

DBPLUS
Performance Monitor for Oracle
description of changes in version 2021.1

Date: April 9, 2021

Table of Contents

1.1. Anomaly Monitor	3
1.1.1. Generating a report based on Anomaly Monitor	6
1.1.2. Exclude queries from monitoring	6
1.2. Wait Dictionary	7
1.3. Enq: TM – contention lock support and lock export.....	8
1.4. Query formatting and parsing	9
1.5. General improvements	10
1.5.1. Load trends Compare	10
1.5.2. New RMAN Backup screen	10
1.5.3. I/O Stats with Sql Statements.....	11
1.5.4. Performance reports with the option to print in * .docx format	11
1.5.5. Quick selection of lines (Shift)	12
1.5.6. Additional filters search session history.....	12
1.5.7. End of support for the old version of Dashboard	12

Below is a list of changes to the DBPLUS Performance Monitor system for monitoring Oracle databases.

New in version 2021.1

1.1. Anomaly Monitor

The Anomaly Monitor module is available in the DBPLUS Performance Monitor application. The module presents information about detected problems in the monitored database. Data on the website are presented in two tabs:

- Anomaly Monitor
- Alerts Browser

The **Anomaly Monitor** tab presents the events detected in the monitored database based on the trend study for the most important performance statistics. The verification covers queries performed in the database, the performance of the disk array, and statistics on the main waits that affect the database performance. Monitoring has information about issues divided into classes:

- **Change Plan**

The problems presented in this class relate to changing the execution plan. In the event of detecting a decrease in the performance of a given query, the application verifies whether the event is related to a plan change, and at the same time checks whether a faster / better plan can be found in the history of the query. The number of days taken into account to find a faster plan depends on the **History Days for Plan Change Estimation** parameter available in the **Alerts Settings** menu in the General settings tab. Default value is set to 30 days.

- **CPU**

The CPU class occurs when problems with rising **resmgr: cpu quantum** wait events are detected. The increase in the level of this expectation is related to the pressure on the CPU by queries performed at a given moment in the database.

- **I/O**

This class indicates problems with the disk array. The application checks whether the indicators related to the performance of the disk array have deteriorated (eg Single Block Read Time). Additionally, the level of expectations related to I / O resources is verified.

- **Latch**

The occurrence of such problems is related to the high level of Latches. Depending on the source of the problem, the application informs the User about the reason for the decline in database performance caused by Latches.

- **Lock**

The class contains events related to the occurrence of locks in the monitored database.

- **Memory**

Problems related to this class indicate that there are memory problems in the monitored database.

- **New SQL Statement**

The reason for the occurrence of such problems is the appearance of a new query that has not been performed so far or has not been previously detected by DBPLUS monitoring. At the same time, it significantly utilizes the database resources where it is performed. Information about the detection of a new query will be visible for a period of 3 days (from the occurrence moment). It depends on the new parameter **Number of days to consider the statement as NEW** available in the **Alerts Settings** menu in the **General settings** tab. **The New SQL Statement** problem will also not occur if the query does not have an important impact on the performance of the monitored database.

- **Remote**

The problem arises because the response from the remote external host (database / application) is awaiting. The flow of data between hosts on the network and remote aspect applications require time to process each request addressed to it. The application that communicates with the remote host must wait for the read data to arrive.

- **SQL Statement**

The SQL Statements class indicates a problem with queries where overall Elapsed Time statistics have deteriorated. However, the source of the deterioration in query duration has not been identified.

- **SQL statement - buffer gets**

For queries assigned to this class, the amount of data read from the memory buffer has increased.

- **SQL statement - disk reads**

This class groups queries where an increase in the amount of data read from disk resources has been detected, compared to the historical trend for a given statistic.

- **SQL statement – executions**

For queries assigned to this problem class, an increase in the number of executions in a given snap was detected in relation to the query history for the last 30 days. The administrator should verify what is the reason for the sudden increase in the amount of performances associated with the problem query.

- **Space**

The presence of this class suggests a large increase in the use of UNDO space in the monitored database.

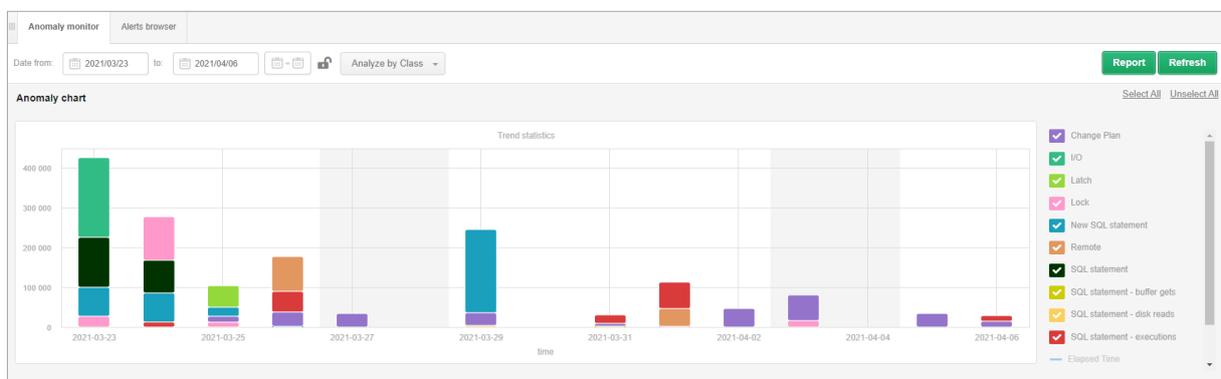
- **Other**

This class contains an increase in the level of expectations for which a dedicated class has not been defined. In the next versions of the application, the cases included in the Other class will be described and analyzed in terms of finding the source of the problem.

