DBPLUS
better performance

DBPLUS
Performance Monitor for SQL Server
description of changes in version 2020.3

Date: October 9, 2020

Table of Contents

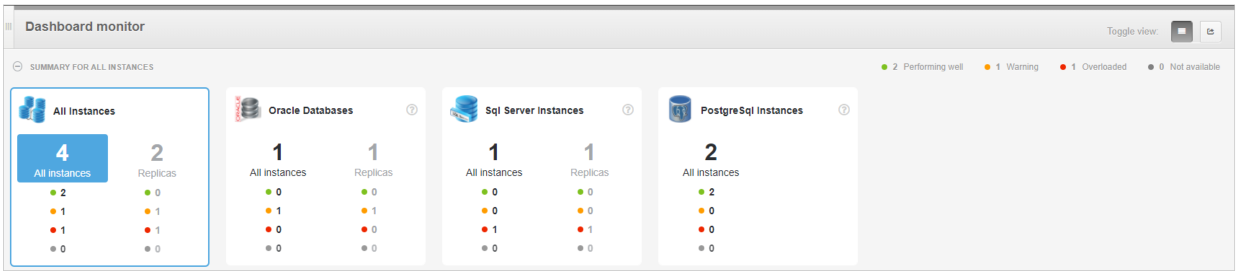
Below is a list of changes to the DBPLUS Performance Monitor system for monitoring Microsoft SQL Server instances.

New in 2020.3

1.1 *Shared Dashboard - next edition*

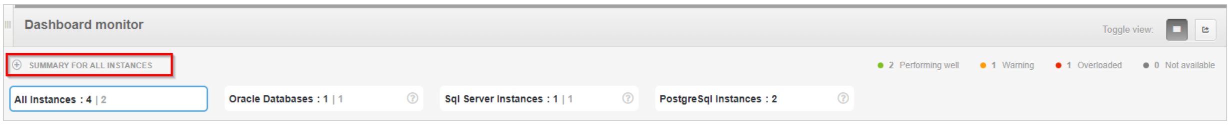
In version 2020.3.1, we improved the Dashboard screen. This screen allows, as before, an online view for all monitored databases for each of the three available platforms (SQL Server, Oracle, PostgreSQL). Information about status of monitored databases is available in the summary in two variants:

- Primary view:



This view allows you to filter the databases depending on the needs, by the current status, and can easily extract a group of databases for which the Always On AVAILABILITY GROUP is configured (by selecting the Replicas part).

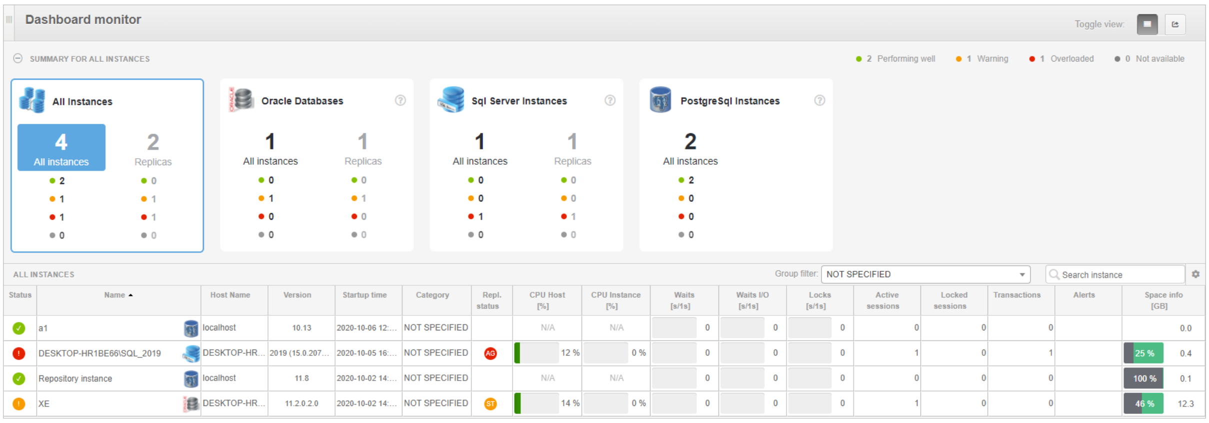
- Minimized view



This view is available after clicking the minus icon in the **SUMMARY FOR ALL INSTANCES** bar. It only contains information on the number of databases in a given and group, and allows easy navigation between them.

Additionally, in the latest version in the grid available for each SQL Instance, apart from the columns presenting basic data about the instance information is also available on:

- CPU Host – CPU utilization for the server where the instance is located,
- CPU Instance – CPU utilization of a given SQL instance
- Waits – total waits [s/1s]
- Waits I/O – I/O related waits [s/1s]
- Locks – level of locks on a given instance [s/1s]
- Active Sessions – number of active sessions
- Locked Sessions – blocked sessions level
- Transactions – the number of active transactions in the database
- Alerts – number of alerts (critical/warning)
- Space info – the level of used space for a given instance [GB]



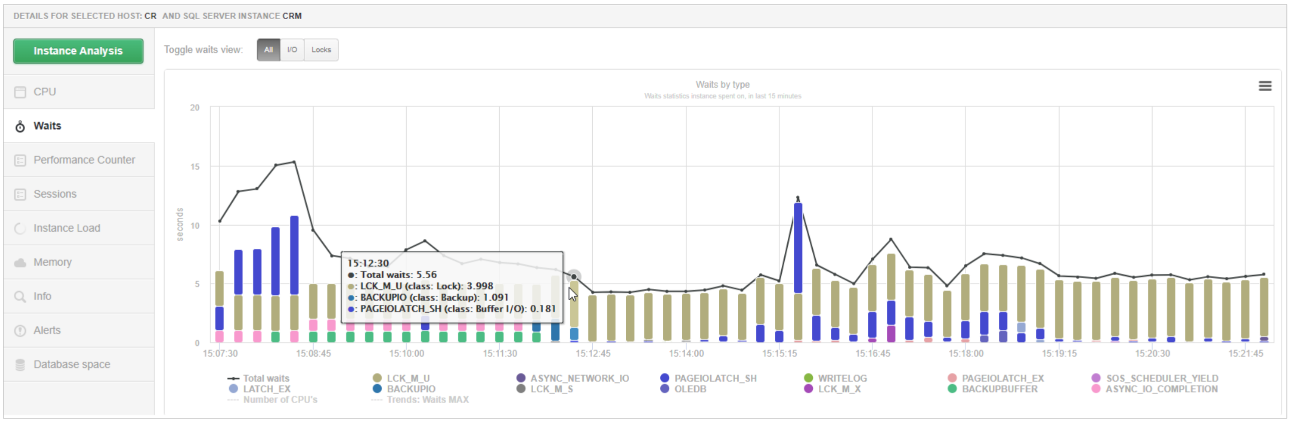
ALL INSTANCES

Group filter: NOT SPECIFIED

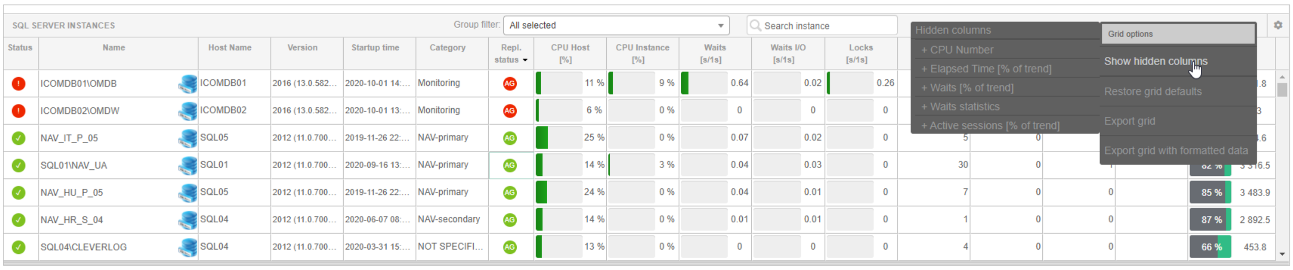
Search instance

Status	Name ▲	Host Name	Version	Startup time	Category	Repl. status	CPU Host [%]	CPU Instance [%]	Waits [s/1s]	Waits I/O [s/1s]	Locks [s/1s]	Active sessions	Locked sessions	Transactions	Alerts	Space info [GB]
✓	a1	localhost	10.13	2020-10-06 12:...	NOT SPECIFIED		N/A	N/A	0	0	0	0	0	0		0.0
!	DESKTOP-HR1BE66\SQL_2019	DESKTOP-HR...	2019 (15.0.207...	2020-10-05 16:...	NOT SPECIFIED	AG	12 %	0 %	0	0	0	1	0	1		25 % 0.4
✓	Repository instance	localhost	11.8	2020-10-02 14:...	NOT SPECIFIED		N/A	N/A	0	0	0	0	0	0		100 % 0.1
!	XE	DESKTOP-HR...	11.2.0.2.0	2020-10-02 14:...	NOT SPECIFIED	ST	14 %	0 %	0	0	0	1	0	0		46 % 12.3

By selecting a specific column for a given SQL instance, the User receives information on changes in statistics in the form of a graph for the last 15 minutes. In the case below, we get information on top waits for CRM instances.


DETAILS FOR SELECTED HOST: CR AND SQL SERVER INSTANCE CRM

The dashboard allows access to other columns that the User can freely add to the dashboard screen after selecting the **Show hidden columns** option in the settings of a given grid (cog icon).


SQL SERVER INSTANCES

Group filter: All selected

Search instance

Status	Name	Host Name	Version	Startup time	Category	Repl. status	CPU Host [%]	CPU Instance [%]	Waits [s/1s]	Waits I/O [s/1s]	Locks [s/1s]						
!	ICOMDB01\OMDB	ICOMDB01	2016 (13.0.582...	2020-10-01 14:...	Monitoring	AG	11 %	9 %	0.64	0.02	0.26						.8
!	ICOMDB02\OMDW	ICOMDB02	2016 (13.0.582...	2020-10-01 13:...	Monitoring	AG	6 %	0 %	0	0	0						3
✓	NAV_IT_P_05	SQL05	2012 (11.0.700...	2019-11-26 22:...	NAV-primary	AG	25 %	0 %	0.07	0.02	0	5	0				.6
✓	SQL01\NAV_UA	SQL01	2012 (11.0.700...	2020-09-16 13:...	NAV-primary	AG	14 %	3 %	0.04	0.03	0	30	0			82 %	3 316.5
✓	NAV_HU_P_05	SQL05	2012 (11.0.700...	2019-11-26 22:...	NAV-primary	AG	24 %	0 %	0.04	0.01	0	7	0	0		85 %	3 483.9
✓	NAV_HR_S_04	SQL04	2012 (11.0.700...	2020-06-07 08:...	NAV-secondary	AG	14 %	0 %	0.01	0.01	0	1	0	0		87 %	2 892.5
✓	SQL04\CLEVERLOG	SQL04	2012 (11.0.700...	2020-03-31 15:...	NOT SPECIFI...	AG	13 %	0 %	0	0	0	4	0	0		66 %	453.8

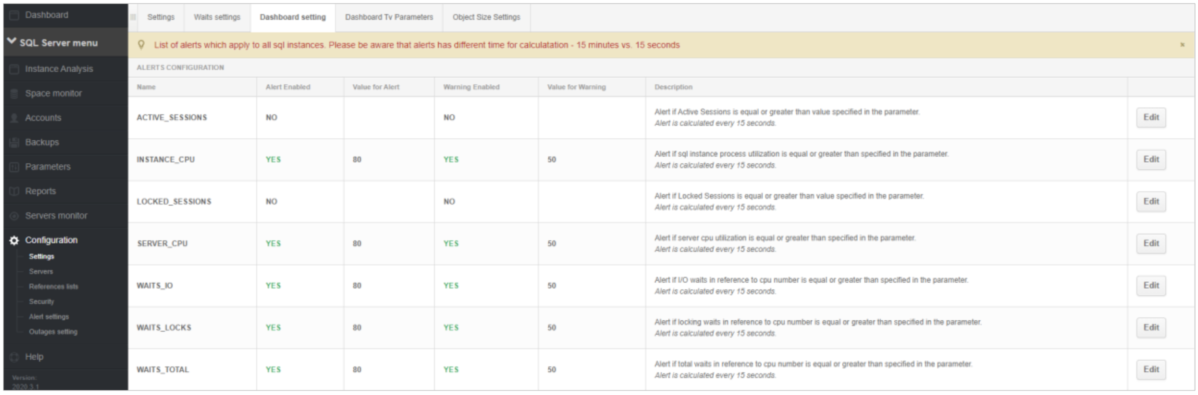
Changing the status of a monitored SQL instance

The change of the instance status is related to exceeding the alarm thresholds for the most important database performance parameters. In the event that the database status changes to orange (Warning) or red (Critical), the information about the cause is available after pointing the mouse at the given status.

The change in the instance status is related to exceeding the alarm thresholds for the most important performance parameters of SQL instances:

- Active_Sessions
- Instance_CPU
- Locked_Sessions
- Server_CPU
- Waits I/O
- Waits Locks
- Waits Total

The User can define his own alarm thresholds for each of the main database parameters. This configuration is available in the **Configuration> Settings** menu in the **Dashboard setting** tab. Exceeding a given parameter causes a change of status for a given instance depending on the exceeded threshold. Configuration is available for all or a dedicated SQL instance.



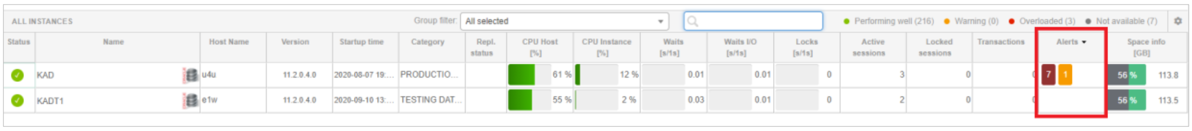
List of alerts which apply to all sql instances. Please be aware that alerts has different time for calculatation - 15 minutes vs. 15 seconds

ALERTS CONFIGURATION

Name	Alert Enabled	Value for Alert	Warning Enabled	Value for Warning	Description	
ACTIVE_SESSIONS	NO		NO		Alert if Active Sessions is equal or greater than value specified in the parameter. *Alert is calculated every 15 seconds.*	Edit
INSTANCE_CPU	YES	80	YES	50	Alert if sql instance process utilization is equal or greater than specified in the parameter. *Alert is calculated every 15 seconds.*	Edit
LOCKED_SESSIONS	NO		NO		Alert if Locked Sessions is equal or greater than value specified in the parameter. *Alert is calculated every 15 seconds.*	Edit
SERVER_CPU	YES	80	YES	50	Alert if server cpu utilization is equal or greater than specified in the parameter. *Alert is calculated every 15 seconds.*	Edit
WAITS_IO	YES	80	YES	50	Alert if I/O waits in reference to cpu number is equal or greater than specified in the parameter. *Alert is calculated every 15 seconds.*	Edit
WAITS_LOCKS	YES	80	YES	50	Alert if locking waits in reference to cpu number is equal or greater than specified in the parameter. *Alert is calculated every 15 seconds.*	Edit
WAITS_TOTAL	YES	80	YES	50	Alert if total waits in reference to cpu number is equal or greater than specified in the parameter. *Alert is calculated every 15 seconds.*	Edit

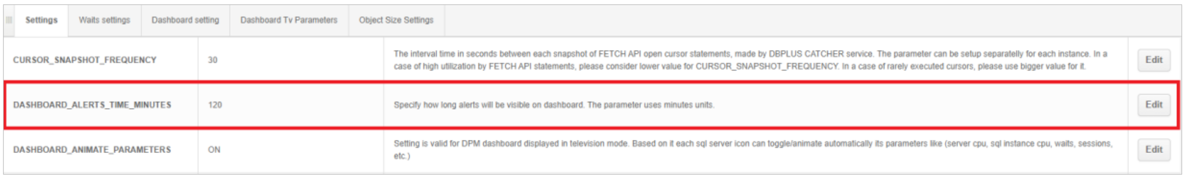
Alerts on the Dashboard screen

In the latest version, Alerts are displayed on the Dashboard screen. Depending on exceeding the alarm threshold, the user receives information about the criticality of the alert and the number of occurrences. Alerts are visible on the screen for a period of 2 hours after their occurrence, for example, if the alert occurred at 09:20, the information on the Dashboard about the alert will be visible until 11:20.



