DBPLUS
better performance

DBPLUS Performance Monitor description of changes in the versions 2018.4.1,2018.4.2

Date: December 21st, 2018

DBPLUS
better performance

Table of Contents

1 New in version 2018.4.1,2018.4.2 3
1.1 SQL Plan Guides management 3
1.1.1 Object search 3
1.2 Permission management in the DBPLUS Performance Monitor 4
1.2.1 Own permissions 4
1.2.2 Inherited permisions form parents 5
1.3 Information about statistics from the OS 7
1.4 Grup wait by class, screen Waits > Analyze 8
1.5 Session menu 8
1.5.1 Searching for sessions for a given waits 8
1.5.2 Information about sessions version store Usage 9
1.6 Anomaly Monitor 10
1.6.1 Problems viewer in the SQL Instance 10
1.6.1.1 Reasons Analysis 10
1.6.1.2 Reasons Overwiew 12
1.6.2 Setting a class for a given cause of the problem 13
1.6.3 Change in verifying the change of the query explain plan 13
1.6.3 New parameter controlling the alert function 14
1.7 General Improvements 15
1.7.1 The ability to export Performance Counters statistics 15
1.7.2 The ability to generate a Performance Report in hourly mode 15

Below we present a list of changes in the DBPLUS Performance Monitor system for monitoring MSSQL instances.

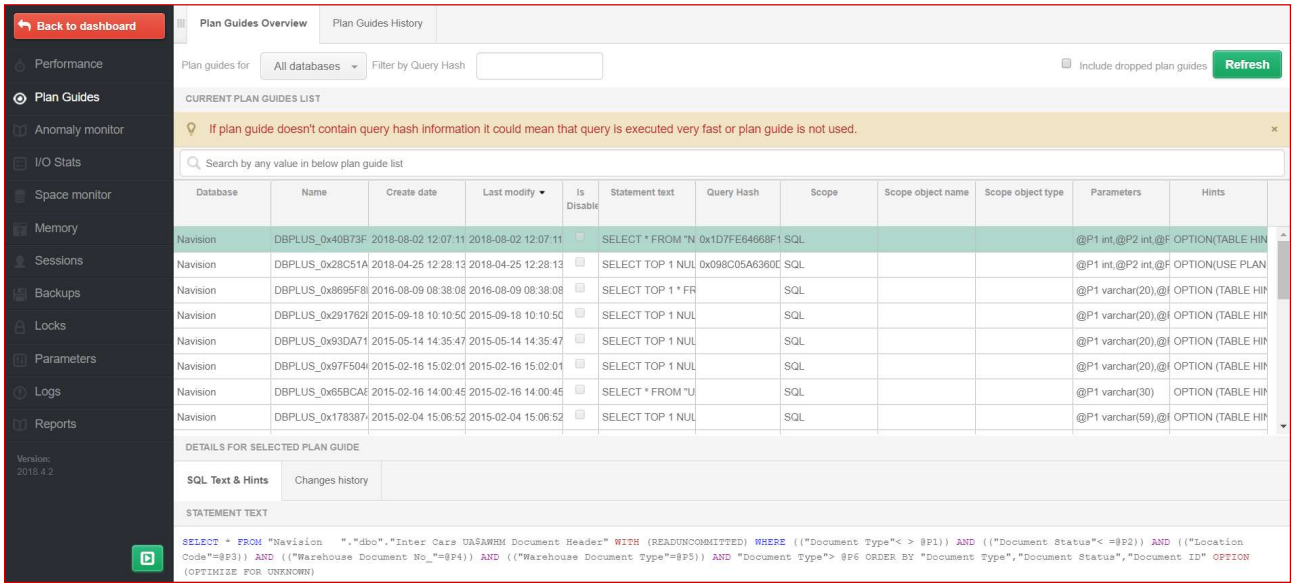
1 New in version 2018.4.1,2018.4.2

1.1 *SQL Plan Guides management*

New version of the application adds the ability to manage objects such as Plan Guides.

1.1.1 Object search

For this purpose, a new Plan Guides menu has been added, available from Instance Analysis for each instance. Information on the Plan Guides established in a given instance is available on the screen. Current information as well as historical data are available.



The Plan Guide Overview tab contains the following information as:

- Database name
- Name – Plan Guide name,
- Create date
- Last modify – date of last modification
- Is Disable – information about Plan Guide status,
- Statement text
- Query Hash – query ID assigned with Plan Guide
- Scope – [OBJECT/SQL/TEMPLATE]
- Scope object name
- Scope object type –(e.g. procedurę, functions)
- Parameters – list of parameters
- Hints – hints related with Plan Guide

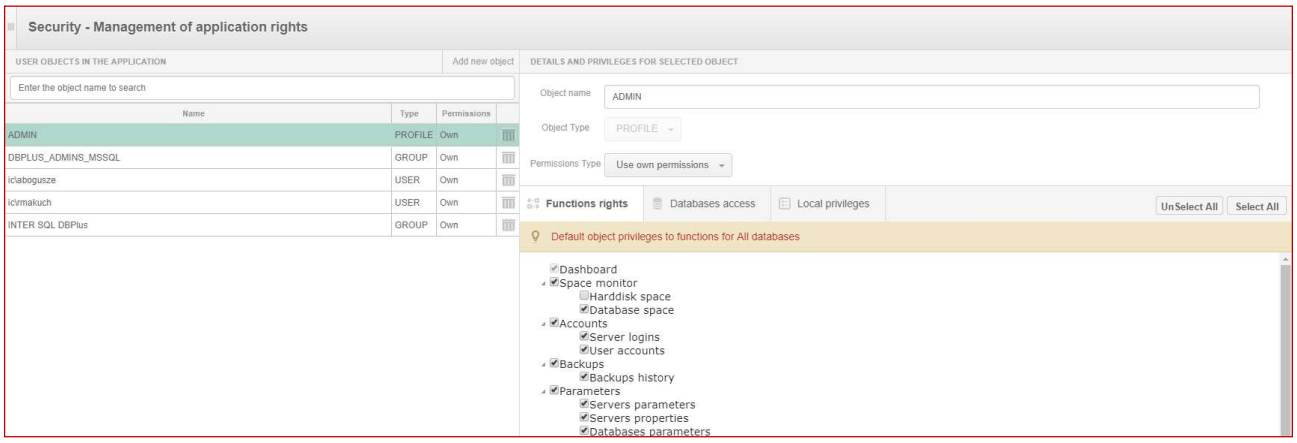
Note! Not all Guide Plan will be assigned Query Hash. This will refer specifically to those Guide Plans that have been created in the past and for which queries are not currently performed.

After clicking on the row in the table, below (the SQL Text & Hints tab), the content of the query will be presented as well as the used hints within the plan. The Changes History tab presents information about what changes were made to a given Plan Guide (e.g. Insert / Change / Drop)

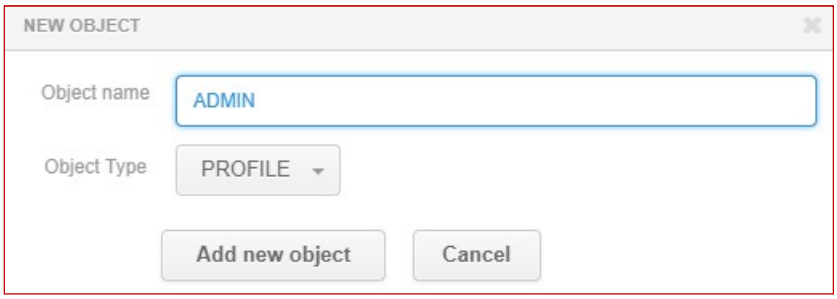
In addition, the Plan Guide History tab stores information about all Plan Guide in the SQL Instance. To search for a Historical Plan Guide, select the appropriate date range.

1.2 *Permission management in the DBPLUS Performance Monitor*

In the new version of the application, the functionality of giving access to the DBPLUS Performance Monitor screens has been modified. In the new version, the PROFILE access object has been added, which allows assigning appropriate access to the profile and then granting rights by assigning the profile to the user. The way of granting access to each group of objects has also been modified.

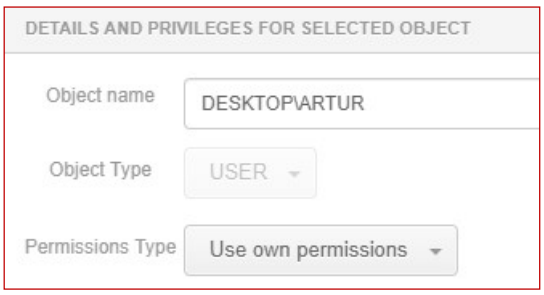


In order to create a new object, e.g. a profile (PROFILE), click on [Add new object], then select the object type "PROFILES" and give the name of the object.



