SAP® MaxDB™ Expert Session

SAP® MaxDB™ Database Analyzer

Christiane Hienger, Bettina Laidler November 12, 2013

Public

SAP® MaxDB™ – Expert Session

SAP® MaxDB™ Database Analyzer

Christiane Hienger
Bettina Laidler
IMS MaxDB/liveCache Development Support
November 12, 2013

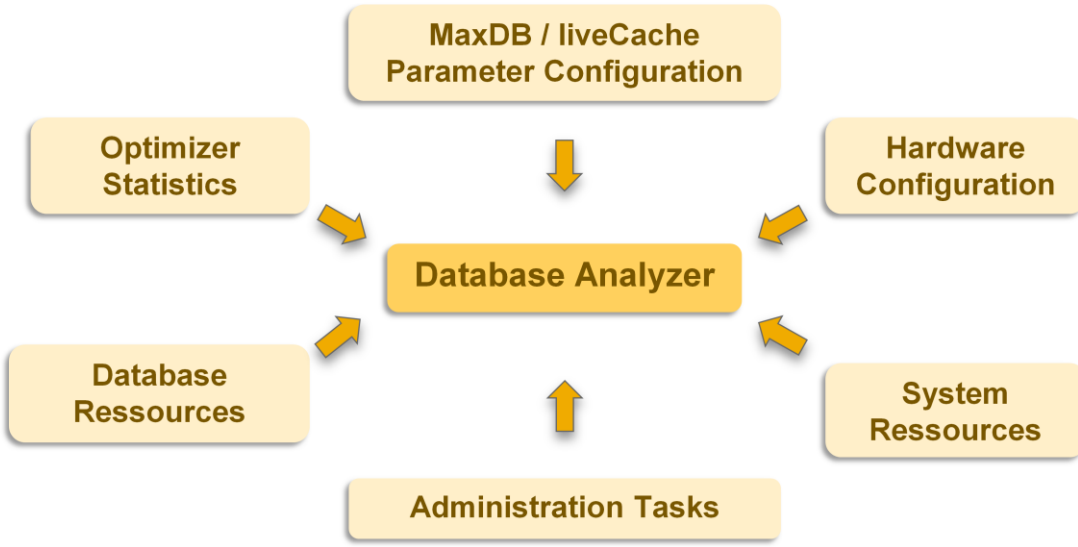
Agenda

1. Introduction
2. Functional Chain
   - 2.1. Software Components
   - 2.2. How Database Analyzer Collects Statistics
   - 2.3. Log Files and Views
3. Ways to Manage Database Analyzer
   - 3.1. Start/Stop Database Analyzer
   - 3.2. Display Collected Statistics
   - 3.3. Statistics Aggregation
   - 3.4. Statistics Administration
4. Parameter Check With Database Analyzer
5. Expert Analysis
   - 5.1. Optimize Runtime of Data Backup
   - 5.2. Aggregation Analysis
6. Additional Useful Information

Agenda

1. Introduction
2. Functional Chain
3. Ways to Manage Database Analyzer
4. Parameter Check With Database Analyzer
5. Expert Analysis
6. Addional Useful Information

1. Introduction



The Database Analyzer is a SAP MaxDB tool for long-term performance analysis. It creates snapshots about the database status in configurable intervals and logs these in several files.
This provides statistics data, which also enables an analysis of past performance issues.

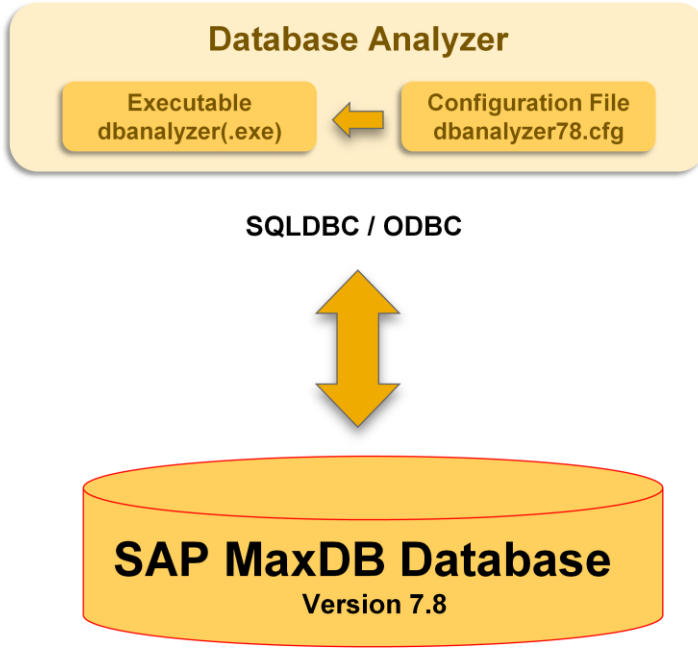
The root cause of performance issues could be different. Maybe the reason is related to:

- not optimal MaxDB/liveCache parameter configuration
- lack of hardware resources (CPU, memory, swapping)
- not optimal optimizer strategies
- high I/O response times (reading data, writing log)
- collisions on SQL locks, critical regions, internal structures
- administration tasks running in parallel (backup, update statistics, consitency check)
- …

Agenda

1. Introduction
2. Functional Chain
3. Ways to Manage Database Analyzer
4. Parameter Check With Database Analyzer
5. Expert Analysis
6. Additional Useful Information

2.1. Software Components (1)



The Database Analyzer consists of two components, the executable and the configuration file. The Database Analyzer program (executable dbanalyzer(.exe)) has no knowledge about the data to be collected. It is just “infrastructure”. The “heart” of the monitoring via Database Analyzer is the configuration file (dbanalyzer<version>.cfg).

2.1. Software Components (2)

Executable dbanalyzer(.exe)	Configuration file dbanalyzer<version>.cfg
• independent from the MaxDB/liveCache version to monitor	• the statistics that have to be collected and from which table they have to be selected from
• able to connect to a MaxDB/liveCache instance via local (shared memory, default) or remote (TCP/IP, SAP Ni, for support purposes) communication	• in which file which statistics have to be stored (grouped by theme)
• communication is realized via SQLDBC (new) / ODBC (old)	• how often statistics have to be collected (different intervals for different statistics possible)
• able to create views, performing SELECT statements, create/write to .csv/.prt files	• rules, when statistics indicate a performance issue and a rating how critical it is (different warning levels are supported)
• to interpret a configuration file of a specified structure	

The Database Analyzer executable is MaxDB kernel software independent. The Database Analyzer configuration file is release dependent. With each software installation only one executable related to the MaxDB kernel software version is delivered. Multi Database Analyzer configuration files are part of the software installation package for each MaxDB major Kernel version one.

Database Analyzer configuration file depends on the MaxDB/liveCache version to be monitored (dbanalyzer77.cfg, dbanalyzer78.cfg , … located in <installtion_path>\env)

2.2. How Database Analyzer Collects Statistics

The Database Analyzer collects statistics periodically

- interval as startup parameter (default 900s)
- some statistics will be collected more often or less often; defined in configuration file; no startup option e.g. statistics about running commands

All statistics evaluated/logged by the Database Analyzer are based on MaxDB/liveCache system views (or views that are based on system views)

- The Database Analyzer collects statistics periodically.
  - The snapshot interval is given as a startup parameter (default 900s)
  - Some statistic values are collected more often, independent from the user specified interval.
  - Some statistic values which change less often or produce higher workload during collection are not collected in each interval. These user interval independent statistics values are defined in the configuration file and their interval cannot be modified (no startup option).

- Source of all statistics evaluated/logged by the Database Analyzer are MaxDB/liveCache system views (or specific Database Analyzer views that are based on system views).

- Most of the specific Database Analyzer views are created by the first start of Database Analyzer (schema <sysdba>, e.g. superdba). Additional Database Analyzer specific views are created by the Database Analyzer parameter check in schema <sysdba> too. Therefore it is important to use the correct user to start the Database Analyzer Check (note: 1423935).

2.3. Log Files and Views

Administrative Files	<rundirectory>/analyzer
DBAN.err	error log file
DBAN.inf	information regarding an active Database Analyzer (pid, session, interval, used configuration file, …)
DBAN.pid	contains the pid of an active Database Analyzer process
DBAN.run	indicates that a Database Analyzer process is already running
DBAN.sid	contains the session id inside the MaxDB/liveCache, to which the Database Analyzer is connected to

Statistics Files	<rundirectory>/analyzer/<YYYYMMDD>
DBAN.prt	text file, that contains useful information, critical warnings
DBAN_*.csv	contains raw statistics, semicolon separated values
DBAN_*.prt	text file, that contains information for a variable number of information (lines) per snapshot

All administrative files of the Database Analyzer are located in subdirectory *analyzer* of the rundirectory of the MaxDB/liveCache instance to be monitored.

A bunch of statistics files are updated per interval with the collected statistics. For each day the statistics files will be stored in a separate folder.

DBAN.prt is evaluated by tools (DBACOCKPIT/LC10, Database Studio), different warning levels will be displayed in different colors.

The first three lines (header) of each file DBAN_*.csv contain information about the content of each column. This header is evaluated by the tool for display / aggregation purposes.