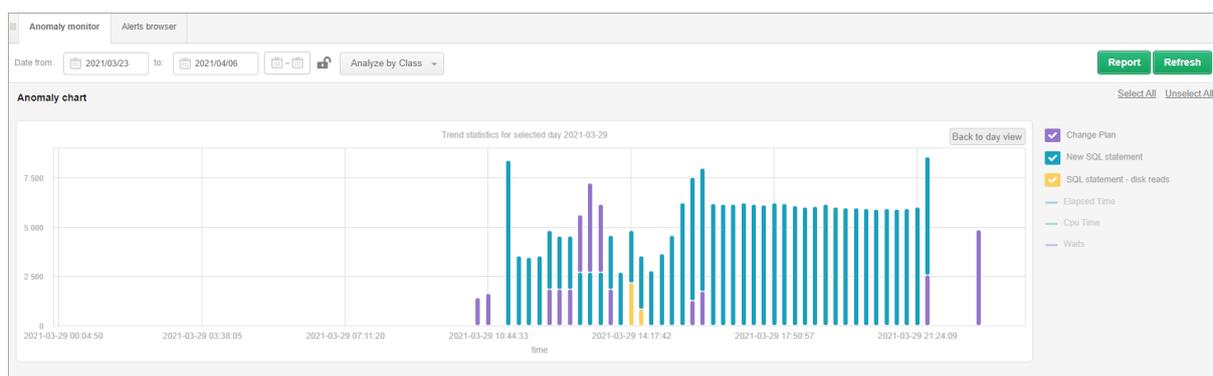
After enter the Anomaly screen, the monitor will be presented with a graph that shows the performance anomalies for the last two weeks.

Important!! The mechanism of calculating problems based on trends is not retroactive. Immediately after updating the application to version 2021.1, the screen may not show the above-mentioned data. Manual alert recalculation is available. Use the mechanism available in the Configuration - Alert settings menu in the Adhoc Alerts calculator tab for the monitored database.

Each class available on the chart is presented in a separate color. The height of the bar depends on the effect of the class on the database performance. The bigger / more serious problem is detected, the higher the column of the given color will be.



Below the graph, there is a list of performance problem classes that were detected in the presented period. To verify the problem, select the day by click on the chart to verify the performance problems. The next chart shows the problem classes found on a given day.



For a detailed analysis, click on the row with the problem class for which User wants to verify the cause of the problem.

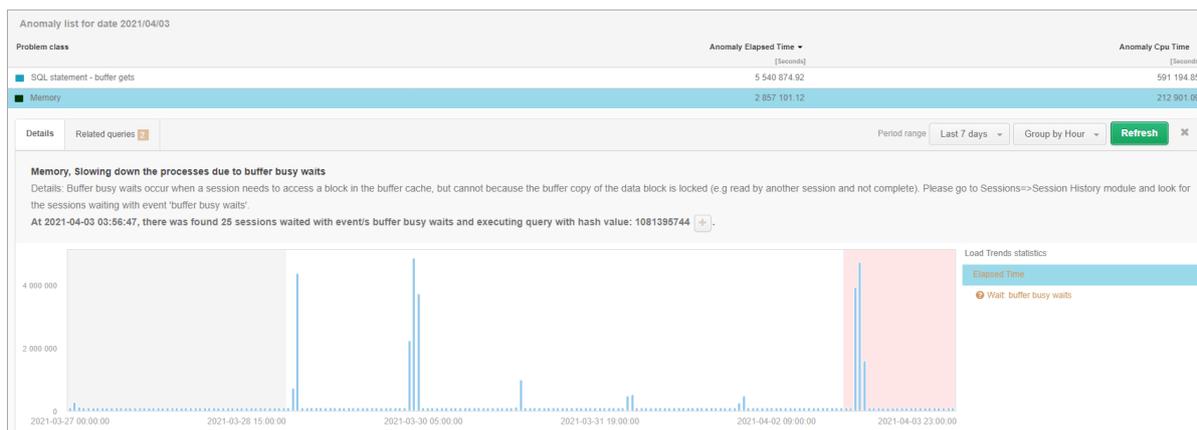
Problem class	Anomaly Elapsed Time <small>[seconds]</small>	Anomaly Cpu Time <small>[seconds]</small>
New SQL statement	208 741.73	128 591.74
Change Plan	31 540.09	6 222.31
SQL statement - disk reads	2 969.46	74.19

The query problem for which a plan change event has been detected is presented below. By selecting a given row, a panel with the details of a given query opens. In the legend on the right side of the graph, the statistics where the threshold values were exceeded are highlighted in orange.



A general description of the problem is presented above the graph (Details), and the fastest plan for the query under examination is indicated. When the plan changes, the chart presents information grouped for each plan.