To assign permissions to a given object, select it from the list on the left side of the screen. After clicking on the object on the right side, the page with the access configuration will be displayed.
First you need to choose whether the permissions will be:

- own (Use own permissions).
- inherited permissions form parents.



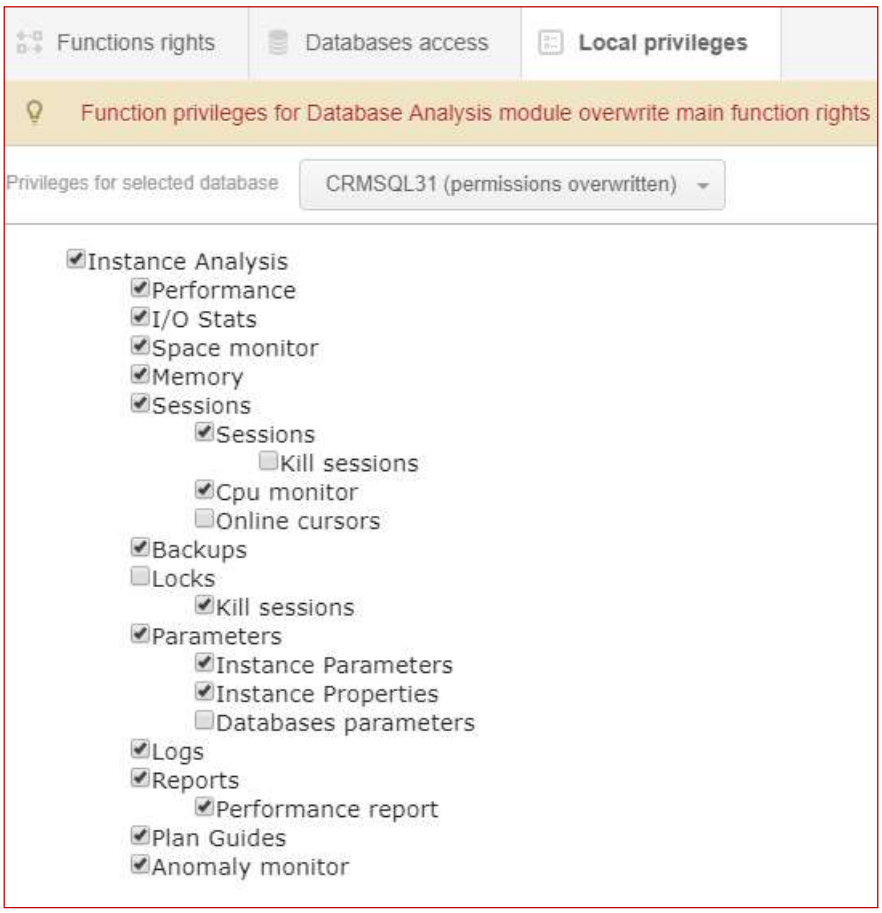
1.2.1 Own permissions

If you choose (own permissions), you have three tabs to configure permissions:

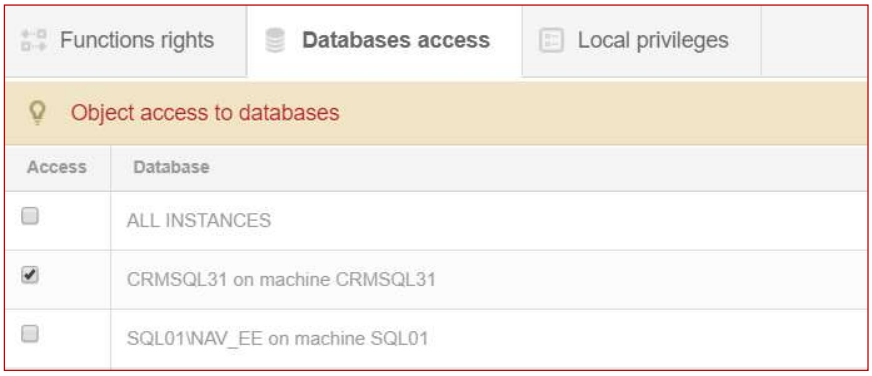
- Function rights,
- Databases access,
- Local privileges.

Functional settings allow you to give rights to pages or functionality in the application at the global level for a given user / group or profile for all databases. You can override these rights by granting custom permissions for a specific SQL instance. Custom permissions can only be changed for the Instance Analysis module. Local

permission is superordinate to a given SQL Instance in relation to functional rights. If you assign custom permissions, the (permissions overwritten) message will be displayed next to the SQL instance name.

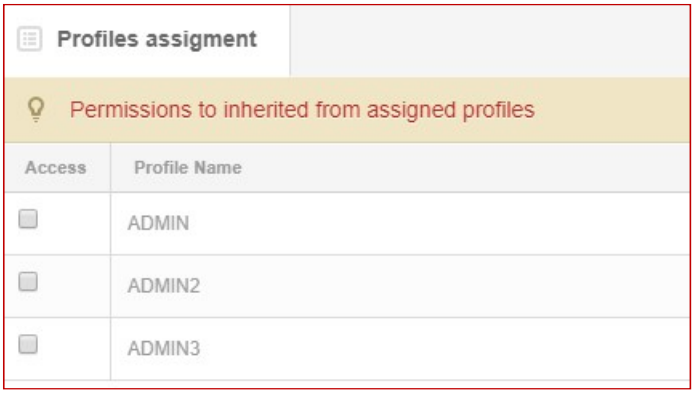


In addition, you can restrict access to specific SQL Instance. To do this, in the Database access tab, select the appropriate check boxes for a given database or select ALL_INSTANCES. If certain bases are restricted, this will also limit the Local privileges tab.

