

DBPLUS
Performance Monitor for Microsoft SQL Server
description of changes in version 2022.3

Table of contents

| | | |
|----------|--|----------|
| 1 | Anomaly monitor – improvements | 3 |
| | Report | 3 |
| | Analysis of locking problems..... | 3 |
| 2 | Trace session - filter by Hash value | 4 |
| 3 | Bug fixes and improvements | 5 |
| | 3.1. Improvement of Plan Guide generation mechanism..... | 5 |
| | 3.2. Improvements to the lock screen | 5 |
| | 3.3. Adding Latch BUFFER to the Latches screen..... | 5 |

Below is a list of changes to the DBPLUS Performance Monitor system for Microsoft SQL Server database monitoring.

New in 2022.3 version

1 Anomaly monitor – improvements

Anomaly monitor mechanism is a module responsible for automatic real-time detection of performance problems in the monitored database.

Report

The Anomaly Monitor report has been modified in the latest version. This report is available at the instance details level in the *Anomaly Monitor - Reports menu*. The change was to the default version of the report. Users can still create and modify their own versions of the report.

The changes concern the presentation of information on the general performance characteristics of the instance, as well as the presentation of information on the main performance problems detected in the instance in the period for which the report was prepared.

Analysis of locking problems

The problem of locks is one of the most common performance problems in relational databases. In the Performance Monitor application, there is an Anomaly Monitor module in which occurrences of events detected by the application affecting the performance of the monitored instance are presented. One of the series of events is Locks.

In the latest version, in addition to sealing the mechanism for collecting locks, functionality was added to verify the causes of locks. The following scenarios have been distinguished in the application:

High locking due to sleeping session

The problem caused by leaving an inactive session with an open transaction that holds unapproved changes in the database. The problem cannot be solved on the database server side. Application code, connection and transaction management should be verified to resolve the problem.

High locking due to long transaction

The problem caused by opening a long transaction in the database. The process causing the locking and the application code should be verified. The recommendation is to split the process into using shorter database transactions. In addition, verify the transaction isolation mode and check whether the blocked queries, used optimal execution plans.

High locks due to DDL commands

A process running a DDL command such as ALTER, CREATE, DROP, DBCC took part in the blocking and could drastically increase blocked sessions. Run these types of commands during off-hours or when the database is less loaded.

High locking due to long running statement

A problem caused by a long-running query (or a large number of executions). Verify the query's statistics, performance and execution plan.

High locking due to long running statement (read only transaction)

Problem caused by a long-running query (or a large number of runs) reading data. Query statistics, performance and execution plan should be verified. In addition, the transaction isolation mode should be verified.

High locking due to long transaction or processing on application site

Problem caused by opening a long-running database transaction or application-side processing. The process and application code should be verified. Recommendation is to split the process to use shorter database

transactions. In addition, verify the transaction isolation mode and check whether the blocked queries, used optimal execution plans.

High locking due to processing on application site

Problem caused by application-side processing. Verify the process and application code. In addition, check whether the blocked queries, used optimal execution plans.

If any of the scenarios occurs, information about the event will be presented in the application on the Dashboard screen, Anomaly Monitor as well as other screens available in the application.

In addition, information on the cause of the blockages will be available in the report for the period in which the problems described above occurred.

Note that information on the causes of locks requires additional data collection during monitoring. Therefore, the causes of blockages will be visible only for problems that were detected after the application was updated to the latest version.

2 Trace session - filter by Hash value

In the latest version, the mechanism has been extended by the possibility to search for user sessions which, at the time of verification, execute the query indicated in the filter. The collected data can be analyzed from the application level or exported to *.csv file.

Result for trace using Hash value in filter:

| Date from | Date to | Last status change | Trace status | Interval (s) | Max sessions to trace | Number of traced rows | Trace filters |
|---------------------|---------------------|---------------------|--------------|--------------|-----------------------|-----------------------|------------------------------------|
| 2022-10-07 13:29:00 | 2022-10-07 13:39:00 | 2022-10-07 13:30:10 | running | 15 | 1 | 10 | 107 Query Hash: 0xC89D7D52FB18B9BE |