Status	Name	Host Name	Version	Startup time	Category	Repl. status	CPU Host [%]	CPU Instance [%]	Waits [s/1s]	Waits I/O [s/1s]	Locks [s/1s]	Active sessions	Locked sessions	Transactions	Alerts	Space info [GB]
✓	KAD	u4u	11.2.0.4.0	2020-08-07 19:...	PRODUCTIO...		61 %	12 %	0.01	0.01	0	3	0	0	7 1	56 % 113.8
✓	KADT1	e1w	11.2.0.4.0	2020-09-10 13:...	TESTING DAT...		55 %	2 %	0.03	0.01	0	2	0	0		56 % 113.5

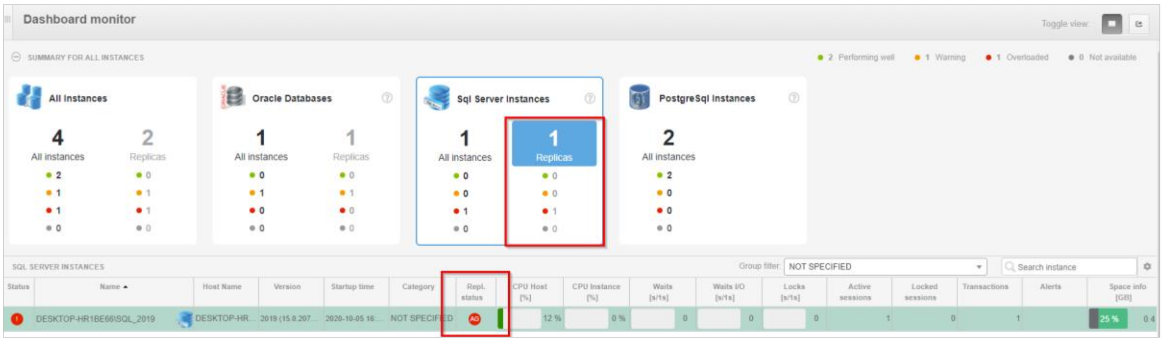
The visibility time of alerts on the Dashboard screen can be configured separately for each platform, using the **DASHBOARD_ALERTS_TIME_MINUTES** parameter available in the **Configuration> Settings** menu.



CURSOR_SNAPSHOT_FREQUENCY	30	The interval time in seconds between each snapshot of FETCH API open cursor statements, made by DBPLUS CATCHER service. The parameter can be setup separatelly for each instance. In a case of high utilization by FETCH API statements, please consider lower value for CURSOR_SNAPSHOT_FREQUENCY. In a case of rarely executed cursors, please use bigger value for it.	Edit
DASHBOARD_ALERTS_TIME_MINUTES	120	Specify how long alerts will be visible on dashboard. The parameter uses minutes units.	Edit
DASHBOARD_ANIMATE_PARAMETERS	ON	Setting is valid for DPM dashboard displayed in television mode. Based on it each sql server icon can toggle/animate automatically its parameters like (server cpu, sql instance cpu, waits, sessions, etc.)	Edit

1.1.1 Monitoring AlwaysOn Availability Group

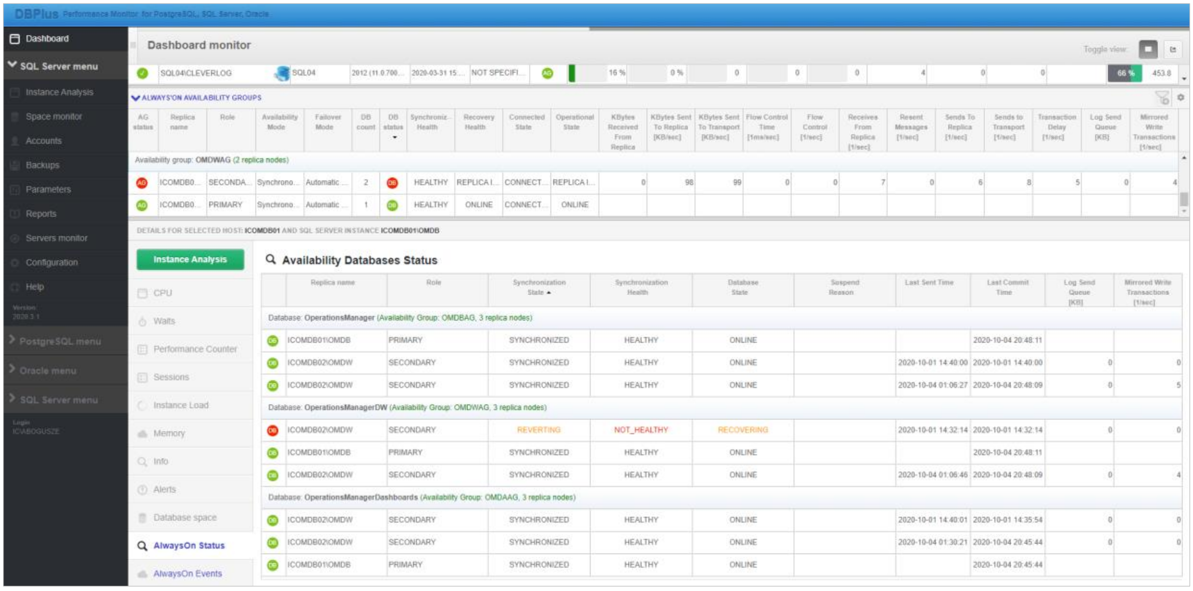
In the latest version of Dashbord, monitoring of Always On groups has been added. For this purpose, at the main level, we have added a new **Replicas Status** column, which represents the general status of Always On groups available in each SQL instance. The Always On status affects the overall status presented for a given SQL instance. If the Always On group doesn't work properly, the status of the entire SQL instance changes to red.



All Instances		Oracle Databases		Sql Server Instances		PostgreSql Instances
4 All instances	2 Replicas	1 All instances	1 Replicas	1 All instances	1 Replicas	2 All instances
● 2	● 0	● 0	● 0	● 0	● 0	● 2
● 1	● 1	● 1	● 1	● 0	● 0	● 0
● 1	● 1	● 0	● 0	● 1	● 1	● 0
● 0	● 0	● 0	● 0	● 0	● 0	● 0

Status	Name ▲	Host Name	Version	Startup time	Category	Repl. status	CPU Host [%]	CPU Instance [%]	Waits [s/1s]	Waits I/O [s/1s]	Locks [s/1s]	Active sessions	Locked sessions	Transactions	Alerts	Space info [GB]
!	DESKTOP-HR1BE66\SQL_2019	DESKTOP-HR...	2019 (15.0.207...	2020-10-05 16:...	NOT SPECIFIED	AG	12 %	0 %	0	0	0	1	0	1		25 % 0.4

The user can verify the cause of the AG group that doesn't work properly. Click on the icon AG for further analysis. Clicking opens an additional ALWAYS ON AVAILABILITY GROUP section in the grid below. This grid contains information on all groups associated with a given SQL instance.



Dashboard monitor

SQL04\CLEVERLOG | SQL04 | 2012 (11.0.700... | 2020-03-31 15:... | NOT SPECIFI... | AG | 16 % | 0 % | 0 | 0 | 0 | 4 | 0 | 0 | 66 % | 453.8

ALWAYS'ON AVAILABILITY GROUPS

AG status	Replica name	Role	Availability Mode	Failover Mode	DB count	DB status	Synchroniz... Health	Recovery Health	Connected State	Operational State	KBytes Received From Replica	KBytes Sent To Replica [KB/sec]	KBytes Sent To Transport [KB/sec]	Flow Control Time [1ms/sec]	Flow Control [1/sec]	Receives From Replica [1/sec]	Resent Messages [1/sec]	Sends To Replica [1/sec]	Sends to Transport [1/sec]	Transaction Delay [1/sec]	Log Send Queue [KB]	Mirrored Write Transactions [1/sec]
Availability group: OMDWAG (2 replica nodes)																						
AG	ICOMDB0...	SECONDA...	Synchrono...	Automatic ...	2	DB	HEALTHY	REPLICA I...	CONNECT...	REPLICA I...	0	98	99	0	0	7	0	6	8	5	0	4
AG	ICOMDB0...	PRIMARY	Synchrono...	Automatic ...	1	DB	HEALTHY	ONLINE	CONNECT...	ONLINE												

DETAILS FOR SELECTED HOST: **ICOMDB01** AND SQL SERVER INSTANCE **ICOMDB01\OMDB**

Availability Databases Status

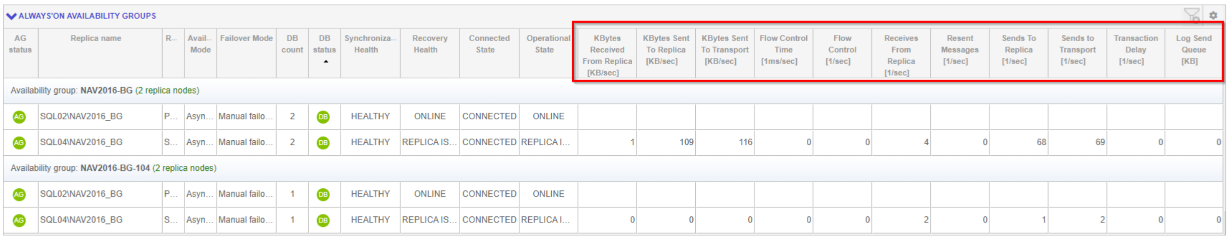
	Replica name	Role	Synchronization State	Synchronization Health	Database State	Suspend Reason	Last Sent Time	Last Commit Time	Log Send Queue [KB]	Mirrored Write Transactions [1/sec]
Database: OperationsManager (Availability Group: OMDBAG, 3 replica nodes)										
DB	ICOMDB01\OMDB	PRIMARY	SYNCHRONIZED	HEALTHY	ONLINE			2020-10-04 20:48:11		
DB	ICOMDB02\OMDW	SECONDARY	SYNCHRONIZED	HEALTHY	ONLINE		2020-10-01 14:40:00	2020-10-01 14:40:00	0	0
DB	ICOMDB02\OMDW	SECONDARY	SYNCHRONIZED	HEALTHY	ONLINE		2020-10-04 01:06:27	2020-10-04 20:48:09	0	5
Database: OperationsManagerDW (Availability Group: OMDWAG, 3 replica nodes)										
DB	ICOMDB02\OMDW	SECONDARY	REVERTING	NOT_HEALTHY	RECOVERING		2020-10-01 14:32:14	2020-10-01 14:32:14	0	0
DB	ICOMDB01\OMDB	PRIMARY	SYNCHRONIZED	HEALTHY	ONLINE			2020-10-04 20:48:11		
DB	ICOMDB02\OMDW	SECONDARY	SYNCHRONIZED	HEALTHY	ONLINE		2020-10-04 01:06:46	2020-10-04 20:48:09	0	4
Database: OperationsManagerDashboards (Availability Group: OMDAAG, 3 replica nodes)										
DB	ICOMDB02\OMDW	SECONDARY	SYNCHRONIZED	HEALTHY	ONLINE		2020-10-01 14:40:01	2020-10-01 14:35:54	0	0
DB	ICOMDB02\OMDW	SECONDARY	SYNCHRONIZED	HEALTHY	ONLINE		2020-10-04 01:30:21	2020-10-04 20:45:44	0	0
DB	ICOMDB01\OMDB	PRIMARY	SYNCHRONIZED	HEALTHY	ONLINE			2020-10-04 20:45:44		

At the grid level, the general status of each AG group is presented along with the most important parameters, which prove the efficiency of the AG mechanism. This information is updated each time the Dashboard screen is refreshed (every 15 seconds).
In this case, the User can make a choice:

- clicking on the header of a given group - receives information on all databases in the AG group (role PRIMARY and SECONDARY),
- by clicking on the AG group level, depending on the selected role, the User receives information about all databases available in each group for the indicated role (PRIMARY or SECONDARY).

Additionally, for each SQL instance using the Always On mechanism, information about events related to AlwaysOn replication is available in the **Always On Events** tab. These data are available in tabular form.

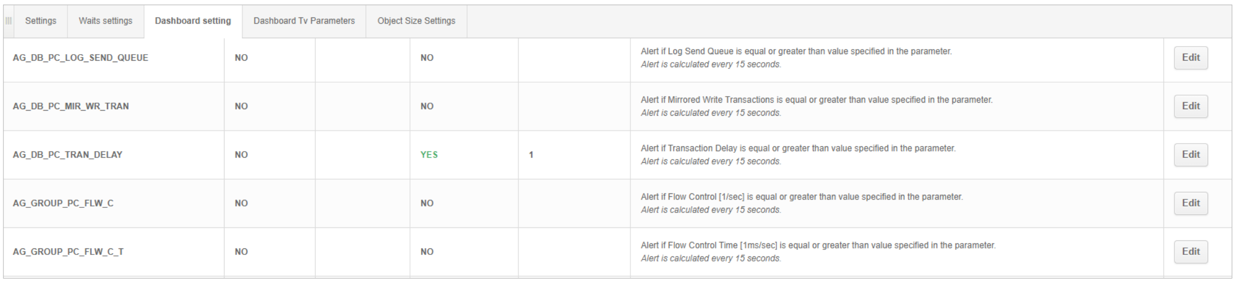
Changing the status of the Always On group depends primarily on the status available in the system views. The user can define alert thresholds for each of the parameters available in the group, beyond which information will be displayed on the dashboard screen.



ALWAYS'ON AVAILABILITY GROUPS

AG status	Replica name	R...	Avail... Mode	Failover Mode	DB count	DB status	Synchroniza... Health	Recovery Health	Connected State	Operational State	KBytes Received From Replica [KB/sec]	KBytes Sent To Replica [KB/sec]	KBytes Sent To Transport [KB/sec]	Flow Control Time [1ms/sec]	Flow Control [1/sec]	Receives From Replica [1/sec]	Resent Messages [1/sec]	Sends To Replica [1/sec]	Sends to Transport [1/sec]	Transaction Delay [1/sec]	Log Send Queue [KB]
Availability group: NAV2016-BG (2 replica nodes)																					
AG	SQL02\NAV2016_BG	P...	Asyn...	Manual failo...	2	DB	HEALTHY	ONLINE	CONNECTED	ONLINE											
AG	SQL04\NAV2016_BG	S...	Asyn...	Manual failo...	2	DB	HEALTHY	REPLICA IS...	CONNECTED	REPLICA I...	1	109	116	0	0	4	0	68	69	0	0
Availability group: NAV2016-BG-104 (2 replica nodes)																					
AG	SQL02\NAV2016_BG	P...	Asyn...	Manual failo...	1	DB	HEALTHY	ONLINE	CONNECTED	ONLINE											
AG	SQL04\NAV2016_BG	S...	Asyn...	Manual failo...	1	DB	HEALTHY	REPLICA IS...	CONNECTED	REPLICA I...	0	0	0	0	0	2	0	1	2	0	0

To change the thresholds, go to the settings available in the Configuration> Settings> Dashboard Settings menu.



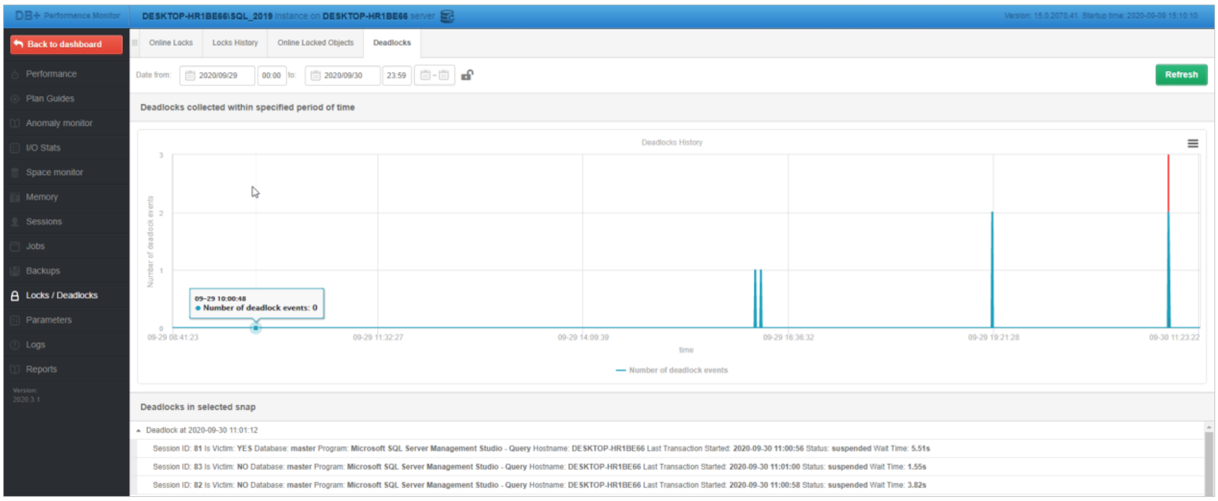
AG_DB_PC_LOG_SEND_QUEUE	NO		NO		Alert if Log Send Queue is equal or greater than value specified in the parameter. *Alert is calculated every 15 seconds.*	Edit
AG_DB_PC_MIR_WR_TRAN	NO		NO		Alert if Mirrored Write Transactions is equal or greater than value specified in the parameter. *Alert is calculated every 15 seconds.*	Edit
AG_DB_PC_TRAN_DELAY	NO		YES	1	Alert if Transaction Delay is equal or greater than value specified in the parameter. *Alert is calculated every 15 seconds.*	Edit
AG_GROUP_PC_FLW_C	NO		NO		Alert if Flow Control [1/sec] is equal or greater than value specified in the parameter. *Alert is calculated every 15 seconds.*	Edit
AG_GROUP_PC_FLW_C_T	NO		NO		Alert if Flow Control Time [1ms/sec] is equal or greater than value specified in the parameter. *Alert is calculated every 15 seconds.*	Edit

1.2 *Monitoring Deadlocks*

We have added deadlock event monitoring in the latest version of the application. Information is available at the database detail level in the **Locks / Deadlocks** menu in the **Deadlocks** tab.

After entering the tab, User can see the chart for the current day with deadlock events. By click the point on the chart where the blockage occurred, User gets additional information about the sessions that participated in the event.

In the presented example, the Deadlock event that took place on September 30, 2020 at 11:01:12 in which 3 sessions participated is visible. One of them was indicated by the database as a victim, whose operation was interrupted by the SQL Server database mechanism.



Deadlock at 2020-09-30 11:01:12

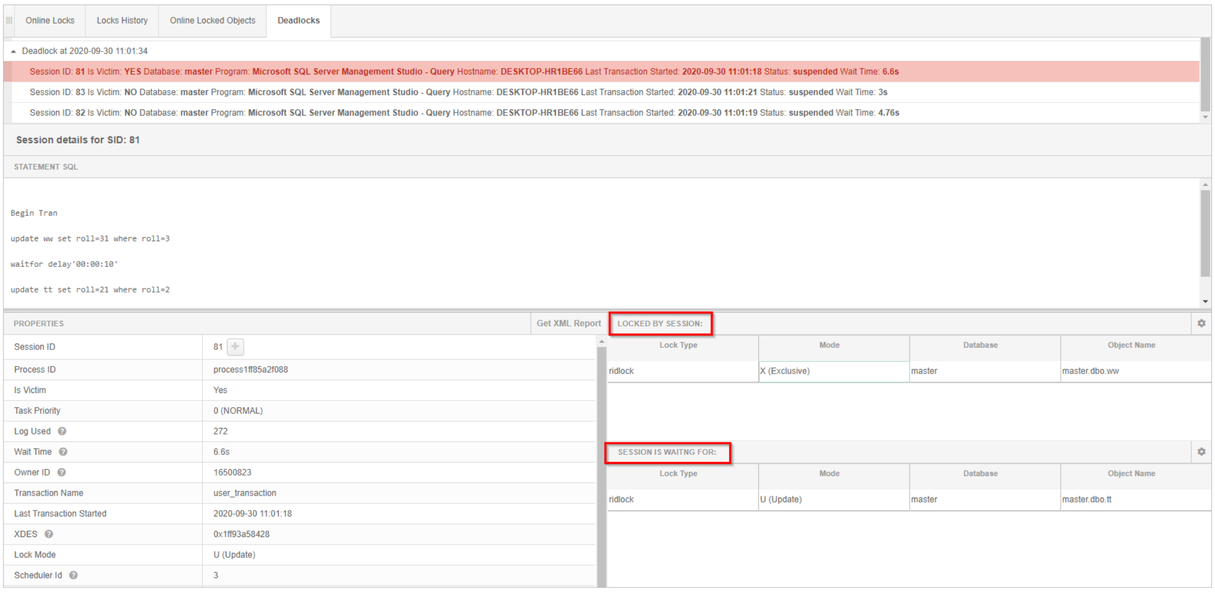
Session ID: **81** Is Victim: **YES** Database: **master** Program: **Microsoft SQL Server Management Studio - Query** Hostname: **DESKTOP-HR1BE66** Last Transaction Started: **2020-09-30 11:00:56** Status: **suspended** Wait Time: **5.51s**

Session ID: **83** Is Victim: **NO** Database: **master** Program: **Microsoft SQL Server Management Studio - Query** Hostname: **DESKTOP-HR1BE66** Last Transaction Started: **2020-09-30 11:01:00** Status: **suspended** Wait Time: **1.55s**

Session ID: **82** Is Victim: **NO** Database: **master** Program: **Microsoft SQL Server Management Studio - Query** Hostname: **DESKTOP-HR1BE66** Last Transaction Started: **2020-09-30 11:00:58** Status: **suspended** Wait Time: **3.82s**

By indicating the session in the deadlock block, the User obtains additional information about the event. The application provides, among others information about the query executed by the session when deadlock occurred as well as details about the event itself.

On the right, information about objects locked by a given session (Locked by seesion) and objects whose modification has been blocked by deadlock is presented.



Deadlock at 2020-09-30 11:01:34

Session ID: **81** Is Victim: **YES** Database: **master** Program: **Microsoft SQL Server Management Studio - Query** Hostname: **DESKTOP-HR1BE66** Last Transaction Started: **2020-09-30 11:01:18** Status: **suspended** Wait Time: **6.6s**

Session ID: **83** Is Victim: **NO** Database: **master** Program: **Microsoft SQL Server Management Studio - Query** Hostname: **DESKTOP-HR1BE66** Last Transaction Started: **2020-09-30 11:01:21** Status: **suspended** Wait Time: **3s**

Session ID: **82** Is Victim: **NO** Database: **master** Program: **Microsoft SQL Server Management Studio - Query** Hostname: **DESKTOP-HR1BE66** Last Transaction Started: **2020-09-30 11:01:19** Status: **suspended** Wait Time: **4.76s**

Session details for SID: 81

STATEMENT SQL

```
Begin Tran

update ww set roll=31 where roll=3

waitfor delay'00:00:10'

update tt set roll=21 where roll=2
```

PROPERTIES	Get XML Report
Session ID	81
Process ID	process1ff85a2f088
Is Victim	Yes
Task Priority	0 (NORMAL)
Log Used	272
Wait Time	6.6s
Owner ID	16500823
Transaction Name	user_transaction
Last Transaction Started	2020-09-30 11:01:18
XDES	0x1ff93a58428
Lock Mode	U (Update)
Scheduler Id	3

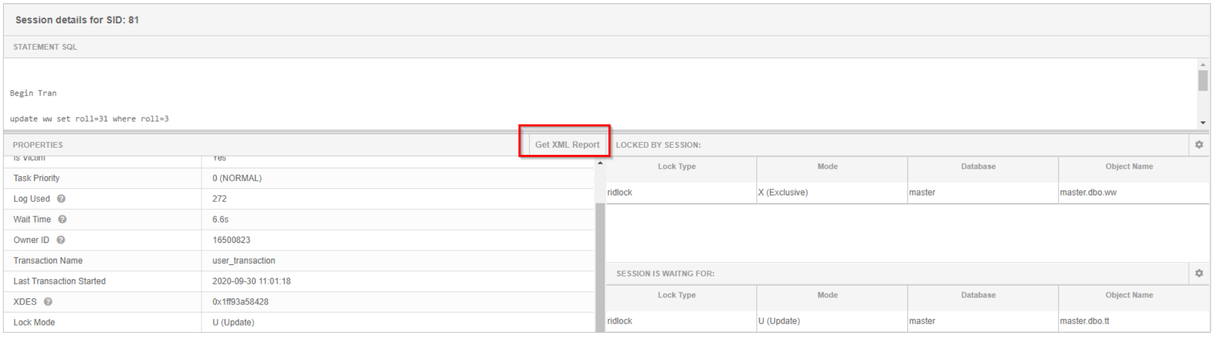
LOCKED BY SESSION:

Lock Type	Mode	Database	Object Name
ridlock	X (Exclusive)	master	master.dbo.ww

SESSION IS WAITNG FOR:

Lock Type	Mode	Database	Object Name
ridlock	U (Update)	master	master.dbo.tt

The user also gets access to the XML report generated by the SQL instances about the event itself. The report is available after clicking the [Get XML report]



Session details for SID: 81

STATEMENT SQL

```
Begin Tran

update ww set roll=31 where roll=3
```

PROPERTIES — Get XML Report

Is Victim	Yes
Task Priority	0 (NORMAL)
Log Used	272
Wait Time	6.6s
Owner ID	16500823
Transaction Name	user_transaction
Last Transaction Started	2020-09-30 11:01:18
XDES	0x1ff93a58428
Lock Mode	U (Update)

LOCKED BY SESSION:

Lock Type	Mode	Database	Object Name
ridlock	X (Exclusive)	master	master.dbo.ww

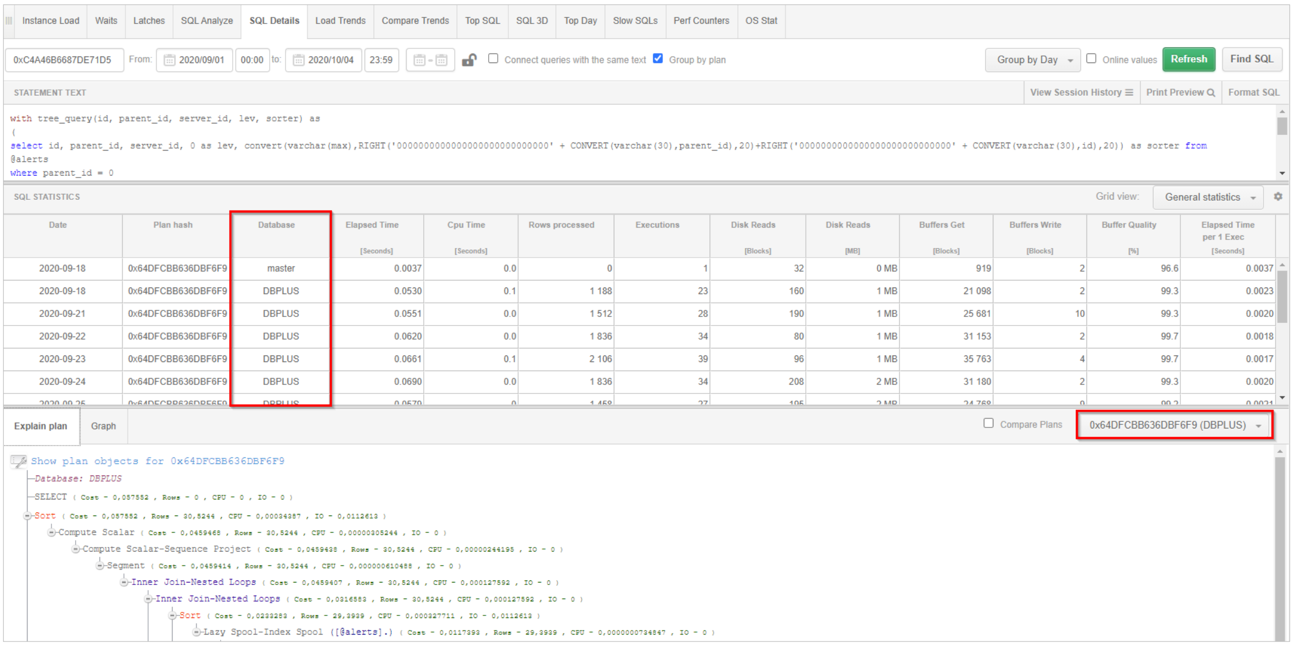
SESSION IS WAITNG FOR:

Lock Type	Mode	Database	Object Name
ridlock	U (Update)	master	master.dbo.tt

Information about the occurrence of a deadlock event is retrieved using the **Extended Events** mechanism. The user has the option to disable the retrieval of this information by changing the **MONITOR_XEVENT_DEADLOCKS** parameter in **the Configuration> Settings** menu. The parameter is enabled by default.

1.3 *Query statistics for databases*

In the latest version of the application, we have added the ability to view detailed query statistics (SQL Details tab) with a division into the database where the query is performed.
From this version, an additional Database column will be visible for the query details in the grid. The statistics presented in the grid will be additionally presented with a breakdown by database. If User need to view the data without dividing it into a database, deselect the **Group by plan** option.



0xC4A46B6687DE71D5 From: 2020/09/01 00:00 to: 2020/10/04 23:59 □ Connect queries with the same text ☑ Group by plan — Group by Day □ Online values — Refresh — Find SQL

STATEMENT TEXT — View Session History — Print Preview — Format SQL

```
with tree_query(id, parent_id, server_id, lev, sorter) as
(
select id, parent_id, server_id, 0 as lev, convert(varchar(max),RIGHT('000000000000000000000000000' + CONVERT(varchar(30),parent_id),20)+RIGHT('000000000000000000000000000' + CONVERT(varchar(30),id),20)) as sorter from
@alerts
where parent_id = 0
```