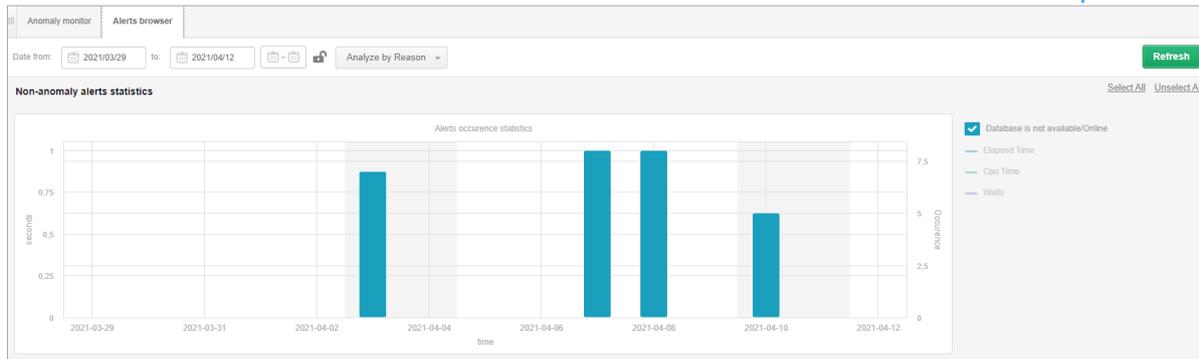
In the case of detection of a problem related to exceeding the trend level for a given wait (eg Buffer busy waits), the Details tab presents general information on the exceeded statistics for a given wait with information about the reason for the increase in the level of a given wait. In addition, the tab Related queries were related queries where the period of time waiting for the wait was the biggest.



The **Alerts browser** tab contains information on events not directly related to the performance of queries performed in the database. The tab presents alerts:

- Online,
- User own alerts,
- Associated with excess space used,
- Others not related to queries.

After entering the tab, alerts are presented in the form of a graph for the last 2 weeks. The height of the bar in the graph corresponds to the number of occurrences of a given problem on a given day. After clicking on a point on the graph, the User receives information about all the occurrences of problems on the indicated day.



1.1.1. Generating a report based on Anomaly Monitor

In addition to presenting performance problems in the application on the Anomaly Monitor screen, it is also possible to generate a report that includes performance problems detected by the application. The report contains information about the overall performance of the monitored database, information about leading waits and, above all, performance problems detected by the alert module for a given date range.

Report generation is possible by clicking the **[Report]** button. Before generating the report, User has the option of setting:

- the language version of the report (Polish / English),
- the date range for which the report is to be generated,
- number of queries assigned to a given problem class (3 by default),
- list of classes to be included in the report.

REPORT DEFINITION

Language: English Polish

Reported period: Begin: End:

Report title:

Report file name:

Max number of queries related to specific problem class:

Problem classes included in the report:

New SQL statement I/O SQL statement Lock Remote SQL statement - executions

Change Plan Latch SQL statement - disk reads SQL statement - buffer gets

The report generation time depends on the date range and the number of classes for which it is to be generated.

1.1.2. Exclude queries from monitoring

Excluding queries from monitoring is a useful functionality when the alarm thresholds are often exceeded for a given query and the Administrator cannot improve the performance of a given query (e.g. in the case of a query in system views or without limiting conditions). The configuration of excluding the query from monitoring is available in the **Configuration > Alert Settings settings** menu, in the **Exclude Alerts > SQL Statement Exclude** tab.

After entering the website, there are three options to exclude the query:

- for the query id Query hash,
- for the module from which the query is run,
- for text or a piece of query text.

Adding a new exclude is possible by clicking the **[Add new exclude]** button for the selected exclusion option. Exclusions from monitoring can always be made for a selected database. Excluding monitoring means that from that moment in the event of an alert related to the query, such alert will not be presented in the application and no e-mail notification will be sent.

Mail settings		General settings		Alerts definition		Reasons & Problems definition		Events subscription		Adhoc Alerts calculator		Exclude Alerts	
Filter by database: All databases													Refresh
Database Exclude		SQL Statement Exclude											+ Add new exclude
QUERY HASH EXCLUDE													
Database	Query Text	Hash Value	Enabled	Date added	Description								
NMA	SQL Analyze(1) / select /*+ full() no_parallel() no_parallel_index() dbms_stats cursor_sharing...	3303926762	<input checked="" type="checkbox"/>	2021-04-13 17:06:10	Flag change from Anomaly Monitor								
NMA	select * from (select /*+ INDEX(this_INDEXT_DOC_ALL_HE_VIEW) /* this_id as id5_0 this_etc...	2118537747	<input checked="" type="checkbox"/>	2021-04-13 17:06:22	Flag change from Anomaly Monitor								
NMA	select * from (select /*+ INDEX(this_INDEXT_DOC_ALL_HE_VIEW) /* this_id as y0_ from DOCU...	2557996310	<input checked="" type="checkbox"/>	2021-04-13 17:06:29	Flag change from Anomaly Monitor								
NMA	create table *OSBIC	2090895666	<input checked="" type="checkbox"/>	2021-04-13 17:06:41	Flag change from Anomaly Monitor								
MODULE EXCLUDE													
Database	Module	Enabled	Date added	Description									
CA	Data Replicator - SAFO_TO_KAFKA	<input checked="" type="checkbox"/>	2021-04-09 19:33:15	DBPLUS Data Replicator queries									
EXCLUDE BASED ON QUERY TEXT													
Database	Query text	Enabled	Date added	Description									
BDR	SELECT SUM(DECODE(event	<input checked="" type="checkbox"/>	2021-04-13 17:07:57	Exclude by query text									

In the case of re-enabling the query for alerting, select the editing options from the list and disable the given configuration or delete the entered configuration by selecting the **[Delete]** option. After changing the settings, information about problems for the query will be presented again.