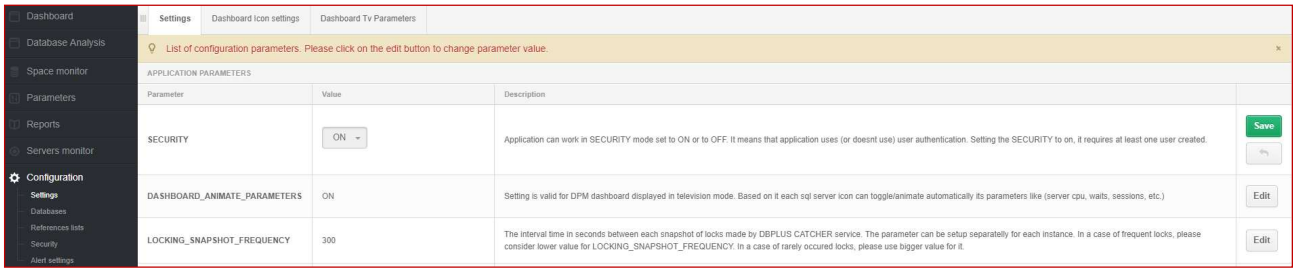


1.2.2 Inherited permisions form parents

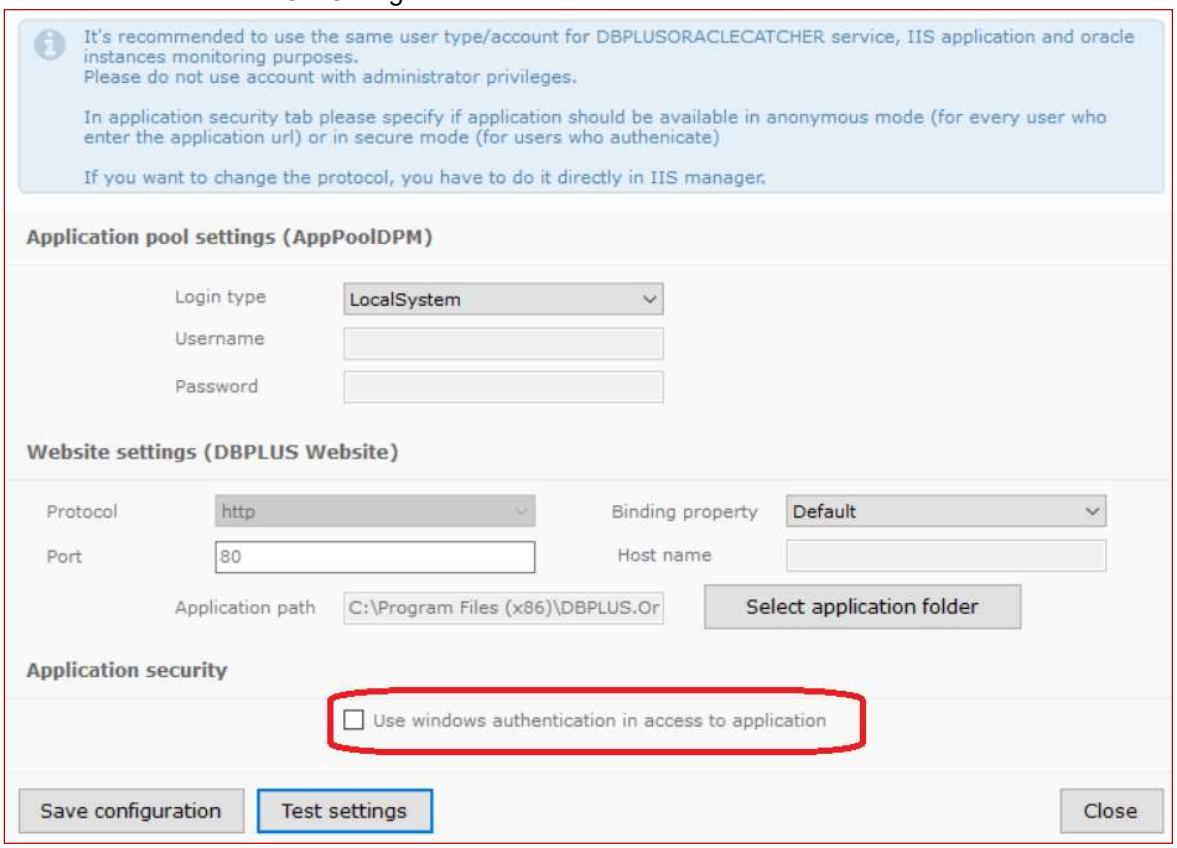
If you choose inherited rights, you can specify which profile or profiles to use for a given user or user group. Each profile contains a list of objects and access to which. Granting permissions to multiple profiles for the user will result in the entitlement for a given user being the sum of rights for selected profiles.



Attention! In order to enable the functionality of limited access to the application, you must change the settings at the level of the DBPLUS Configuration Wizard> Applications settings> Applications Options> Configure. As well as change the status of the **SECURITY** parameter to ON

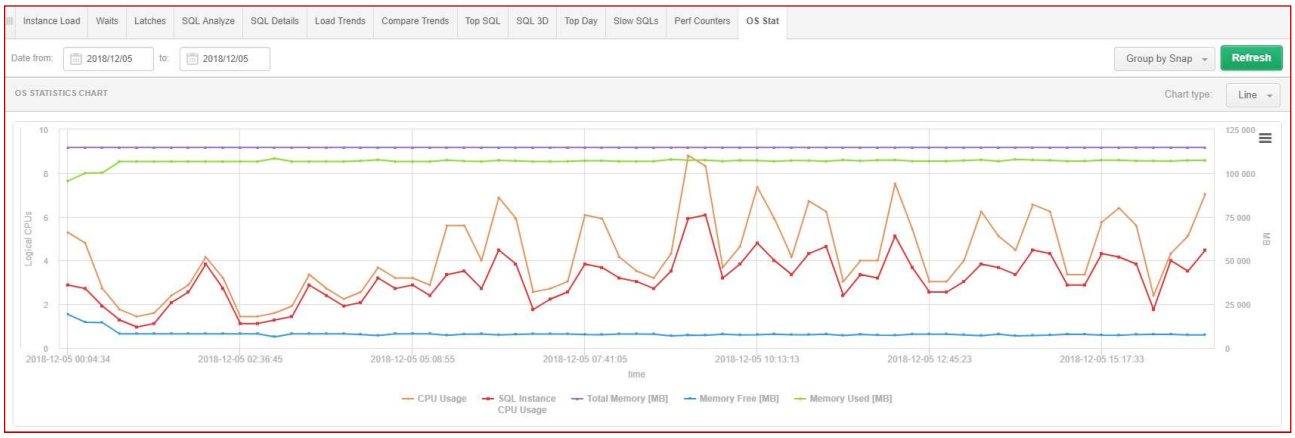


Below screen with the DBPLUS Configuration Wizard:



1.3 *Information about statistics the OS*

In the new version of the application, information about statistics collected at the operating system level has been added:



The OSS Stat tab contains the following information as:

- Logical CPUs – number of available processor,
- SQL Instance Logical CPUs – number of available processor on SQL Instance,
- CPU Idle [Seconds] – the number of processor inactivity seconds, relative to all processors
- CPU Usage [Seconds] - number of seconds in which the processor was busy executing the user or kernel code, including all processors on the server
- SQL Instance CPU Usage – [Seconds] - number of seconds in which the processor was busy executing the user or kernel code, including all processors on the SQL Instance,
- Total Memory [MB] - total amount of physical memory.
- Memory Free [MB] – total amount of free physical memory.
- Memory Used [MB] - total amount of used physical memory.

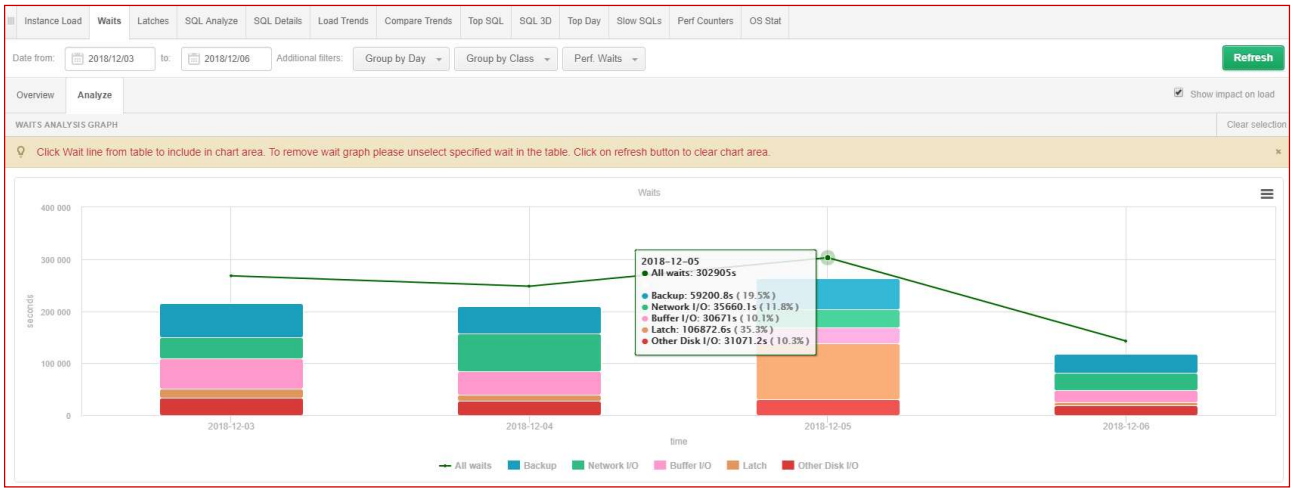

OS STAT STATISTICS

Clear selection

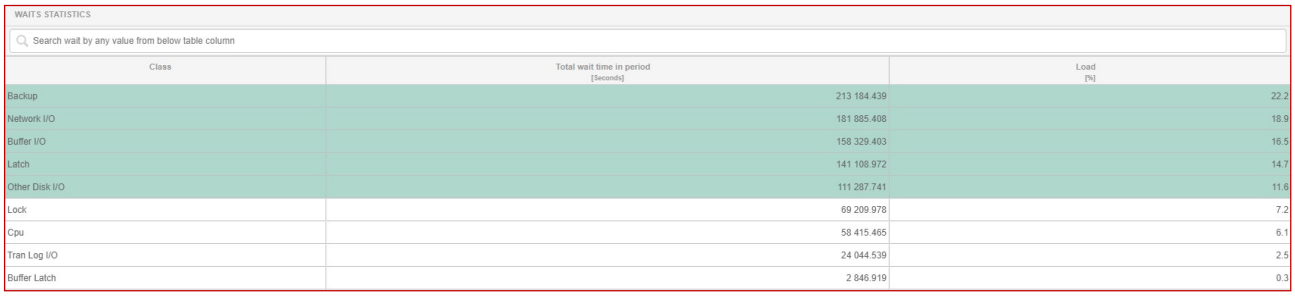
Logdate	Logical CPUs	SQL Instance Logical CPUs	CPU Idle	CPU Usage	SQL Instance CPU Usage	Total Memory [MB]	Memory Free [MB]	Memory Used [MB]
2018-12-05 00:04:34	16	16	10.72	5.28	2.88	114 687 MB	19 158 MB	95 529 MB
2018-12-05 00:19:47	16	16	11.20	4.80	2.72	114 687 MB	14 723 MB	99 964 MB
2018-12-05 00:35:00	16	16	13.28	2.72	1.92	114 687 MB	14 499 MB	100 188 MB
2018-12-05 00:50:13	16	16	14.24	1.76	1.28	114 687 MB	8 096 MB	106 591 MB
2018-12-05 01:05:26	16	16	14.56	1.44	0.96	114 687 MB	8 092 MB	106 595 MB
2018-12-05 01:20:39	16	16	14.40	1.60	1.12	114 687 MB	8 091 MB	106 596 MB
2018-12-05 01:35:52	16	16	13.60	2.40	2.08	114 687 MB	8 092 MB	106 595 MB
2018-12-05 01:51:05	16	16	13.12	2.88	2.56	114 687 MB	8 093 MB	106 594 MB