| Logdate | Session start | Last request date | Transaction start | Session id | Program | NT user name | Login name | Original login name | Status | Hostname | Context info | Wait Name | Database | Query Hash | Plan Hash | Elapsed Time [seconds] | Cpu Time [seconds] | Blocking Sid | Command |
|---------------------|---------------------|---------------------|---------------------|------------|----------|--------------|-------------|---------------------|---------|----------|--------------|-----------|----------|-----------------------|------------|------------------------|--------------------|--------------|---------|
| 2022-10-07 13:29:45 | 2022-10-07 13:29:45 | 2022-10-07 13:29:45 | 2022-10-07 13:29:45 | 86 | SQLAgent | sqlsa_bctest | INTERsql... | INTERsql... | running | SQL22 | | LCK_M_X | msdb | 0xD9A25E...0x8AASF... | 0x8AASF... | 0 | 0 | 87 | SELECT |
| 2022-10-07 13:29:44 | 2022-10-07 13:29:42 | 2022-10-07 13:29:42 | 2022-10-07 13:29:42 | 145 | SQLAgent | sqlsa_bctest | INTERsql... | INTERsql... | running | SQL22 | | LCK_M_X | msdb | 0xD9A25E...0x8AASF... | 0x8AASF... | 2 | 0 | 130 | SELECT |
| 2022-10-07 13:29:44 | 2022-10-07 13:29:42 | 2022-10-07 13:29:42 | 2022-10-07 13:29:42 | 142 | SQLAgent | sqlsa_bctest | INTERsql... | INTERsql... | running | SQL22 | | LCK_M_X | msdb | 0xD9A25E...0x8AASF... | 0x8AASF... | 2 | 0 | 130 | SELECT |
| 2022-10-07 13:29:44 | 2022-10-07 13:29:42 | 2022-10-07 13:29:42 | 2022-10-07 13:29:42 | 141 | SQLAgent | sqlsa_bctest | INTERsql... | INTERsql... | running | SQL22 | | LCK_M_X | msdb | 0xD9A25E...0x8AASF... | 0x8AASF... | 2 | 0 | 130 | SELECT |
| 2022-10-07 13:29:44 | 2022-10-07 13:29:42 | 2022-10-07 13:29:42 | 2022-10-07 13:29:42 | 132 | SQLAgent | sqlsa_bctest | INTERsql... | INTERsql... | running | SQL22 | | LCK_M_X | msdb | 0xD9A25E...0x8AASF... | 0x8AASF... | 2 | 0 | 130 | SELECT |
| 2022-10-07 13:29:44 | 2022-10-07 13:29:41 | 2022-10-07 13:29:41 | 2022-10-07 13:29:41 | 130 | SQLAgent | sqlsa_bctest | INTERsql... | INTERsql... | running | SQL22 | | LCK_M_X | msdb | 0xD9A25E...0x8AASF... | 0x8AASF... | 3 | 0 | 109 | SELECT |
| 2022-10-07 13:29:44 | 2022-10-07 13:29:42 | 2022-10-07 13:29:42 | 2022-10-07 13:29:42 | 128 | SQLAgent | sqlsa_bctest | INTERsql... | INTERsql... | running | SQL22 | | LCK_M_X | msdb | 0xD9A25E...0x8AASF... | 0x8AASF... | 2 | 0 | 130 | SELECT |

3 Bug fixes and improvements

3.1.Improvement of Plan Guide generation mechanism

The automatic PlanGuide script generation mechanism related to the detection of the mechanism's failure to support the query syntax has been improved. Some queries were erroneously indicated as not supporting the Plan Guide mechanism. The problem has been resolved.

3.2.Improvements to the lock screen

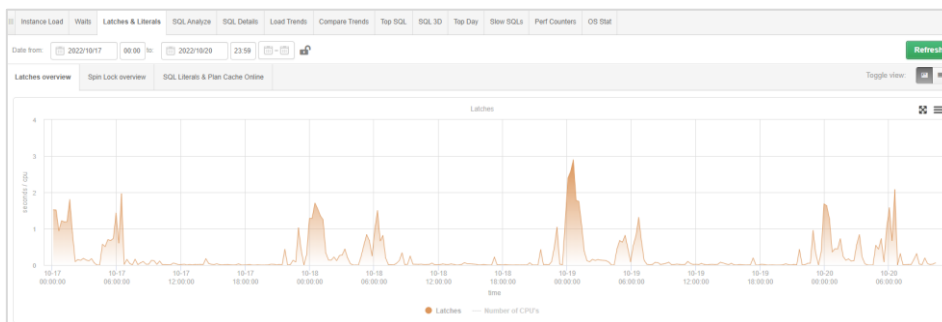
Another fix is related to the presentation of sessions participating in a lock on the lock screen. In particular configurations found in the monitored databases, the lock tree was not presented. The problem has been corrected.

Another fix is related to the presentation of the Last Request Start Time date on the lock screen. The fix is related to a database SQL engine error related to the failure to update the date at the level of the session executing the command. The problem has been corrected at the application level.

3.3.Adding Latch BUFFER to the Latches screen

In the latest version, we have made the Latch information on the Load trends screen and Latches overview screen more consistent. Until now, the Latches **BUFFER** presentation was omitted from the Latches overview screen.

Latches screen:



Load trends screen:

