitoring database 135 to the monitoring application 105 for checking.

In step 205, events are checked by monitoring program 105. In step 207, results from events generated by the monitoring program 105 are then stored in the monitor database 135. And in step 209, results from those events can then be presented on the user interface or ranger monitor display 140.

Referring now to Figures 2-5, these figures illustrate further details of a graphical user interface system 200 of the monitoring program 105. The system 200 includes several different displays: the main monitoring display 140 of Figures 2-3 (also illustrated in Figure 1A) and a configuration display 210 of Figure 4 that comprises specific event displays 215B.

When the main monitoring display 140 is active, a user can review the specific configuration of an event by moving a pointer over the fourth table 150 of events that is entitled, “Ranger Admin.” For example, if a user positions his or her pointer within focused region 205 of the Ranger Admin table 150 as illustrated in Figure 3, the square icon entitled “LTBW 18 > 0 QF” event 215A could be selected.

After selecting the square icon for event 215A as illustrated in Figure 3, the configuration display 210 of Figure 4 could become activated and shown to the user. The configuration display 210 can comprise detailed information for a specific event, such as for the “LTBW 18 > 0 QF” event 215A indicated by reference numeral 215B of Figure 4.

Referring now to Figure 4, this figure illustrates a configuration window 210 that can include the detailed information 215B for the “LTBW 18 > 0 QF” event 215A. The detailed

information 215B for the "LTBW 18 > 0 QF" event 215A can comprise information divided into two columns 216A, 216B1.

Specifically, referring now to Figure 5 that is an enlarged view of configuration window 210 of Figure 4, the detailed information can comprise a first column 216A and a second column 216B1 of information.

The first or left side column 216A can comprise information such as a short description 220 for the event 215; a long description 225 of the event 215; an event type classification 230; a poll interval time limit 235; an enabled/disabled status 240; an error severity level 245; a write XTD History or error status 250; a pause time 255; a pause time remaining status 260; a display status 20; and an application ID 270.

The second or right side column 216B1 for the detailed information 215B of an event 215A provides status information that is dependent on the type of event 215. There can be at least two types of events 215A: a database (SQL) event or a web query event. In Figure 5, the right side column 216B1 illustrates a database (SQL) type of event. Meanwhile, in Figure 6, the right side column 216B2 illustrates a web query type of event.

Referring now to the first column 216A of the detailed information 215B for the "LTBW 18 > 0 QF" event 215A, the poll interval 235 can refer to how often status for this event is checked. The value of the poll interval 235 can be set to any amount of time, such as in minutes.

The enabled status field 240 indicates whether the alert for this event is active or not. If the enabled status field 240 is set to "Yes," then the alert is active. If it is set to "No," the alert is not active. Inactive or alerts that are turned off, according to one exemplary embodiment, are indicated with a blue color. Therefore if the enabled status 240 is set equal to "No," then color of the square icon for the "LTBW 18 > 0 QF" event 215A will be colored blue in table 150 of Figures 2 and 3.

The error severity status 245 allows a user to indicate a level of severity for a particular alert. Each level within a range of levels can be assigned a specific different color. For example, a "medium level" severity status 245 can be associated with the color of yellow. Meanwhile, a "high level" severity status 245 can be associated with the color of red. See Table 1 discussed above.

According to one exemplary embodiment, for a medium level severity status 245, for the first two alerts, the color can be yellow. But after the second alert, the color can be changed to

red. In other words, colors for a specified severity status 245 can be changed to indicate sequential failures in a row.

So, for a medium severity status 245 example, if the alert fails at 9:00 am and it fails again at 9:15 am, the color of the square icon for the "LTBW 18 > 0 QF" event 215A in the Ranger Admin table 150 of Figure 2 will be yellow again. And then at 9:30, it would turn red and as long as the event 215 continues to fail.

The Write XTD History or Error Feature 250 in column 216A of Figure 5 indicates whether some additional historical data about when the event failed should be stored. In some exemplary embodiments, this feature 250 is not used because a record history file can be used.

A time paused feature 255 allows a user to manually take place an event 215 into a temporary maintenance status. When an event 215 is placed in a temporary maintenance status, the color of its square icon as illustrated in the Ranger Monitor window 150 can change. In one exemplary embodiment, the square icon for an event 215 can change to the color blue to indicate a temporary maintenance status.

The pause time remaining feature 260 allows a user to place an event in a temporary maintenance status for a specified period of time such as in minutes. This allows a user to place an event into a temporary maintenance status while the user troubleshoots an event 215 so that the event 215 does not indicate its failed status (such as a blinking red square icon). In this way, the user can place an event 215 into a temporary maintenance status so that its color could be a solid, non-flashing blue square icon. After the pause time provided, the event 215 will go back to a normal operating status. If the event 215 is in alert status, the square blue icon will change in color to a flashing red.

The display status 265 determines whether the results of an event query get displayed when the results are successful or not. The display status 265 can be either a "yes" or "no" value. If the display status 265 is set equal to "no," then a square icon for the "LTBW 18 > 0 QF" event 215A in the Ranger Admin table 150 of Figure 2 is not displayed if there is not an alert. But if the display status is set equal to "yes," then a square icon for the "LTBW 18 > 0 QF" event 215A in the Ranger Admin table 150 of Figure 2 is displayed at all times, irrespective of its alert status. The display status 245 allows for the conservation of space in table 150 so that only alerts that have negative results are displayed if desired.

Application ID 270 indicates which hardware device such as the engine or monitoring program 105 that executes the description 225 for a particular event 215. The poll interval 235 discussed above determines how often the hardware device runs the description 225 for the event 215.