1.4 *Grup wait by class, screen Waits > Analyze*

The new version of the software has added wait sort functionality by class. Depending on the period you selected, you can now see the share of a given class in the total number of waits. The functionality works for grouping after performance wait and for all waits.



The data is also available in a tabular version. A validation class was assigned to each valid.



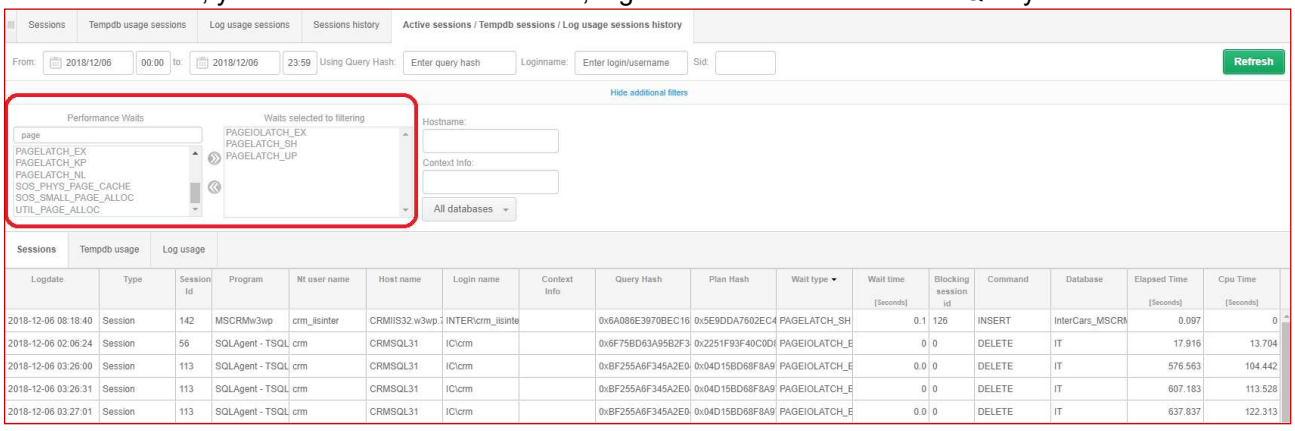
Class	Total wait time in period [Seconds]	Load [%]
Backup	213 184.439	22.2
Network I/O	181 885.408	18.9
Buffer I/O	158 329.403	16.5
Latch	141 108.972	14.7
Other Disk I/O	111 287.741	11.6
Lock	69 209.978	7.2
Cpu	58 415.465	6.1
Tran Log I/O	24 044.539	2.5
Buffer Latch	2 846.919	0.3

1.5 *Session menu*

The new version of the application adds the ability to search information about the historical user's session using a given type of wait and also for the session using Tempdb the version store information.

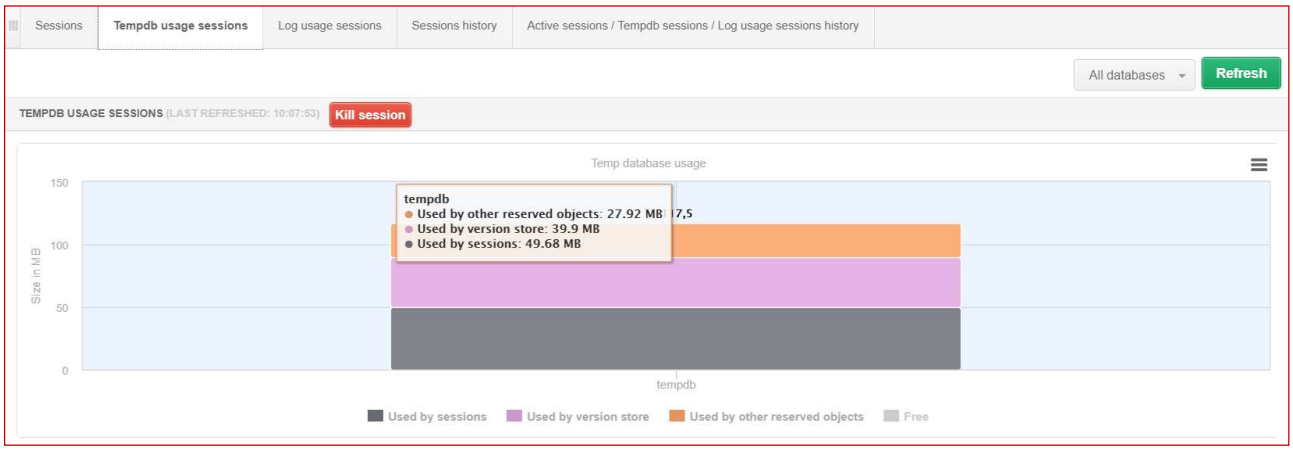
1.5.1 Searching for sessions for a given waits

The new version of the application adds the ability to search information about the user's session using a given type of wait. We start the search by pressing the "Hide additional filters" button and then from the list of available waits we add the ones we want to view.
After pressing the Refresh button, only those sessions that were waiting, for a wait selected by the user from the list will be presented in the given period.
At the same time, you can also select other filters, e.g. such as SID session ID or Query Hash value.



1.5.2 Information about sessions version store Usage

The new version of the application has been added information about the use of memory in the Tempdb database via query version store
This information is visible on the chart in the Tempdb usage sessions tab.



This information is useful when the "read_committed_snapshot with ON" parameter is enabled on the basis (the parameter is switched on so that "select" type queries do not block the query that make changes).

This setting generates additional entries in the Tempdb database because the change version is kept until the transaction is closed.

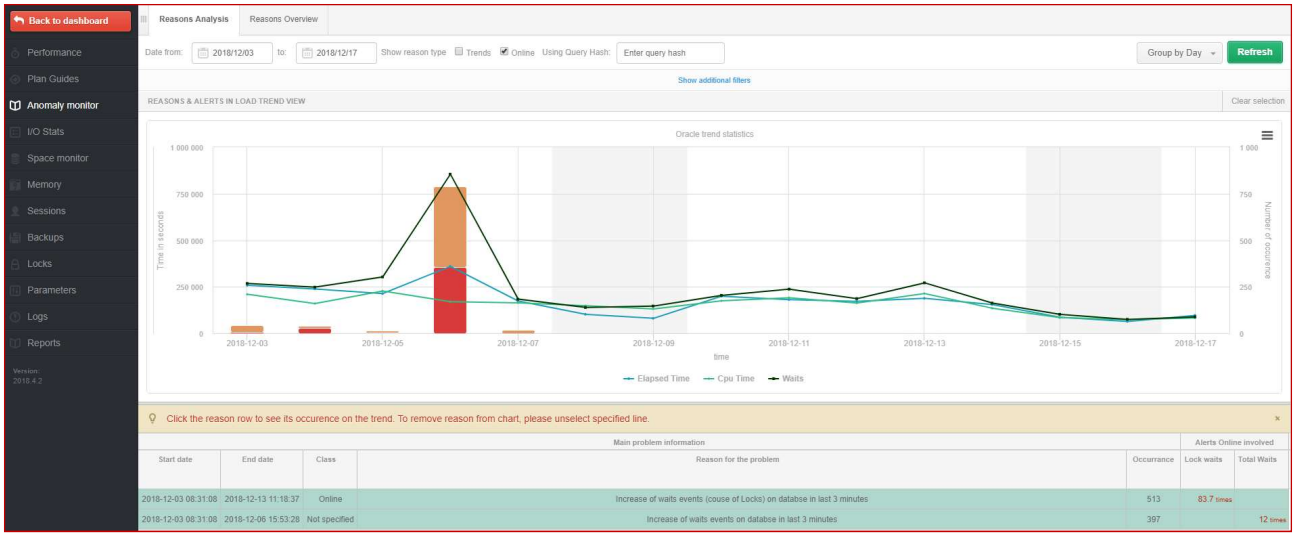
1.6 *Anomaly Monitor*

In the new version of the application, the functionality of viewing anomalies (alerts) has been added. The browser is available from the Instance Analysis> Anomaly Monitor SQL Instance details.