SQL STATISTICS — Grid view: General statistics

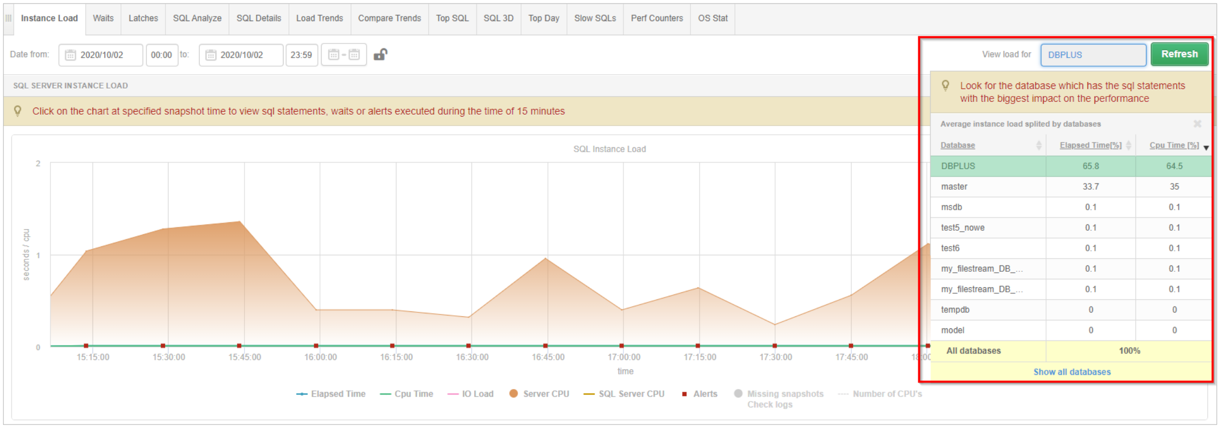
Date	Plan hash	Database	Elapsed Time [Seconds]	Cpu Time [Seconds]	Rows processed	Executions	Disk Reads [Blocks]	Disk Reads [MB]	Buffers Get [Blocks]	Buffers Write [Blocks]	Buffer Quality [%]	Elapsed Time per 1 Exec [Seconds]
2020-09-18	0x64DFCBB636DBF6F9	master	0.0037	0.0	0	1	32	0 MB	919	2	96.6	0.0037
2020-09-18	0x64DFCBB636DBF6F9	DBPLUS	0.0530	0.1	1 188	23	160	1 MB	21 098	2	99.3	0.0023
2020-09-21	0x64DFCBB636DBF6F9	DBPLUS	0.0551	0.0	1 512	28	190	1 MB	25 681	10	99.3	0.0020
2020-09-22	0x64DFCBB636DBF6F9	DBPLUS	0.0620	0.0	1 836	34	80	1 MB	31 153	2	99.7	0.0018
2020-09-23	0x64DFCBB636DBF6F9	DBPLUS	0.0661	0.1	2 106	39	96	1 MB	35 763	4	99.7	0.0017
2020-09-24	0x64DFCBB636DBF6F9	DBPLUS	0.0690	0.0	1 836	34	208	2 MB	31 180	2	99.3	0.0020
2020-09-25	0x64DFCBB636DBF6F9	DBPLUS	0.0570	0	1 458	27	105	2 MB	24 768	9	99.2	0.0021

□ Compare Plans — 0x64DFCBB636DBF6F9 (DBPLUS)

```
Show plan objects for 0x64DFCBB636DBF6F9
 ├Database: DBPLUS
 ├SELECT ( Cost - 0,057552 , Rows - 0 , CPU - 0 , IO - 0 )
 ⊖Sort ( Cost - 0,057552 , Rows - 30,5244 , CPU - 0,00034387 , IO - 0,0112613 )
    ⊖Compute Scalar ( Cost - 0,0459468 , Rows - 30,5244 , CPU - 0,00000305244 , IO - 0 )
       ⊖Compute Scalar-Sequence Project ( Cost - 0,0459438 , Rows - 30,5244 , CPU - 0,00000244195 , IO - 0 )
          ⊖Segment ( Cost - 0,0459414 , Rows - 30,5244 , CPU - 0,000000610488 , IO - 0 )
             ⊖Inner Join-Nested Loops ( Cost - 0,0459407 , Rows - 30,5244 , CPU - 0,000127592 , IO - 0 )
                ⊖Inner Join-Nested Loops ( Cost - 0,0316583 , Rows - 30,5244 , CPU - 0,000127592 , IO - 0 )
                   ⊖Sort ( Cost - 0,0233283 , Rows - 29,3939 , CPU - 0,000327711 , IO - 0,0112613 )
                      ⊖Lazy Spool-Index Spool ([@alerts].) ( Cost - 0,0117393 , Rows - 29,3939 , CPU - 0,0000000734847 , IO - 0 )
```

Additionally, it will be possible to analyze execution plans for the query for each database independently, for this purpose, by performing the Compare options, User will be able to compile the query plans for each database independently.

The change additionally includes improved mechanisms for calculating the share of a given query in the load for a given database in an SQL instance.



Date from: 2020/10/02 00:00 to: 2020/10/02 23:59

SQL SERVER INSTANCE LOAD

Click on the chart at specified snapshot time to view sql statements, waits or alerts executed during the time of 15 minutes

View load for: DBPLUS

Look for the database which has the sql statements with the biggest impact on the performance

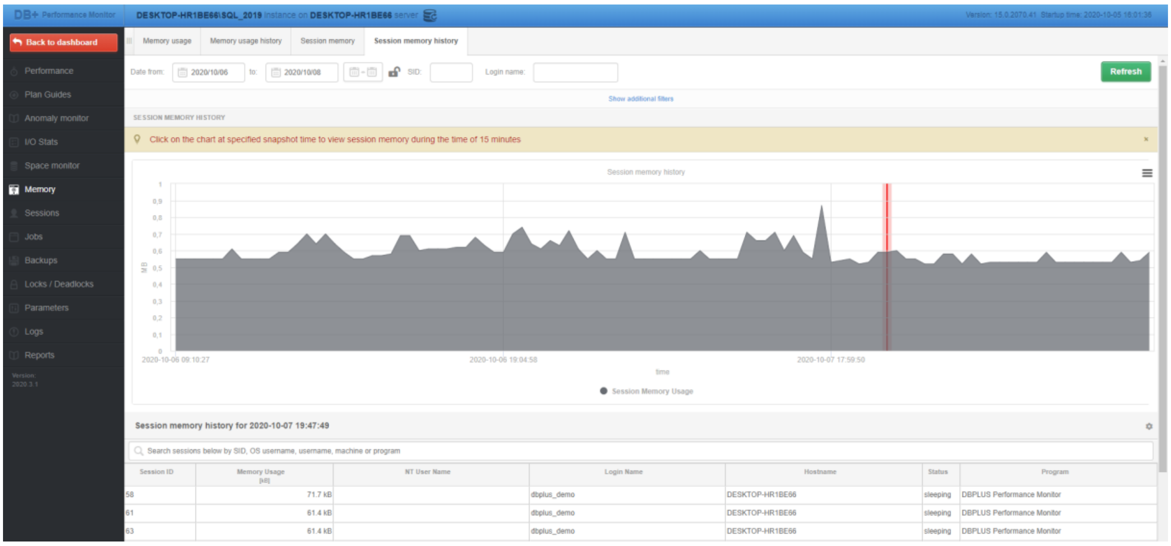
Average instance load splited by databases

Database	Elapsed Time[%]	Cpu Time [%]
DBPLUS	65.8	64.5
master	33.7	35
msdb	0.1	0.1
test5_nowe	0.1	0.1
test6	0.1	0.1
my_filestream_DB_...	0.1	0.1
my_filestream_DB_...	0.1	0.1
tempdb	0	0
model	0	0
All databases	100%	

1.4 *Collecting information about the Session memory*

In the latest version, the ability to collect information about sessions that waste memory has been added. To verify which sessions were utilizing memory at a given moment, go to the details of a given database, then select Memory from the left menu and go to the Session memory history tab.

As before, the graph with the distribution of memory consumption will be presented, click on a point on the graph will give the User information about the sessions that utilized memory at a given point in time. Like in the picture below.