2.3. Log Files and Views

Database Analyzer Views (schema <sysdba>)	Based on MaxDB/liveCache System Views
DBAN_ACTIVEDIAGNOSTICFUNCTIONS	SYSINFO.ACTIVEDIAGNOSTICFUNCTIONS
DBAN_CACHE_INFORMATION	SYSINFO.IOBUFFERCACHES, SYSINFO.DATACACHE
DBAN_FILLING	SYSINFO.DATASTATISTICS, SYSINFO.LOGSTATISTICS
DBAN_IOTHREAD_STATISTICS	SYSINFO.IOTHREADSTATISTICS
DBAN_LOG_IO	DOMAIN.SYSMON_TASK_DETAIL
DBAN_MACHINEUTILIZATION	SYSINFO.MACHINEUTILIZATION
DBAN_NUM_CONNECTED_USERTASKS	DOMAIN.SYSMON_US
DBAN_SHOW_ACTIVE_TASKS	DOMAIN.SYSMON_ACTIVE_TASK, SYSINFO.TASKGROUPSTATISTICS
…	

Database Analyzer creates its own views in the MaxDB/liveCache instance to be monitored in schema <sysdba>. All these views (DBAN_*) are based on one or more MaxDB/liveCache system views. Only a subset is displayed here.

The MaxDB system views contain huge amount of information which is not necessary totally collect with each snapshot, therefore the <DBAN_views> contain only a subset of information.

The information in the Database Analyzer log files are based on the Database Analyzer views and the system views.

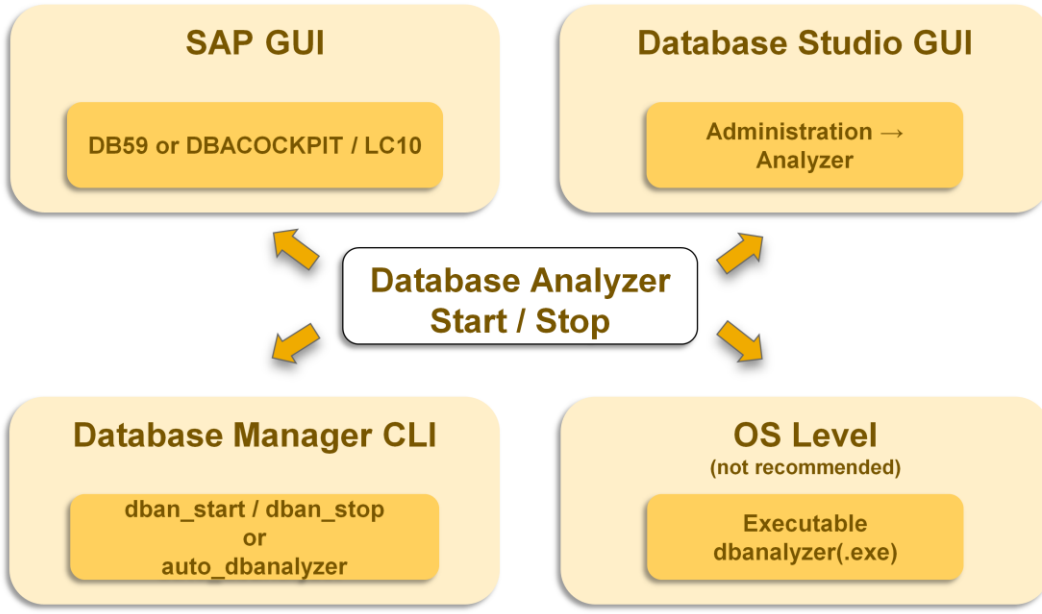
To check which Database Analyzer view is based on which system view use the following SELECT statement:

```
select * from viewdefs where viewname = <DBAN_view>
```

Agenda

1. Introduction
2. Functional Chain
3. Ways to Manage Database Analyzer
4. Parameter Check With Database Analyzer
5. Expert Analysis
6. Additional Useful Information