1.6.1 Problems viewer in the SQL Instance

On the page user can choose between two tabs: Reasons Analysis and Reasons Overview.

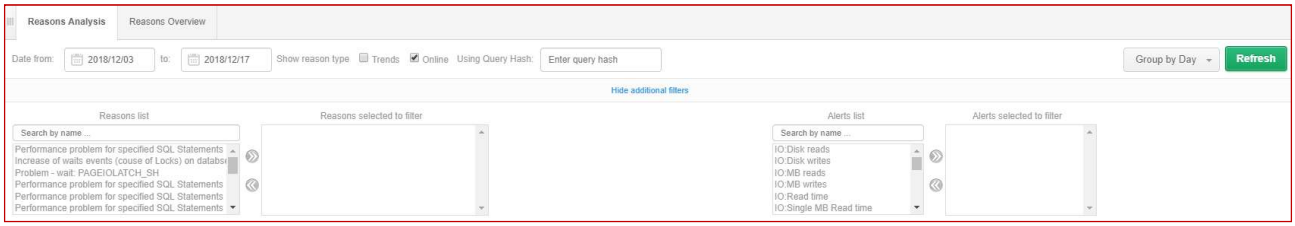
1.6.1.1 Reasons Analysis



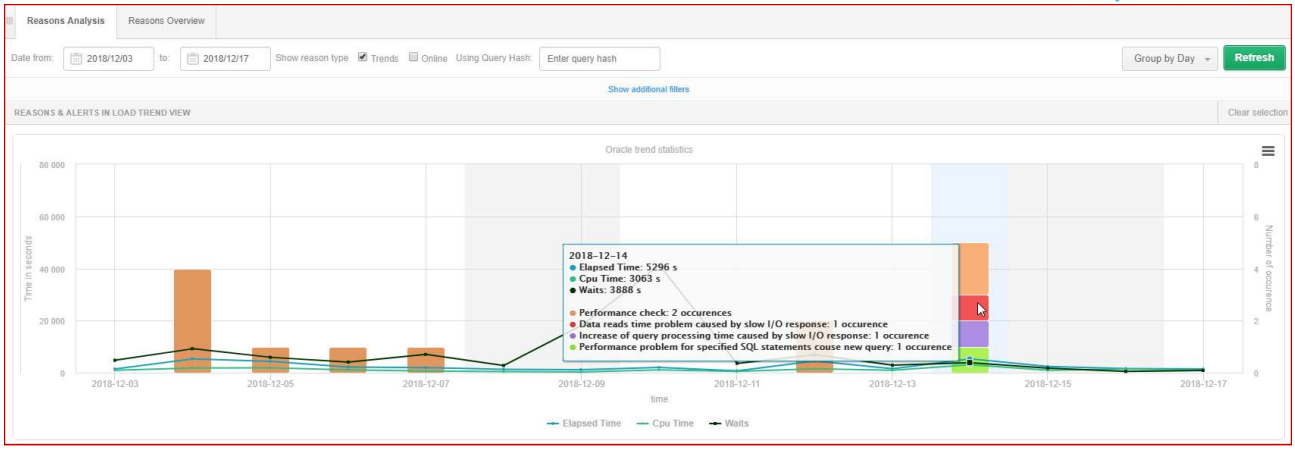
On the page you can choose several filters to help you find the problem you are looking for. As part of the filtering these options are available:

- choosing a date or range of dates,
- sorting after month, day, hour, snap,
- (Trends or Online) - the ability to indicate which type of alerts we want to view,
- Hash value - selection of alerts in which the indicated query identifier occurred,
- Reason list - the opportunity to indicate the dedicated causes of the problem,
- Alert lists - the ability to indicate dedicated alerts.

Screen of available filters on the Anomaly Monitor page:



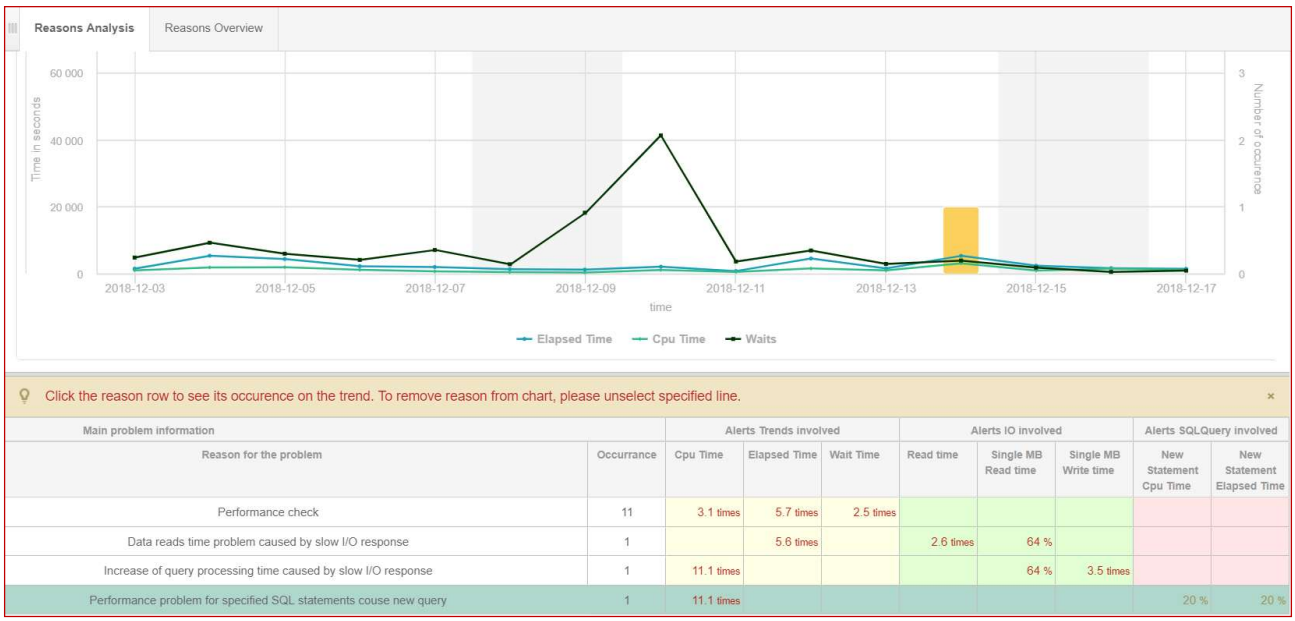
After configuring the appropriate filters, click the [Refresh] button. As a result, a graph will be presented in which, apart from the SQL Instance such as Elapsed Time, CPU Time o Waits, the number of occurrences of a given problem per unit of time will be presented in the form of bars in the graph. By indicating a given bar on the chart, a tooltip will be presented with information containing data on basic measures as well as the number of instances of a given problem per unit of time.



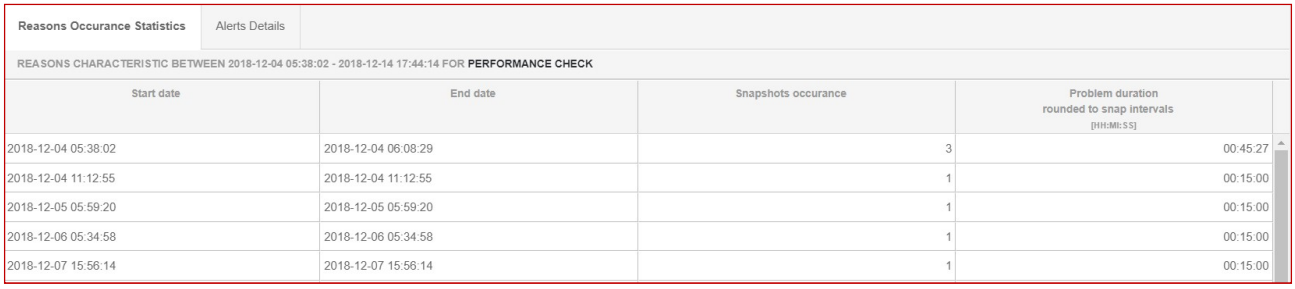
Below the graph is a table containing additional information about the causes of alerts presented in the graph. The table contains:

- **Start date / End date** - date range in which the given reason occurred,
- **Class -** the class / area to which the given reason was assigned,
- **Reason for the problem -** the cause of the problem,
- **Occurance** - the number of instances of a given cause in the selected date range
- **Alerts trends involved / IO involved / SQL Query involved -** sets of alerts included in the definition of a given problem cause.