DESKTOP-HR1BE66\SQL_2019 instance on DESKTOP-HR1BE66 server

Version: 15.0.2070.41 Startup time: 2020-10-05 16:01:36

Date from: 2020/10/06 to: 2020/10/08 SID: Login name:

SESSION MEMORY HISTORY

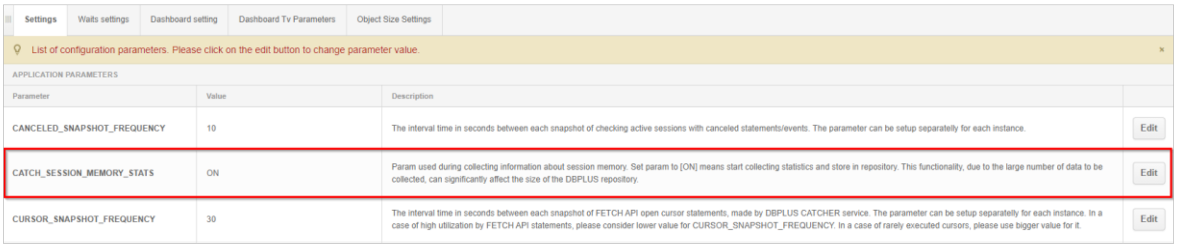
Click on the chart at specified snapshot time to view session memory during the time of 15 minutes

Session memory history for 2020-10-07 19:47:49

Search sessions below by SID, OS username, username, machine or program

Session ID	Memory Usage [kB]	NT User Name	Login Name	Hostname	Status	Program
58	71.7 kB		dbplus_demo	DESKTOP-HR1BE66	sleeping	DBPLUS Performance Monitor
61	61.4 kB		dbplus_demo	DESKTOP-HR1BE66	sleeping	DBPLUS Performance Monitor
63	61.4 kB		dbplus_demo	DESKTOP-HR1BE66	sleeping	DBPLUS Performance Monitor

To start collect detailed information about memory usage by sessions, change the value of the **CATCH_SESSION_MEMORY_STATS** parameter in the Configuration > Settings menu. Information about memory utilization sessions is **not available** by default.



List of configuration parameters. Please click on the edit button to change parameter value.

APPLICATION PARAMETERS

Parameter	Value	Description	
CANCELED_SNAPSHOT_FREQUENCY	10	The interval time in seconds between each snapshot of checking active sessions with canceled statements/events. The parameter can be setup separatelly for each instance.	Edit
CATCH_SESSION_MEMORY_STATS	ON	Param used during collecting information about session memory. Set param to [ON] means start collecting statistics and store in repository. This functionality, due to the large number of data to be collected, can significantly affect the size of the DBPLUS repository.	Edit
CURSOR_SNAPSHOT_FREQUENCY	30	The interval time in seconds between each snapshot of FETCH API open cursor statements, made by DBPLUS CATCHER service. The parameter can be setup separatelly for each instance. In a case of high utilization by FETCH API statements, please consider lower value for CURSOR_SNAPSHOT_FREQUENCY. In a case of rarely executed cursors, please use bigger value for it.	Edit

1.5 *Unused indexes report*

In the latest version, we have added the ability to run a report checking the use of indexes by the database engine in each period. The report looks for indexes that were not used by queries. The check consists in verifying whether the index was used in the execution plan in queries stored in the DBPLUS repository.

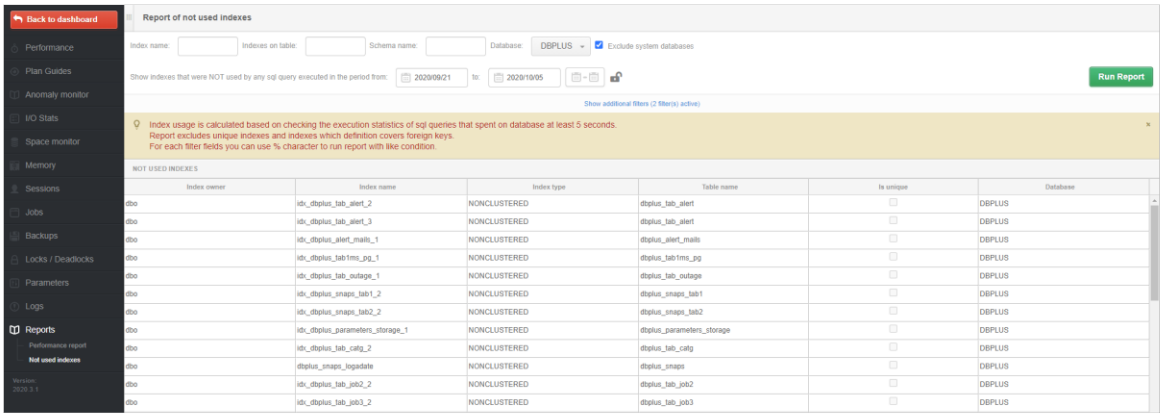
To execute a report, go to the instance details and then select the Not used indexes option from the Reports menu.

On the website, User can provide additional information to prepare a report:

- Index Name

- Indexes od Table
- Schema name
- Database – in this field, we can check all databases or indicate a specific database
- The period User is checking whether the index was used
- Additional filters where User limits the number of databases or schemas where the search will take place.

After performing the check, the User receives information on unused indexes in the indicated databases in the form of a table. The report can be exported to excel as a * .csv file.



Report of not used indexes

Index name:
Indexes on table:
Schema name:
Database: DBPLUS
☑ Exclude system databases

Show indexes that were NOT used by any sql query executed in the period from: 2020/09/21 to: 2020/10/05

Show additional filters (2 filter(s) active)

Index usage is calculated based on checking the execution statistics of sql queries that spent on database at least 5 seconds.
Report excludes unique indexes and indexes which definition covers foreign keys.
For each filter fields you can use % character to run report with like condition.

NOT USED INDEXES

Index owner	Index name	Index type	Table name	Is unique	Database
dbo	idx_dbplus_tab_alert_2	NONCLUSTERED	dbplus_tab_alert	□	DBPLUS
dbo	idx_dbplus_tab_alert_3	NONCLUSTERED	dbplus_tab_alert	□	DBPLUS
dbo	idx_dbplus_alert_mails_1	NONCLUSTERED	dbplus_alert_mails	□	DBPLUS
dbo	idx_dbplus_tab1ms_pg_1	NONCLUSTERED	dbplus_tab1ms_pg	□	DBPLUS
dbo	idx_dbplus_tab_outage_1	NONCLUSTERED	dbplus_tab_outage	□	DBPLUS
dbo	idx_dbplus_snaps_tab1_2	NONCLUSTERED	dbplus_snaps_tab1	□	DBPLUS
dbo	idx_dbplus_snaps_tab2_2	NONCLUSTERED	dbplus_snaps_tab2	□	DBPLUS
dbo	idx_dbplus_parameters_storage_1	NONCLUSTERED	dbplus_parameters_storage	□	DBPLUS
dbo	idx_dbplus_tab_catg_2	NONCLUSTERED	dbplus_tab_catg	□	DBPLUS
dbo	dbplus_snaps_logadate	NONCLUSTERED	dbplus_snaps	□	DBPLUS
dbo	idx_dbplus_tab_job2_2	NONCLUSTERED	dbplus_tab_job2	□	DBPLUS
dbo	idx_dbplus_tab_job3_2	NONCLUSTERED	dbplus_tab_job3	□	DBPLUS

1.6 *General improvements*

1.6.1 Improved ergonomics for database versions based on Azure technology

In the latest version of the application, we have improved the performance of the application. Some of the screens of the DBPLUS application, which presented data from SQL instances based on the cloud solution, did not work optimally. The problem was solved by adding special support for these types of platforms.

1.6.2 Hiding points on the graph

In the latest version, we have added the ability to hide the selected point on the chart. To hide a given point, hold down the **CTRL** key on the keyboard and at the same time **indicate the point** on the graph that is to be hidden. This functionality will significantly facilitate the performance of the analysis performed over a longer period. Hiding a point in the graph does not change the data available in the grid under the graph. The functionality is intended only to facilitate reading the data available in the chart. Changes have been made to the charts available on the tabs:

- Load trends
- SQL Details
- OS Stat
- I/O Stats.

1.6.3 Small fixes and improvements

1.6.3.1 SQL FIND search improved

In the latest version, we have improved the query search using the SQL FIND functionality, available in the SQL Details tab. Saving previously searched queries / objects has been improved. Capitalization does not matter.

1.6.3.2 Improved script display for "Show statement script with filled parameters”

In the latest version, we fixed a bug in the script display functionality for queries with a filled parameters. The problem only affected parameters of decimal type and resulted in no display of the right parenthesis for a parameter of this type. The problem has been fixed.

1.6.3.3 Support for different time zones

In the last version, we fixed the problem with handling different time zones. The problem was the discrepancy in the snap dates when the server with the DBPLUS application was in a different time zone than the database server it was connecting to. The problem has been handled.