Important! After re-enabling monitoring, alerts that were not displayed at the time when the query was excluded from monitoring will not be restored.

Excluding by content module or query id at the moment generates an alert during the procedure, snap verifies that queries complies with the conditions of exclusion and does not generate alerts for designated queries. Verification takes place each time during the alert calculation procedure (every 15 minutes).

On the pages where alerts are visible, we have added the ability to quickly exclude queries from monitoring. To exclude, click on the "bell" icon. After accepting the selection, the given query will be added to the exclusion configuration.

Anomaly list for date 2021/03/28 for class Change Plan

Hash value	Sql Id	Query text	Anomaly Elapsed Time (seconds)	Anomaly Cpu Time (seconds)
2673054730	8p9mlygg750a	SELECT /*+ ROWID ID KOR MAO_KOD.ROD_D_KOD.NR_DAT_W KH_KOD.WAR_B.WAR_N.WAR_V.WAR_O.KH_KOD_2.WAL_KOD_sta_for KUR.KUR_W.TYP_D_ID.KIE.SP_KOD.SPO_PL_KOD.OSO_KOD.OSO...	12 905.14	1 903.06

Details: **Change Plan, Change of the Execution Plan**
 Details: The problem was caused by Oracle's Optimizer that changed execution plan. Before executing the query, the Optimizer verify which execution plan should be used. Add necessary HINTS or force better execution plan (create Outline/Baseline or Profile) to stabilize query performance. For detailed information click "Plus" button on query row and check query details.
 Statistics: Elapsed Time per 1 exec. Last value: 1821 s, History value: 0.0891 s, Faster plan found: 3136659053, actual slower plan: 1451516060. Statistics difference: 0.0575 vs. 1821 s

SQL Query statistics: Elapsed Time, Cpu Time, Cpu Time per 1 exec (Faster plan found), Elapsed Time per 1 exec (Faster plan found), Fetches, Disk reads, Rows processed

In order to re-enable the query for monitoring, the User may again un-click the exclusion from the alert level and restore the query for monitoring. In this case, the exclusion entry will be removed from the configuration.

1.2. Wait Dictionary

In the application, the User can obtain information about the wait level and what the wait is related to, and how to try to solve the problem with the high level of a given wait (expectations).

Sample description for wait: **db file sequential read.**

db file sequential read

Class
User I/O

Description

The db file sequential read wait event is initiated by SQL statements that perform single-block read operations against indexes, rollback (or undo) segments, and tables (when accessed via rowid), control files and data file headers. This wait event normally appears as one of the top five wait events, according to systemwide waits.

The db file sequential read waits occurs:

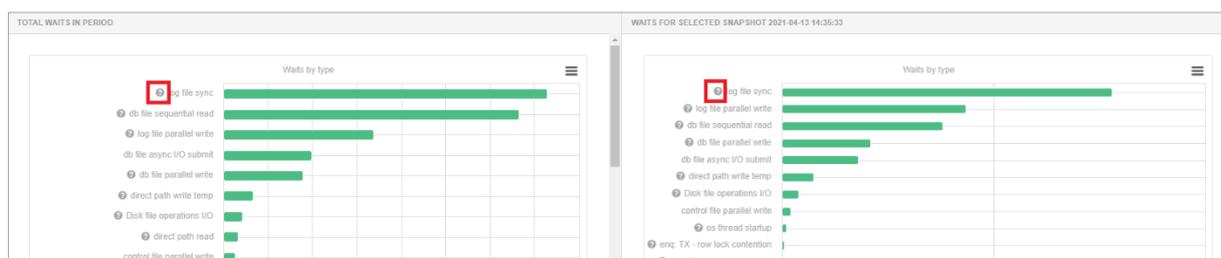
- The Oracle process wants a block that is currently not in the SGA memory, and it is waiting for the database block to be read into the SGA from disk.
- Significant db file sequential read wait time is most likely an application issue.

Solution

- Check the SQL statement reading a large number of blocks. Verify that it is possible to improve query performance by adding appropriate indexes, reducing the number of block reads. Go to the **Session history** tab and search for the queries waiting with wait db file sequential read.
- Check INSERT statements can also wait for this event because it is forced to update inefficient indexes.
- Increase the cache buffer so that more blocks are already in memory instead of having to be read from disk. The query will still have to read the same number of blocks, so tuning is the first recommendation, but if you can't tune the statement, a query to read blocks from memory is much faster than from disk.

Information on waits can be found primarily in the Waits tab. Currently, the dictionary contains information on over 30 most common expectations in an Oracle database. In subsequent versions of the application, the dictionary will be systematically enlarged and updated.