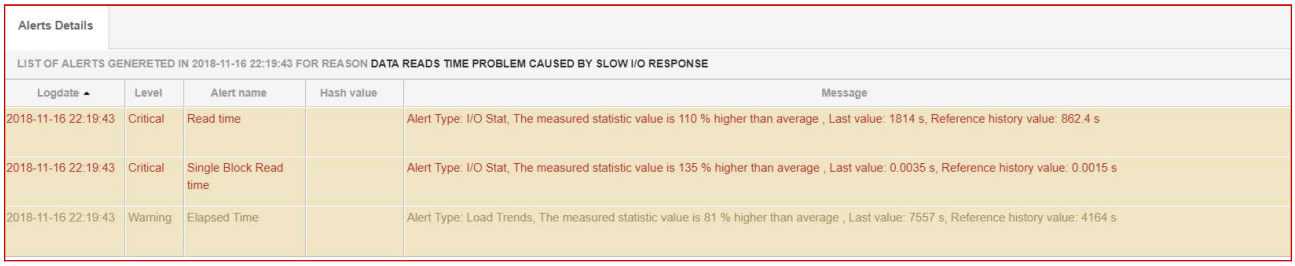
The user can indicate the reasons for the problem in the table. Each selection / uncheck will convert the data in the chart and present only the selected rows.



The data contained in the table are average data for all occurrences of a given reason. For a more in-depth analysis of a given problem, after selecting a row, the table will display additional detailed information in the **Alerts Details** tab. This view contains information on alerts that have exceeded the thresholds defined for the given cause of the problem.
There is also a view grouping alerts of the same type that occurred after each other. This allows you to verify how long the problem lasted. This information is available in the Reasons Occurance Statistics tab.



In the case below for one (Occurrence = 1) occurrence of the problem Data reads time problem caused by slow I / O response. Values for each of the alert defined for this problem which exceeded the threshold values were presented.



Alerts Details

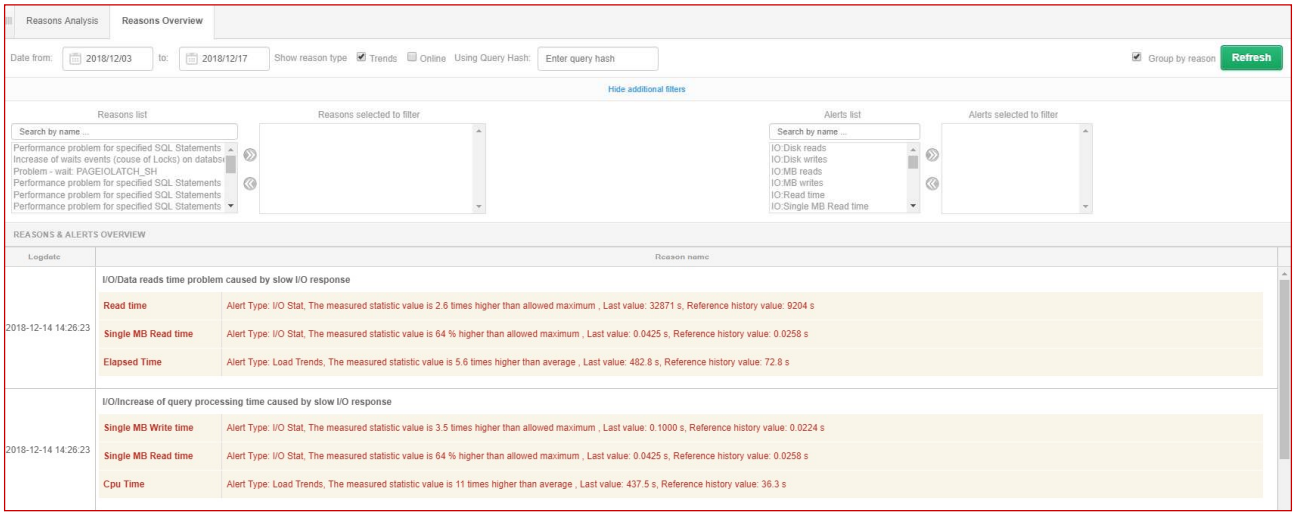
LIST OF ALERTS GENERETED IN 2018-11-16 22:19:43 FOR REASON **DATA READS TIME PROBLEM CAUSED BY SLOW I/O RESPONSE**

Logdate	Level	Alert name	Hash value	Message
2018-11-16 22:19:43	Critical	Read time		Alert Type: I/O Stat, The measured statistic value is 110 % higher than average , Last value: 1814 s, Reference history value: 862.4 s
2018-11-16 22:19:43	Critical	Single Block Read time		Alert Type: I/O Stat, The measured statistic value is 135 % higher than average , Last value: 0.0035 s, Reference history value: 0.0015 s
2018-11-16 22:19:43	Warning	Elapsed Time		Alert Type: Load Trends, The measured statistic value is 81 % higher than average , Last value: 7557 s, Reference history value: 4164 s

Note: information in the Alerts Details tab is only available for the last selected cause of the problem.

1.6.1.2 Reasons Overwiew

As part of this tab, the application allows you to view problems in one set. We can choose the same filters as for the Reasons Analysis tab and additionally the option of marking / deselecting grouping after the Cause.



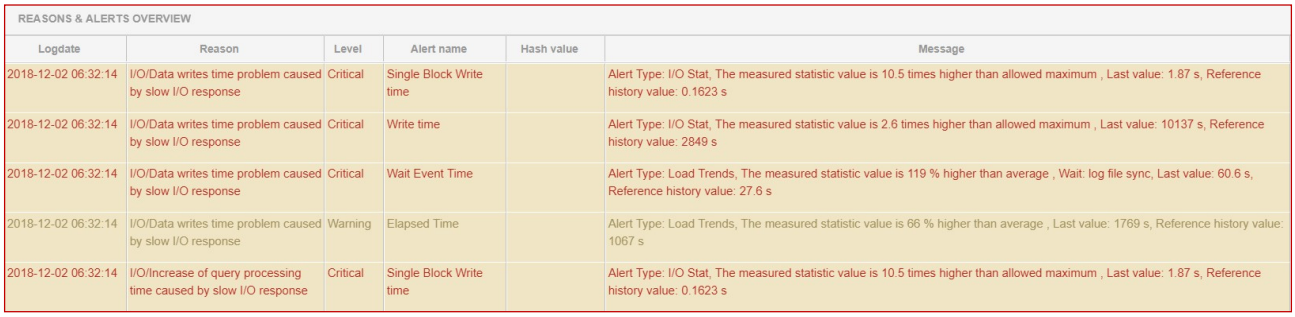
Depending on the checkbox **[Group by reason]**, alert data will be displayed in various lists:

- selected

REASONS & ALERTS OVERVIEW

Logdate	Reason name	
2018-12-02 06:32:14	**I/O/Data writes time problem caused by slow I/O response**	
	Single Block Write time	Alert Type: I/O Stat, The measured statistic value is 10.5 times higher than allowed maximum , Last value: 1.87 s, Reference history value: 0.1623 s
	Write time	Alert Type: I/O Stat, The measured statistic value is 2.6 times higher than allowed maximum , Last value: 10137 s, Reference history value: 2849 s
	Wait Event Time	Alert Type: Load Trends, The measured statistic value is 119 % higher than average , Wait: log file sync, Last value: 60.6 s, Reference history value: 27.6 s
	Elapsed Time	Alert Type: Load Trends, The measured statistic value is 66 % higher than average , Last value: 1769 s, Reference history value: 1067 s