3.1. Start/Stop Database Analyzer

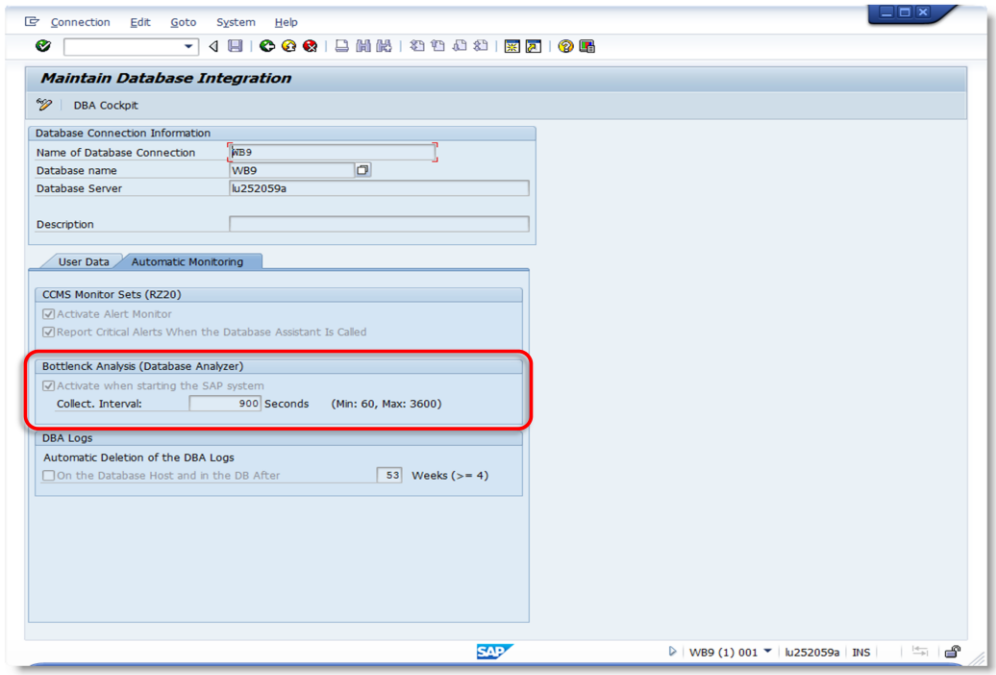


There are several ways to start or stop the Database Analyzer. To do this the graphical tools SAP GUI or Database Studio GUI can be used. On operating system level Database Manager CLI commands are available to start / stop the Database Analyzer manually or automatically.

For test purposes it is possible to start the Database Analyzer executable manually on OS level. It is used for support purposes, mostly in conjunction with a special configuration file (like the parameter check configuration file).

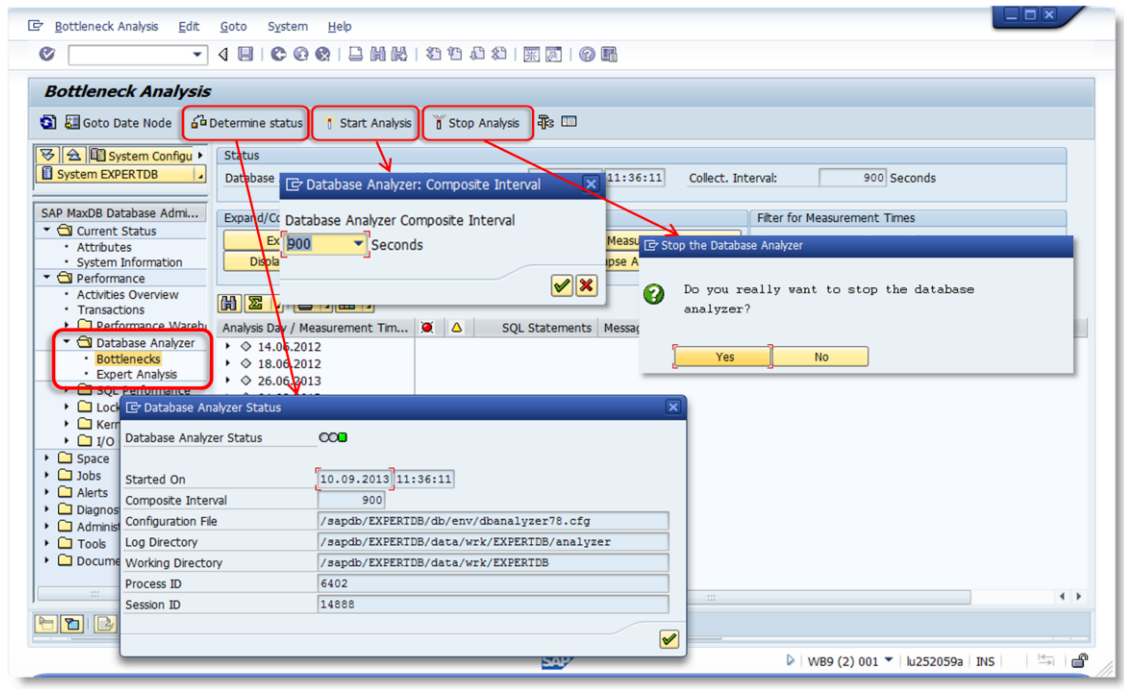
Hint: Please take care not to start a Database Analyzer on the same system twice (eg. via DBACockpit and on OS level). Both Database Analyzer would write into the same log files if the default output directory is not changed via option ‘-o’.