In the case of the **Waits / Overview** tab, information about waits can be read by clicking the question mark icon for waits visible for each of the graphs. If the icon is missing, it means that the given wait has not been added to the dictionary yet and will be completed in subsequent versions.



Similarly tab **Waits / Analyze** information about the waits are visible at the level of the rows in the table and on the main **Dashboard** screen that shows a graph of the top statistics waits for the past 15 minutes.

1.3. Enq: TM – contention lock support and lock export

On the Locks / Locks history screen, support for the TM lock problem has been added. The problem of the presence of locks (enq: TM - contention) results from the lack of indexing of foreign keys in the child / parent tables of the table where the data modification is made. The solution to the above problem is to add the missing index.

In the DBPLUS application, when this type of blockade occurs, after selecting a point on the chart and selecting a session row with a TM blockade, information about the reasons for the blockage will be presented. Also, a script will be displayed that allows to set the missing index, as in the example below:

List of locked sessions at snapshot time: 2021/03/17 13:17:01

- SID: 17330 Serial#: 42927 Session status: ACTIVE Lock Type: TM (DML enqueue lock) BLOCK time (sec.): 2 User Name: ICEDOC (Os User: pawel) Machine: ubuntu Module: DBBeaver 217070 ? SQLEditor ?procedure sprawdz?
- SID: 1529 Serial#: 35789 Session status: ACTIVE Lock Type: TM (DML enqueue lock) WAIT time (sec.): 3 User Name: ICEDOC (Os User: pawel) Machine: ubuntu Module: DBBeaver 217070 ? SQLEditor ?Script287sql?
- SID: 13200 Serial#: 1897 Session status: ACTIVE Lock Type: TM (DML enqueue lock) WAIT time (sec.): 2 User Name: ICEDOC (Os User: edoc) Machine: edoc-app02n Module: JDBC Thin Client

SQL STATEMENT FOR SESSION SID: 1529

```
DELETE CONTACT_APP CA WHERE ID = :B1 AND ID1 = :B2
```

SESSION DETAILS		DBPLUS ADVISOR	
Request	5	Description:	
Sid	1529	A high level of this event indicates that there are restrictions on undeclared foreign keys. This happens when a dependent or child table's foreign key constraint that references a parent table is missing an index on the associated key. Oracle acquires a table lock on a child table if it's performing modifications on the primary key column in the parent table that's referenced by the foreign key of the child table.	
LockType	TM	Solution:	
LockTypeDescription	(DML enqueue lock)	You need to create an index in the child table performing on the column that references the parent table.	
ID1	117303	Create the missing indexes with the script below to solve the enq: TM - contention wait problem:	
ID2	0		
Lmode	3		
CTime	3		
Block	1		
OS User Name	pawel		
Serial#	35789		
User Name	ICEDOC		
Status	ACTIVE		
Machine	ubuntu		
Module	DBBeaver 217070 ? SQLEditor ?Script287sql?		

We encourage Users to verify the presence of TM type locks in databases. This can be done from the **Waits / Analyze** tab. All Users need to do is select the date range to be checked and select the line with wait **enq: TM - contention**. If User finds this type of lock, we recommend to search for sessions that are waiting with a given lock, and to download information about the missing index.

Checking on a given index takes place online, each time for a given session. If, after clicking on the session details, information about the missing index is not presented, it will mean that the missing index has already been added or the tables participating in the lock no longer exist in the monitored database.

If the User wants to send information about blockades, he must indicate the point on the chart where there were blockades. Then, after clicking the right button on the mouse, an option about the possible export of locks will be displayed. The export is performed to a file in the *.csv format.

List of locked sessions at snapshot time: 2021/04/09 05:50:02

- SID: 7629 Serial#: 9424 Session status: INACTIVE Lock Type: TX (Transaction enqueue lock) BLOCK time (sec.): 241081 User Name: SP (Os User: hybriz) Machine: prd-app-02 Module: JDBC Thin Client
- SID: 4661 Serial#: 9409 Session status: ACTIVE Lock Type: TX (Transaction enqueue lock) WAIT time (sec.): 240251 User Name: SP (Os User: hybriz) Machine: prd-app-08 Module: JDBC Thin Client
- SID: 4669 Serial#: 51781 Session status: ACTIVE Lock Type: TX (Transaction enqueue lock) WAIT time (sec.): 240414 User Name: SP (Os User: hybriz) Machine: c-prd-app-11 Module: JDBC Thin Client
- SID: 6926 Serial#: 32750 Session status: ACTIVE Lock Type: TX (Transaction enqueue lock) WAIT time (sec.): 2405105 User Name: SP (Os User: hybriz) Machine: c-prd-app-02 Module: JDBC Thin Client
- SID: 4519 Serial#: 33418 Session status: ACTIVE Lock Type: TX (Transaction enqueue lock) WAIT time (sec.): 240608 User Name: SP (Os User: hybriz) Machine: c-prd-app-06 Module: JDBC Thin Client
- SID: 2553 Serial#: 10694 Session status: ACTIVE Lock Type: TX (Transaction enqueue lock) WAIT time (sec.): 240882 User Name: SP (Os User: hybriz) Machine: c-prd-app-12 Module: JDBC Thin Client

1.4. Query formatting and parsing

In order to speed up the analysis of queries in the DBPLUS application, the query parsing mechanism has been used. This functionality is based on the analysis of components of query objects (tables, indexes). Thanks to the query parsing mechanism, it is possible to quickly verify which tables and columns are used in the analyzed query. The parsing mechanism is available after clicking on the **Show Plan Objects for** the link visible in the query plan. To run the parsing, click on the **Parser SQL Query** option (see the example below). After correct parsing, the User can highlight tables and columns for selected objects. In the latest version, we improved the mechanism in terms of handling new query syntaxes and, above all, the speed of operation.

The screenshot shows the DBPLUS application interface with the following components:

- SQL TEXT (HASH VALUE: 298637718):** Contains an SQL query with comments. A red box highlights the comment: `--(comment substituted by DBPLUS) bind variables replaced by: --at (51,40) VARCHAR2(1('SYS_B_092')) => VARCHAR2(255) --at (22,41) VARCHAR2(1('SYS_B_091')) => VARCHAR2(255)`
- Objects Explorer:** A tree view showing the database schema structure.
- Parser SQL Query:** A button used to initiate the query parsing process.
- EXPLAIN PLAN (PLAN HASH: 367261219):** Displays the execution plan for the query, including details like `TABLE ACCESS (FULL) TAB_MAR`, `TABLE ACCESS (FULL) TAB_ART`, and `TABLE ACCESS (FULL) TAB_ART`.
- OBJECTS USED IN EXPLAIN PLAN:** A table listing the objects used in the plan, including tables like `TKL_INTER`, `TKL_INTER`, `TOW_PK`, `TOW`, `TKL_INTER`, and `MLUKASIK`.
- INDEXES FOR SELECTED OBJECT INTER.TOW:** A table listing the indexes for the selected object, including `TOW_SP_PODST_ZW_FK_I`, `TOW_TYP_LOG_FK_I`, `TOW_NR_KAT_UPP`, `TOW_KOD_UFL_WEW_UK`, `TOW_KOD_UFL_I`, `TOW_KOD_P_RZR_STS_I`, and `TOW_OBU`.

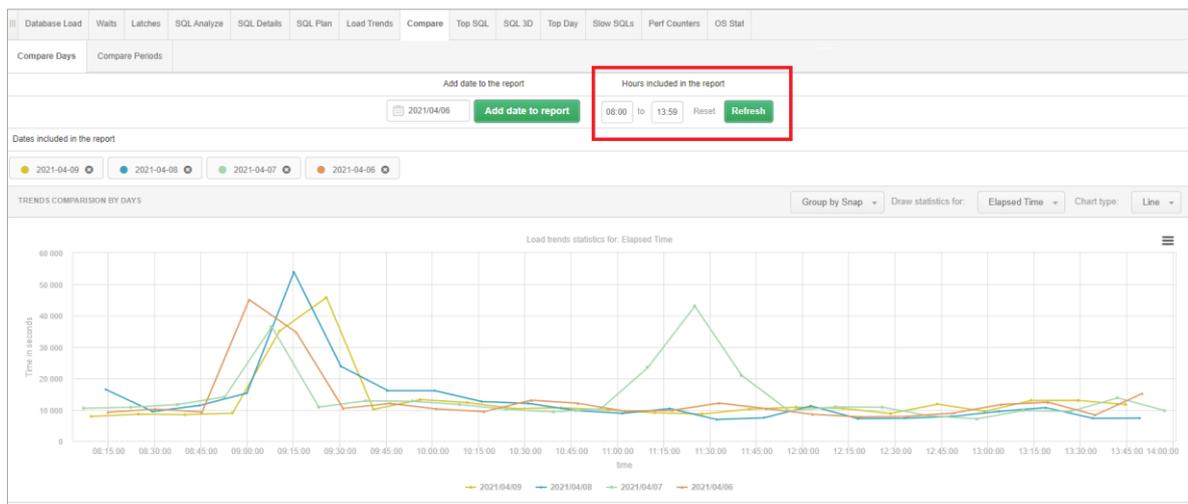
In some cases, the query parsing mechanism forces the query content to be changed by substituting a specific data type for a bound variable. Only variables are changed. Information about the changes made is always visible under the text of the query, marked with an additional comment:

--(comment inserted by DBPLUS) Bind variables replaced by:

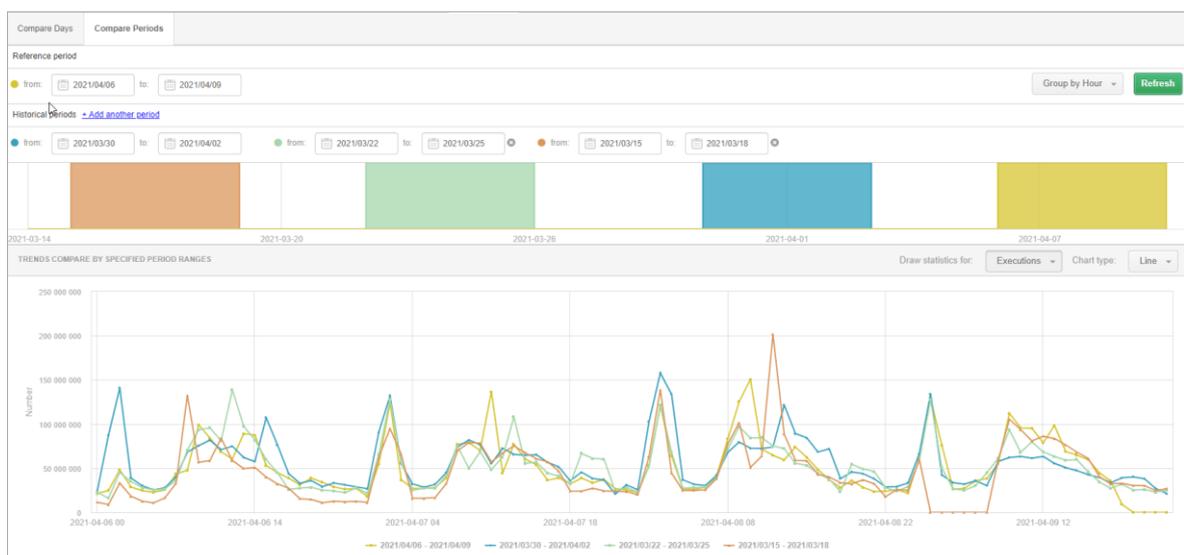
1.5. General improvements

1.5.1. Load trends Compare

In the application, it is possible to compile performance statistics for individual days. To do this, go to the **Compare - Compare Days** tab. By selecting interesting days, the User can easily compile data for a given statistics for the indicated days. Currently, we have added the ability to narrow the range for hours.



Additionally, the latest version offers the possibility of compiling multiple periods on one chart. By going to the **Compare Periods** tab, it is possible to verify the database performance for a given statistic. Several periods should be displayed at once.

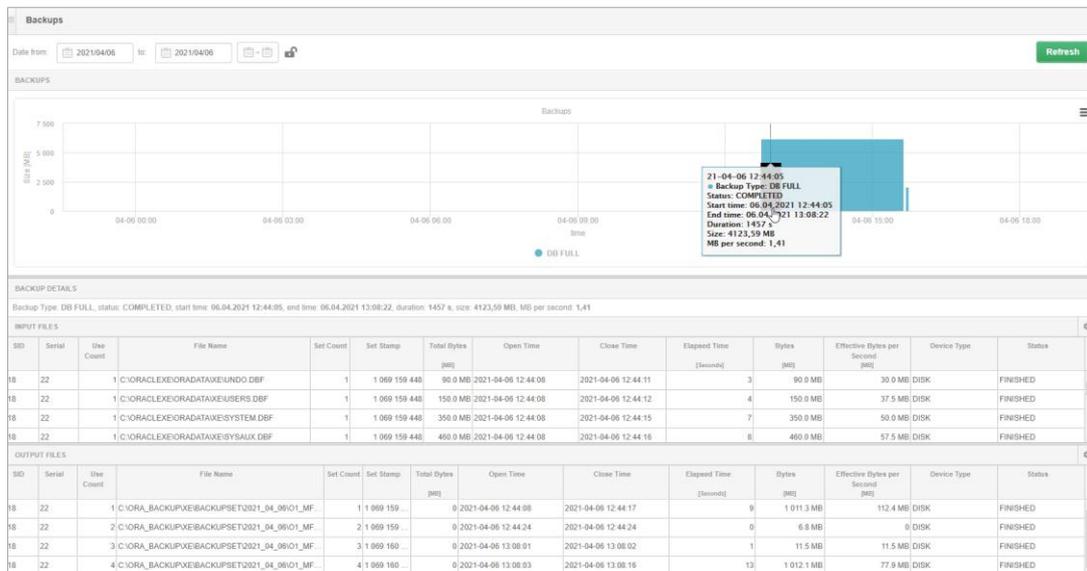


1.5.2. New RMAN Backup screen

The DBPLUS application enables the User to analyze RMAN backups performed in the monitored database. Information about the performed backups is available in the **Backups** menu. In the latest version, we have improved the way of presenting backups on the chart as well as in the tables below the chart. After clicking on

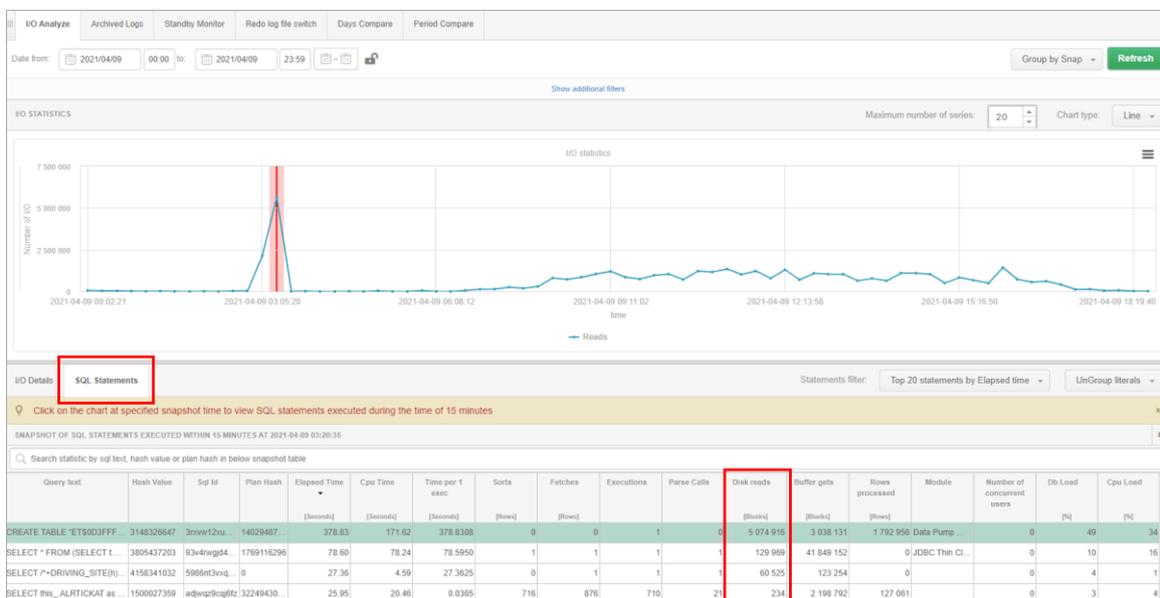
a given backup on the chart, it will be selected and the data in two tables **INPUT** and **OUTPUT FILES** will be presented under the chart.