- unselected

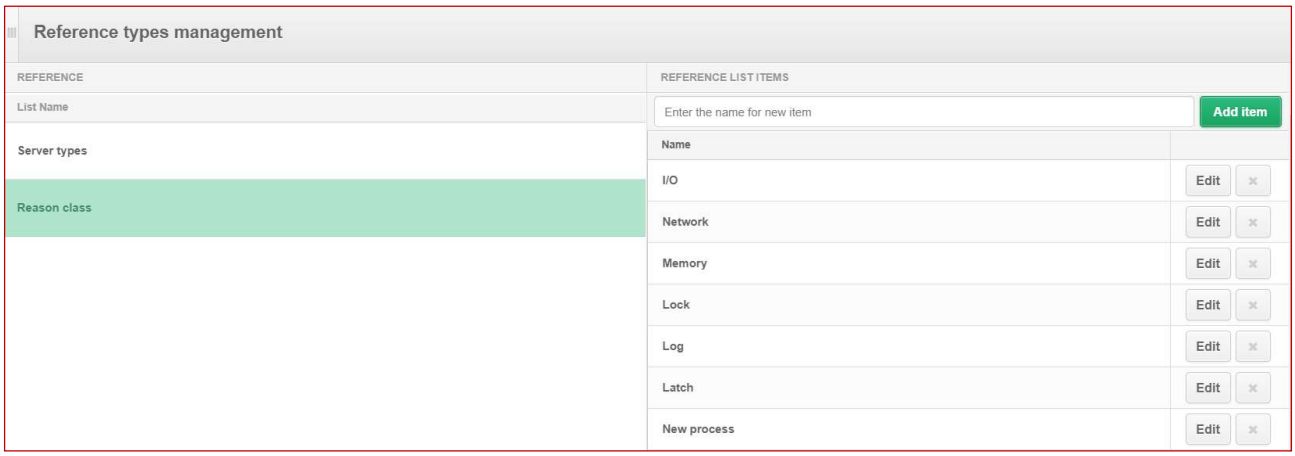


REASONS & ALERTS OVERVIEW

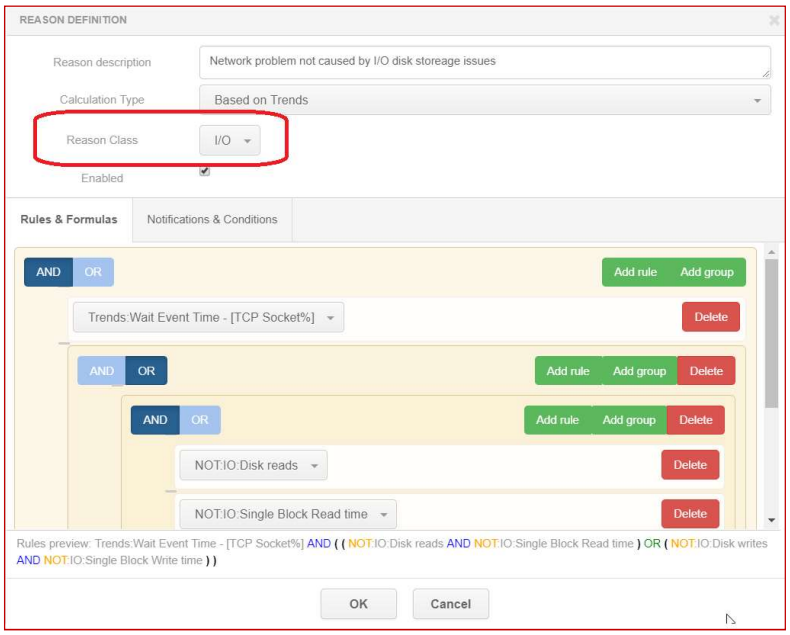
Logdate	Reason	Level	Alert name	Hash value	Message
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Critical	Single Block Write time		Alert Type: I/O Stat, The measured statistic value is 10.5 times higher than allowed maximum , Last value: 1.87 s, Reference history value: 0.1623 s
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Critical	Write time		Alert Type: I/O Stat, The measured statistic value is 2.6 times higher than allowed maximum , Last value: 10137 s, Reference history value: 2849 s
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Critical	Wait Event Time		Alert Type: Load Trends, The measured statistic value is 119 % higher than average , Wait: log file sync, Last value: 60.6 s, Reference history value: 27.6 s
2018-12-02 06:32:14	I/O/Data writes time problem caused by slow I/O response	Warning	Elapsed Time		Alert Type: Load Trends, The measured statistic value is 66 % higher than average , Last value: 1769 s, Reference history value: 1067 s
2018-12-02 06:32:14	I/O/Increase of query processing time caused by slow I/O response	Critical	Single Block Write time		Alert Type: I/O Stat, The measured statistic value is 10.5 times higher than allowed maximum , Last value: 1.87 s, Reference history value: 0.1623 s

1.6.2 Setting a class for a given cause of the problem

In the new version of the application, information about the class assigned to the cause of the problem has been added. Setting the class is nothing but an additional categorization of problems.
The problem class is defined in the dictionary table available in the menu Configuration> References lists> Reasons class.

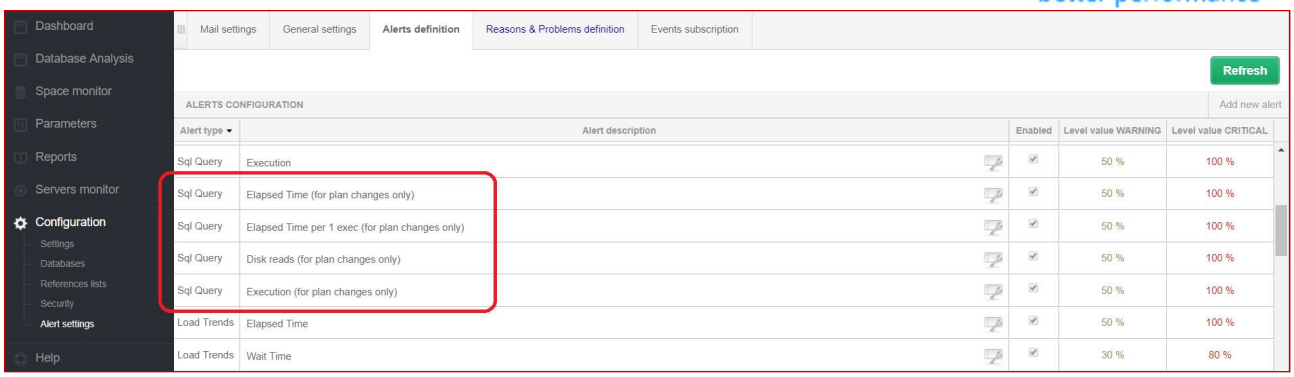


The class can be set by defining the new cause of the problem as well as modifying the existing one in the **Reasons & Problems definition** tab.

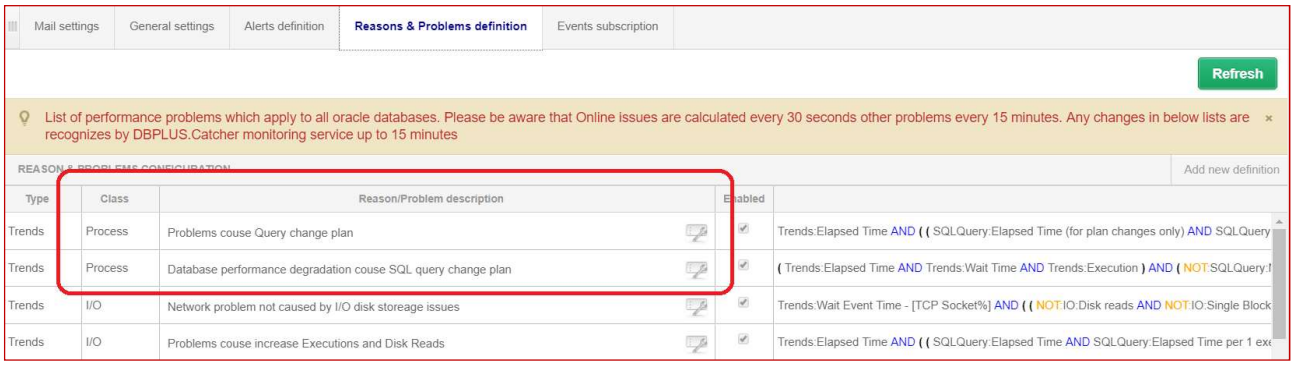


1.6.3 Change in verifying the change of the query explain plan

The new version has modified the functionality associated with verification whether the impact on exceeding the threshold set in the alert definition (applies only to alerts for queries - SQL Query) was influenced by the change of the query plan.
From now on the Alert definition for e.g. Elapsed Time with the change plan check option will be presented and configured separately than the Elapsed Time alert without this option selected.