For the database (SQL) event 215B illustrated in Figure 5, the right side column 216B can comprise a database server identifier 275; a database identifier 280; a procedure type 285; a procedure 290; a low limit 295; and a high limit 297.

The DB Server field 275 in the right side column 216B represents the database name or server name that is being checked by the alert. In exemplary embodiment illustrated in Figure 5, the DB Server Status 275 comprises the name "LTDB1" that denotes a physical piece of hardware in a the computer network that the user wants to monitor by running SQL commands against it.

Database field 280 in the right side column 216B designates the database that can be accessed from the server named in the DB Server field 275. In the example illustrated, DB Server LTDB1 can comprise twenty different databases. In the example, the alert 215 is checking a database named "Commerce" to get its status information.

Procedure type field 285 indicates the type of event that is being tracked. Procedure type field can be either a SQL statement or a web check. Procedure input section 290 describes what production command that is being run against that production database 280. Procedure input section 290 receives the logical statements that are run against the database. The status of qualification forms (QFs) is addressed by the exemplary logic present in the procedure input box 290 illustrated in Figure 5.

Under the procedure input section 290 are the low limit field 295 and high limit field 297. The low limit field 295 describes the lower threshold that is acceptable for the condition set forth in the procedure input section 290. Anything below the lower threshold would trigger a change in the alert 215.

For the high limit field 297, anything over this threshold will trigger an alert. Any amount equal to or less than the high limit field will not trigger an alert 215. Therefore, for the example illustrated in Figure 5, any results less than the lower limit of one qualification form (QF) or greater than the higher limit of two thousand qualification forms (QFs) will produce an alert condition that will cause the alert 215 to change color and flash for the operator.

Referring back to the system diagram of Figure 1A, for the particular event 215A illustrated in Figures 3 through 5, its status is checked every 15 minutes as set in the poll interval field 235 of Figure 5.

Referring again to Figure 1A, engine or monitoring program 105 runs the queries as is indicated by the arrows below program 105. Program 105 conducts scans of the data stores 110 for the server and respective database 120 that are identified in the DB Server field 275 and database field 280 of Figure 5.

Depending upon how complicated the procedure call as set forth in the procedure input section 290 of Figure 5, the program 105 of Figure 1A can retrieve results from the databases 120 in a fairly instantaneous fashion. The amount of time to complete a query by the program 105 depends on how the query is written. For some exemplary queries, results can be retrieved usually in time frames of less than one second.

Once the results of the executed query are obtained, program 105 can store the results in the monitoring database 135, which is physically and logically separated from the databases 120. The monitoring database 135 can comprise the configuration data that is discussed with reference to Figures 4-5. The monitoring database 135 contains all of the information that is displayed in the user interface 140 illustrated in Figures 1 and 2.

In summary, referring back to the square icon of Figure 3 entitled “LTBW 18 > 0 QF” event 215A, this icon comprises a mini-reporting system for this specific event 215 because the database is queried with specific queries that are listed in the detailed information window 215B of Figure 5.

For this specific example of the “LTBW 18 > 0 QF” event 215A, if the results are greater than one QF (qualification form) as listed in the low limit 295 and less than two-thousand as listed in the high limit 297 of Figure 5, then the color of the square icon for the “LTBW 18 > 0 QF” event 215A in the Ranger Admin table 150 of Figure 2 will not change. However, if the results of the database query for the “LTBW 18 > 0 QF” event 215A event go outside the low and high limits 295, 297, then the color of the square icon for the “LTBW 18 > 0 QF” event 215A in the Ranger Admin table 150 of Figure 2 will change to a different color to indicate an alert status to the user.

Referring now to Figure 6, this figure illustrates another enlarged view of configuration window 210 that includes a first column 216A and a second column 216B2 of information.

Figure 6 is similar to the configuration window 210 illustrated in Figure 5. Therefore, only the differences between Figure 5 and Figure 6 will be described. The second column 216B2 comprises a universal resource locater address that corresponds to the web farm 107 that is being evaluated for this particular event.

The second column 216B2 can further include a compare string 510 in which a user can insert a string for the monitoring application to find in a web farm 107. The second column 216B2 can also include an logical operator field 515 that can be adjusted or modified by the user.

Database Schema

Referring now to Figure 7, this figure illustrates a database schema that can be used by the monitoring program 105 and database 135 to track events. According to one exemplary embodiment of the technology, the schema has eighteen tables labeled as 705-790. Table 705 represents the core of the schema and all other tables have a relationship to it. One important element in table 705 is the process id/process description. This information is displayed on Figure 4 in the graphical user interface 210 and further in the details illustrated in Figure 5 of the graphical user interface 220 (short description). The following Correlation Chart identifies the remaining relationships between Figure 5 &6 and tables listed in Figure 7.

Correlation Chart for Figure 7

Tables in Figure 7	Cross Reference to Other Figures	Reference Numeral in Other Figures	Group
705	5	220	Tprocess
705	5	225	Tprocess
705,710	5	230	
705	5	235	
705	5	240	
705,715	5	245	
705	5	250	
705	5	255	
705	5	260	
705	5	265	
705,730,735,740	5	270	ApplicationID -> Site
770,745,750	5	275	
745,750	5	280	
770,780	5	285	SQL Statement
770	5	290	SQL Statement
770	5	295/297	SQL Statement
785	6	505	Web Search
790	6	510	Web Search
790	6	515	Web Search

While several embodiments of the present invention have been shown and described, it is to be understood that many changes and modifications may be made thereunto without departing from the spirit and scope of the invention.