Part of the columns available so far in the tables has been hidden by default, and it is possible to add (**Show hidden columns**) in the settings for the table after clicking on the icon "cog".



1.5.3. I/O Stats with Sql Statements

In the DBPLUS application, disk array performance monitoring is available. This information is collected from the system views of the database as part of the snap procedure every 15 minutes and presented in the **I/O Stats - I/O Analyze** tab. By going to the SQL Statements tab, the User can easily connect whether the change in the disk array statistics is related to the query being executed at that time. The example below shows an increase in the amount of read data caused by the query create a table based on the query result (CREATE TABLE AS SELECT).



1.5.4. Performance reports with the option to print in *.docx format

In the application, it is possible to generate a report (Performance report) which contains a list of top queries that operate in the monitored database. Depending on the selected scope, the report presents information about the queries that have the greatest impact on the database performance in the selected scope. Until now, the data was only available from the application level. From now on, the User can print the report in the *.docx format and send it for analysis outside the DBPLUS application.

1.5.5. Quick selection of lines (Shift)

In the SQL Analyze screen, we added options to quickly select multiple rows in a table. Quick selection is available when User selects the first row in a group, then selects the last row with the **SHIFT** button pressed.

1.5.6. Additional filters search session history

The session history screen has the option to search sessions with **Show additional filters**. In the latest version, we have added the ability to search by the **OS USER** column.

The screenshot shows the 'Sessions history' tab in the DBPLUS interface. At the top, there are search filters for 'From' (2021/04/11 00:00) and 'to' (2021/04/11 23:59), along with fields for 'Using Hash Value/Sql Id', 'Enter hash value or sql id', 'Username', 'Enter username', and 'Sid'. A 'Refresh' button is on the right. Below the filters, there is a section for 'All Waits' and 'Waits selected to filtering'. The 'OS User' filter is set to 'SYSTEM'. A table below shows session details with columns: Logdate, Sid, Serial#, Hash Value, User, Active Time (seconds), Schema, OS User, Machine, Program, Module, Wait, and Bloc. sessl.

Logdate	Sid	Serial#	Hash Value	User	Active Time (seconds)	Schema	OS User	Machine	Program	Module	Wait	Bloc. sessl
2021-04-11 20:41:00	1317	57259	3033312496	DBPLUSNEW	0	DBPLUSNEW	SYSTEM	TERSQLMON	DBPLUS Oracle Catc...	Session Resources S...	SQL*Net message from client	0
2021-04-11 20:36:56	1415	33909	2860469888	DBPLUSNEW	0	DBPLUSNEW	SYSTEM	TERSQLMON	DBPLUS Oracle Catc...	Snapper	SQL*Net message from client	0
2021-04-11 20:36:56	1320	19651	3192435587	DBPLUSNEW	0	DBPLUSNEW	SYSTEM	TERSQLMON	DBPLUS Oracle Catc...	Snapper	db file sequential read	0
2021-04-11 20:34:46	581	46729	3861668669	DBPLUSNEW	2	DBPLUSNEW	SYSTEM	TERSQLMON	DBPLUS Oracle Catc...	Snapper	db file parallel read	0
2021-04-11 20:19:32	581	46729	3861668669	DBPLUSNEW	0	DBPLUSNEW	SYSTEM	TERSQLMON	DBPLUS Oracle Catc...	Snapper	db file parallel read	0
2021-04-11 20:13:29	329	25705	2328843419	DBPLUSNEW	0	DBPLUSNEW	SYSTEM	TERSQLMON	DBPLUS Oracle Catc...	Sessions/Undo/Lock...	SQL*Net message from client	0
2021-04-11 20:04:17	71	21893		DBPLUSNEW	0	DBPLUSNEW	SYSTEM	TERSQLMON	DBPLUS Oracle Catc...	DBPLUS Oracle Catc...	log file sync	1089

1.5.7. End of support for the old version of Dashboard

In version 2021.1, we ended support for the "old" Dashboard. This screen will no longer be expanded and will be replaced by the current Dashboard screen allowing the User to simultaneously analyze all available platforms. The link that points to the old Dashboard has been redirected.