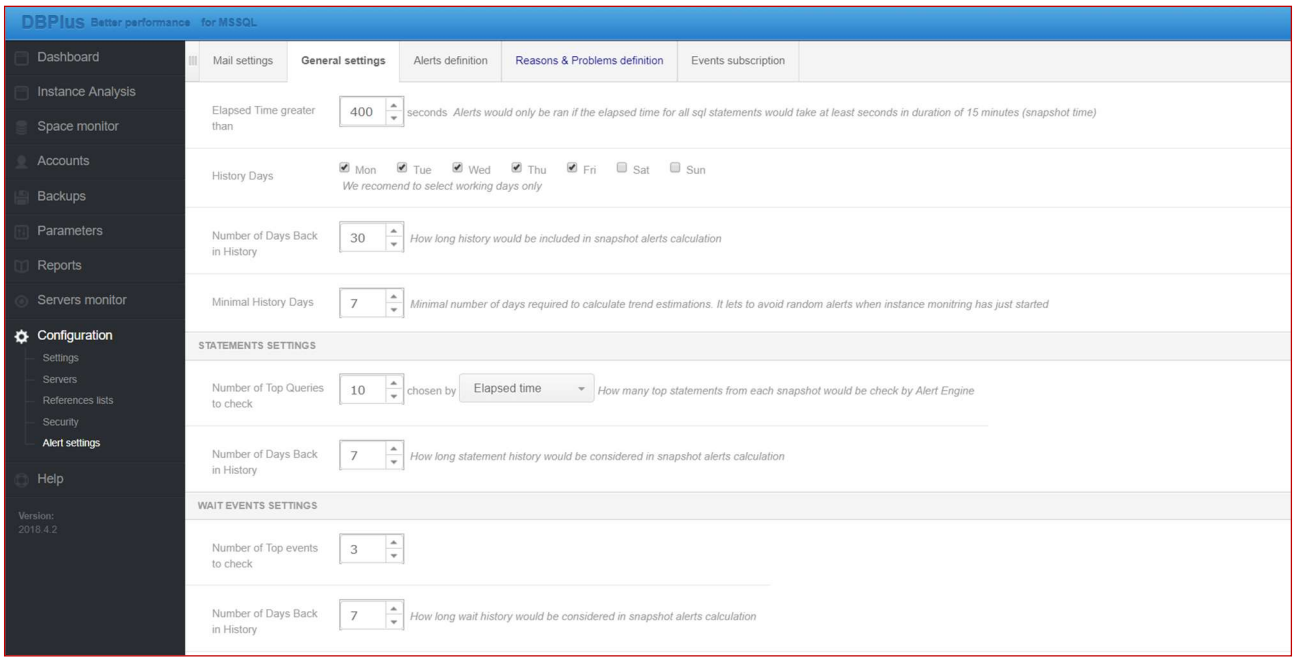


This change allows for more precisely defining problem definitions that cause the SQL Instance performance degradation.



1.6.3 New parameter controlling the alert function

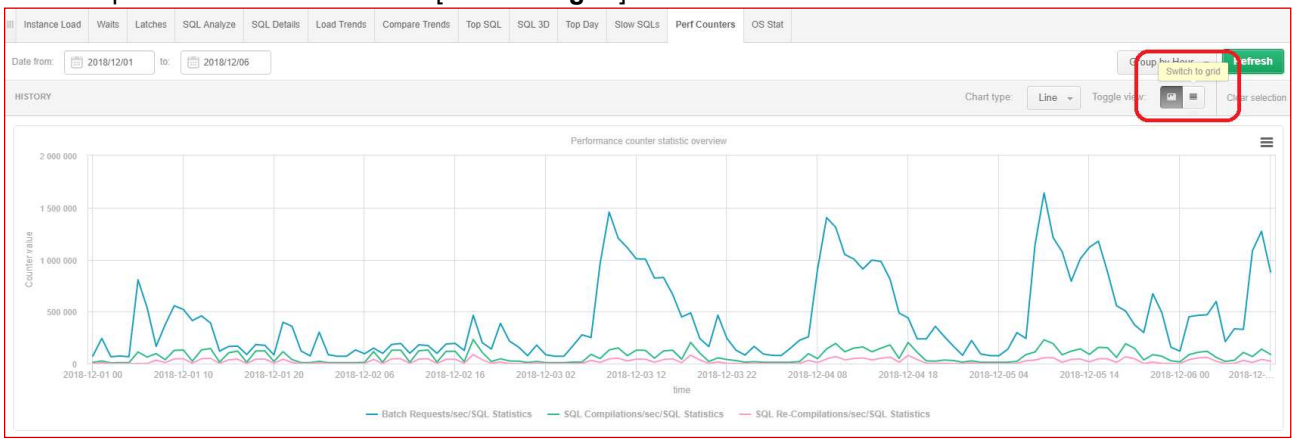
The new Minimal History Days parameter has been added for alert settings. The parameter is visible in the main menu Configuration> Alerts settings in the General settings tab. It mainly concerns new SQL Instance connected to monitoring. Specifies the minimum time after which trend-based alerts will be calculated.
The problem occurred after start monitoring new SQL Instance a large part of alerts was not able to correctly show the problem due to the lack of a "stable" trend. The parameter is modifiable and can be changed if necessary.



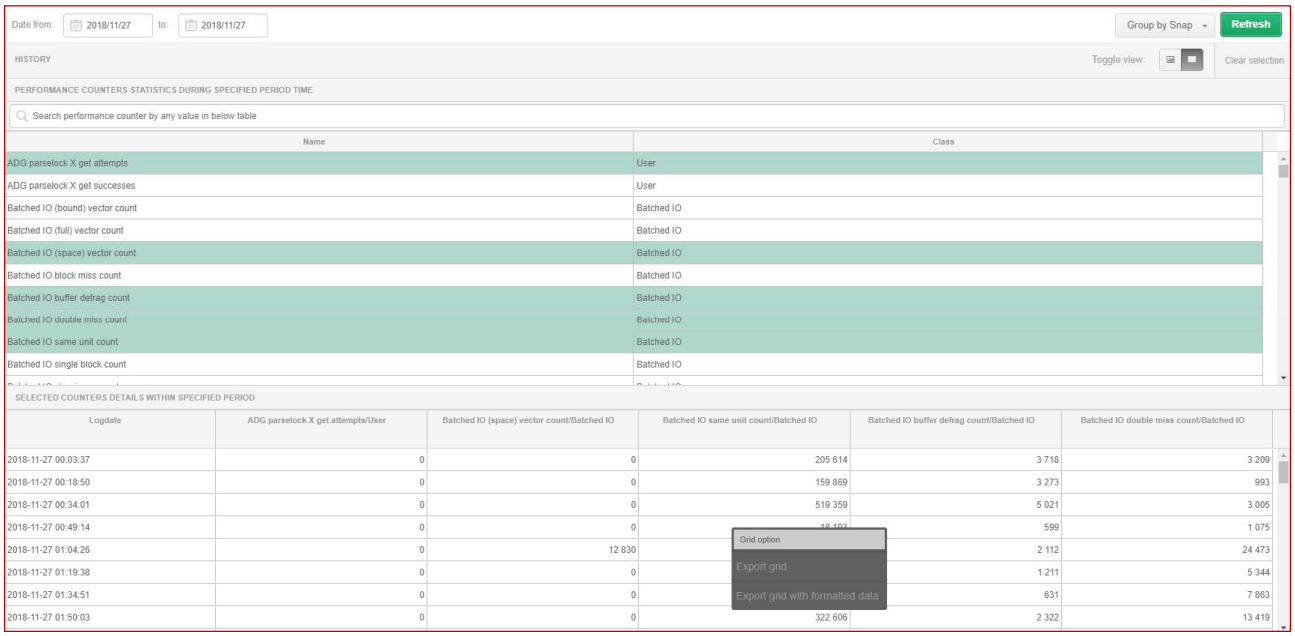
1.7 *General Improvements*

1.7.1 The ability to export Performance Counters statistics

In the new version of the application, the ability to export performance statistics has been added. The functionality is available from the Perf Counters tab under Instance Analysis. Export is possible by changing the chart preview to the tabular form [**Switch to grid**].



Export is performed for statistics previously selected from the table, choosing one from he following Grid options:
Export grid or Export grid with formatted data.



1.7.2 The ability to generate a Performance Report in hourly mode

In the new version of the application, we have made it possible to generate a Performance Report by providing specific hours for which the report should be generated. The report can be generated from the SQL Instance level in the Reports menu.