3.1.1. Start/Stop with SAP GUI: Transaction DB59



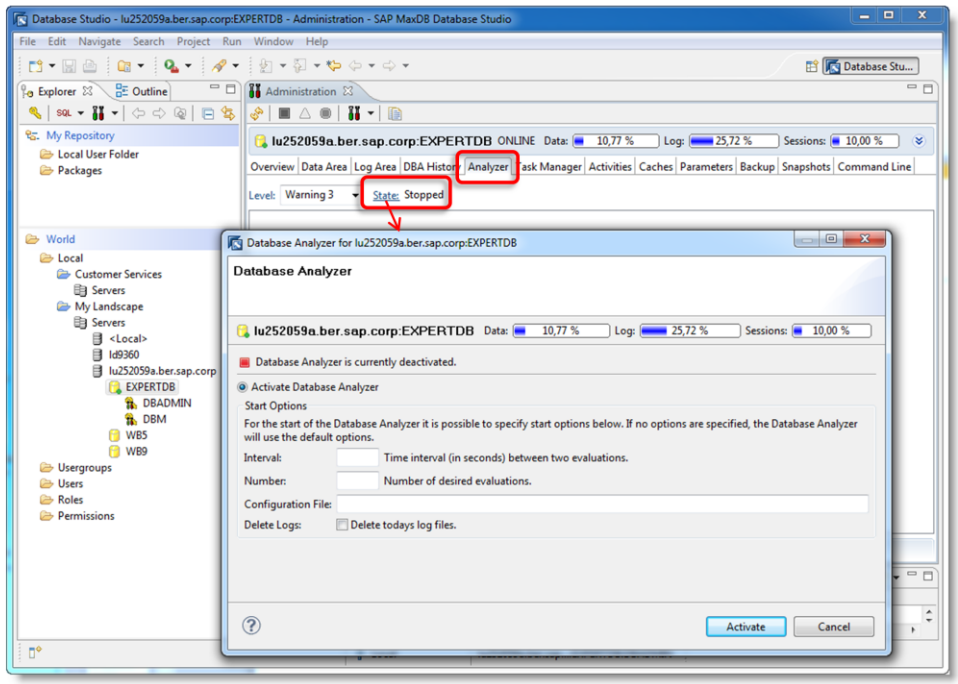
An automatic start of the Database Analyzer can be configured with transaction `DB59 → Integration Data`. Then the Database Analyzer will be started with the defined interval when the SAP system is started. This feature is only possible for the own database of the SAP system.
If a liveCache is integrated into a SAP system the implicite restart of the Database Analyzer after restart of the liveCache can be configured in the same way also via transaction DB59.
This functionality is not supported for any other remote databases.

3.1.1. Start/Stop with SAP GUI: DBACOCKPIT



The Database Analyzer is completely integrated into transaction `DBACOCKPIT/LC10: Performance → Database Analyzer → Bottlenecks`. If you want to check the current status of the Database Analyzer choose button *Determine status*.
Via button *Start Analysis* or *Stop Analysis* the Database Analyzer can be started or stopped manually. The default snapshot interval is 900 seconds. If you want to change the snapshot interval you have first to stop the Database Analyzer and define a new interval with next restart. In transaction DBACockpit snapshot intervals of 60, 120,300, 900 and 3600 seconds are supported.
An useful value during a performance analysis is between 60 and 120 seconds.

3.1.2. Start/Stop with Database Studio GUI



Database Studio GUI supports an intuitive way to start or stop the Database Analyzer. This functionality can be found on tab *Analyzer* of the administration editor . If no start options are specified the Database Analyzer will use the default options (e.g. interval of 900 seconds).
With button *Activate* the Database Analyzer will be started.

3.1.3. Start/Stop with Database Manager CLI

The default interval of 900 seconds will be used too if you start Database Analyzer with DBMCLI command ‘dban_start’ without any options. To specify another interval of collecting statistics option ‘-t’ has to be used. Additionally the count of evaluations can be specified with option ‘-t’.

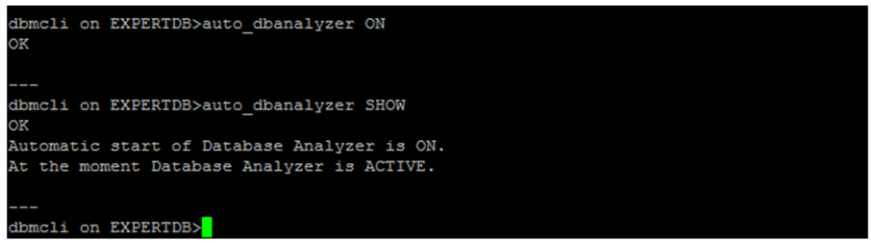
Further options of ‘dban_start’:

-f <configuration>	Name of the configuration file
-i	Deletes any existing log files for the current day
-keep <days> than <days>	Deletes all log files and directories that are older

With DBMCLI command ‘dban_stop’ the Database Analyzer will be stopped. The actual state of the Database Analyzer can be displayed with DBMCLI command ‘dban_state’.

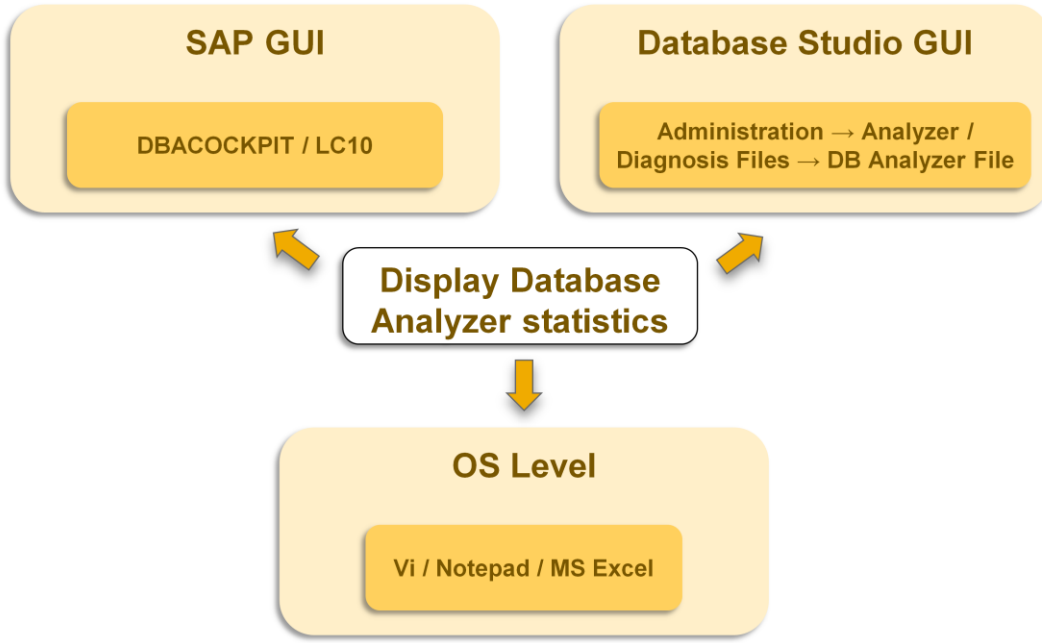
3.1.3. Start/Stop with Database Manager CLI

auto_dbanalyzer ON [<interval>] | OFF | SHOW



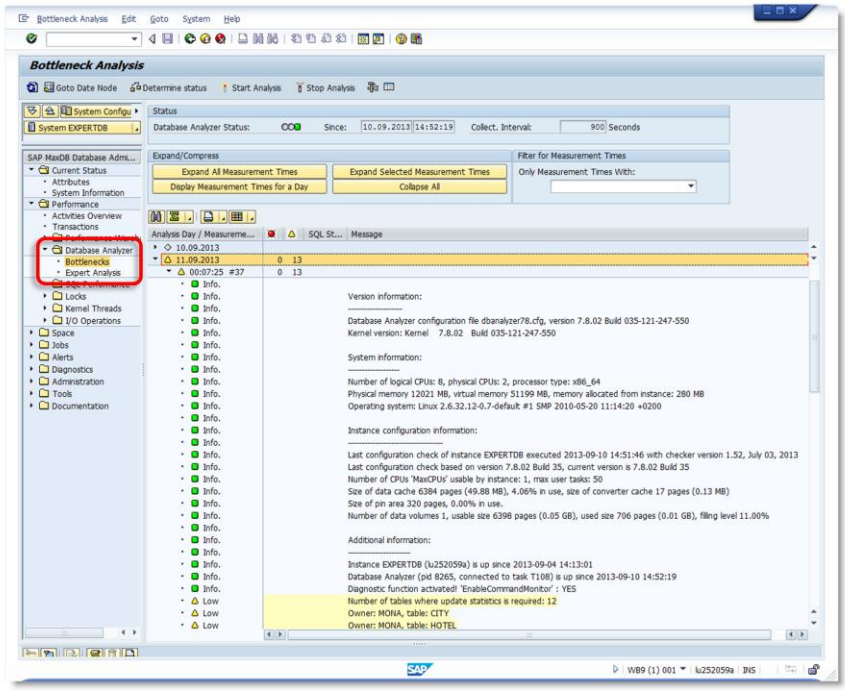
An automatic start of the Database Analyzer can be forced by using the DBMCLI command ‚auto_dbanalyzer'. This command activates the function for automatically starting Database Analyzer when the MaxDB/liveCache database is started. You can switch this function on or off in any of the operational states of the MaxDB/liveCache database.

3.2. Display Collected Statistics



There are several possibilities to display the collected statistic values of the Database Analyzer. In addition to the graphical tools SAP GUI and Database Studio GUI standard text editors (for all *.prt files) respectively MS Excel (for all *.csv files) can be used.

3.2.1. Display Collected Statistics with SAP GUI: Bottlenecks (1)



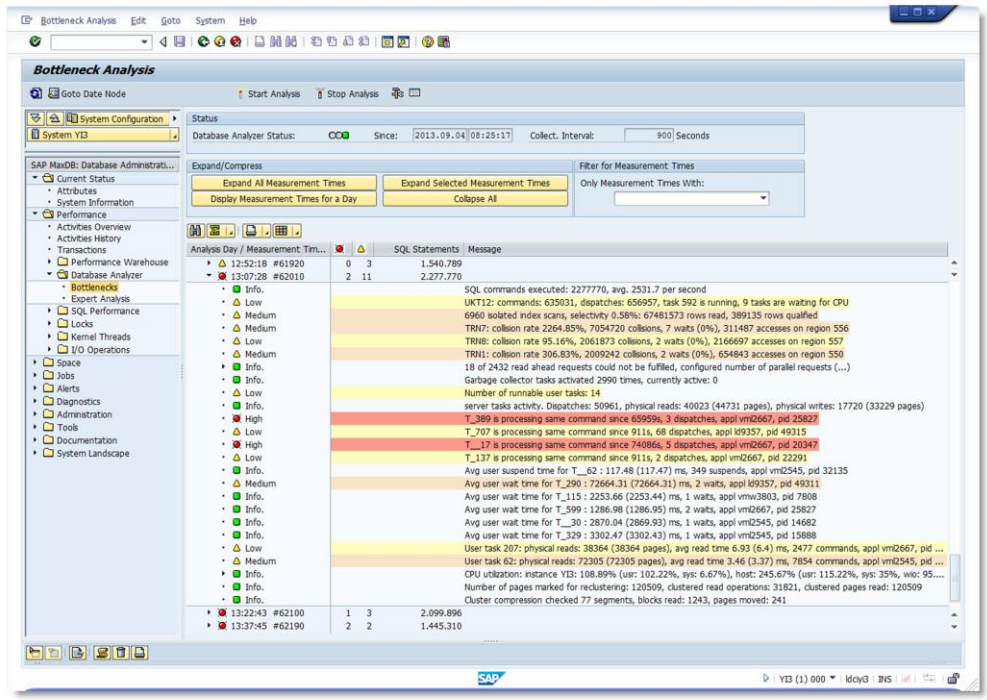
Each MaxDB performance analysis starts with the Bottleneck Analysis in transaction DBACockpit. The Information displayed under `DBACOCKPIT/LC10: Performance → Database Analyzer → Bottlenecks` are based on log file DBAN.prt which is located in subdirectory *analyzer/<YYYYMMDD>* of the rundirectory of the MaxDB/liveCache instance.

This log file starts with some general system information:

- Version information (Database Analyzer configuration file, Database Analyzer executable, database kernel)
- Hardware information (CPU, Memory, operating system)
- Database configuration information (including last executed parameter check)
- Information about required statistics update
- Missing file directory counters

This information are logged in the first interval after starting the Database Analyzer and also at the beginning of each day.

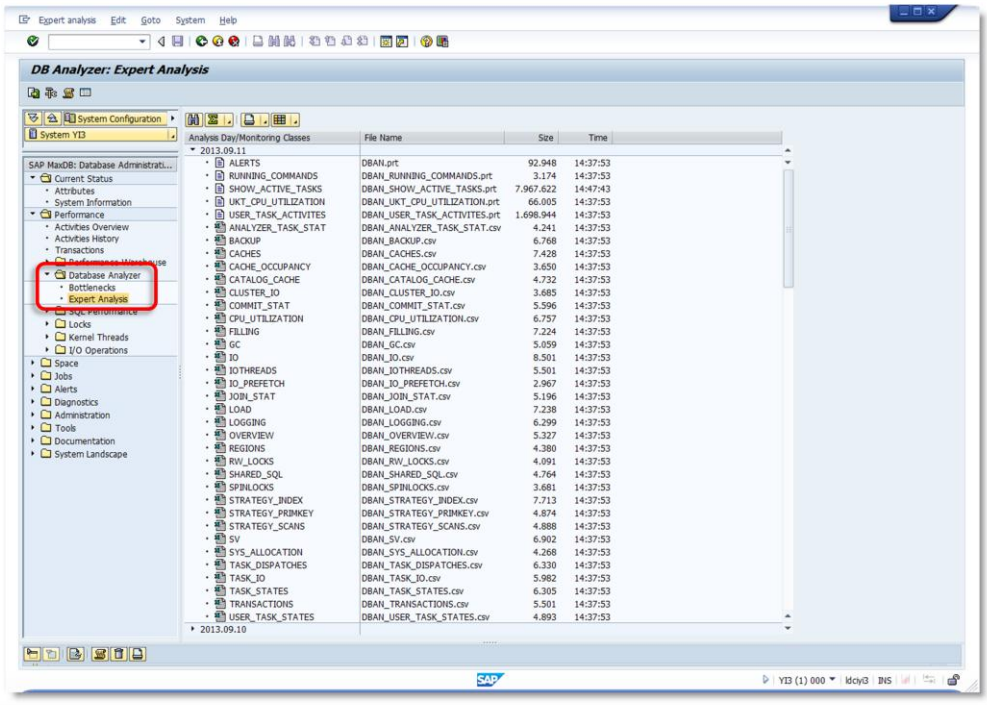
3.2.1. Display Collected Statistics with SAP GUI: Bottlenecks (2)



The other snapshots contain information about possible bottlenecks. Different warning levels will be displayed in different colors.

Note: Not each alert with priority high is pointing to a problem. The alert information should always be related to the workload of the system e.g. region collisions are not critical if there are no additional waits.

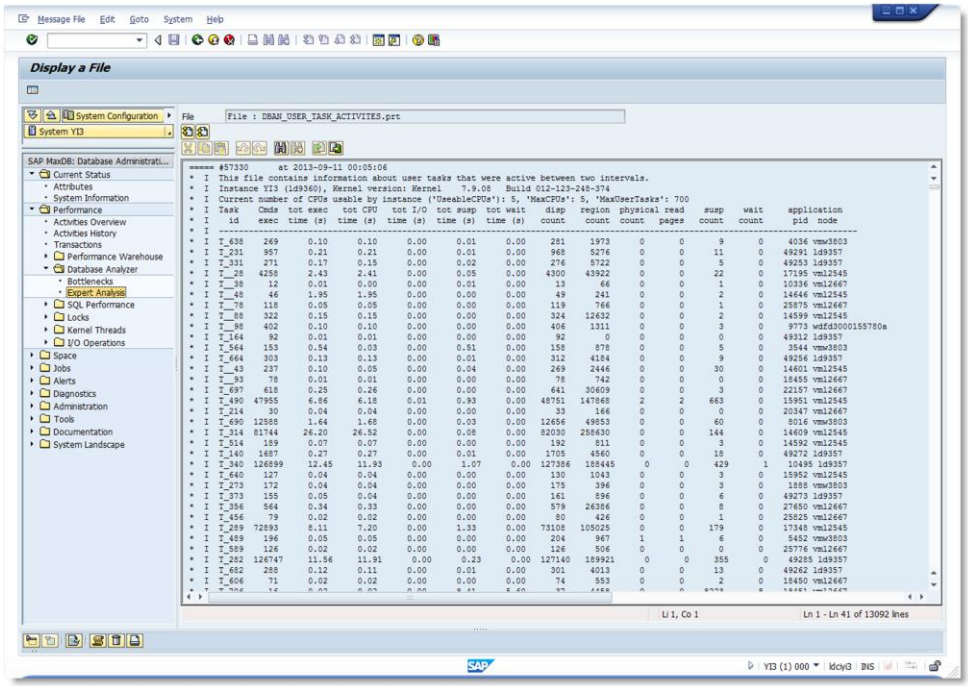
3.2.1. Display Collected Statistics with SAP GUI: Expert Analysis (1)



Detailed information are available via DBACOCKPIT/LC10: Performance → Database Analyzer → Expert Analysis. All DBAN_*.prt and DBAN_*.csv files. The expert analysis is used to start a more detailed analysis. Which file have to be checked depends on the messages in the bottleneck analysis.

To work with the expert analysis detailed knowledge about the MaxDB architecture is necessary otherwise the huge amount of statistics values cannot be interpreted correctly.

3.2.1. Display Collected Statistics with SAP GUI: Expert Analysis (2)

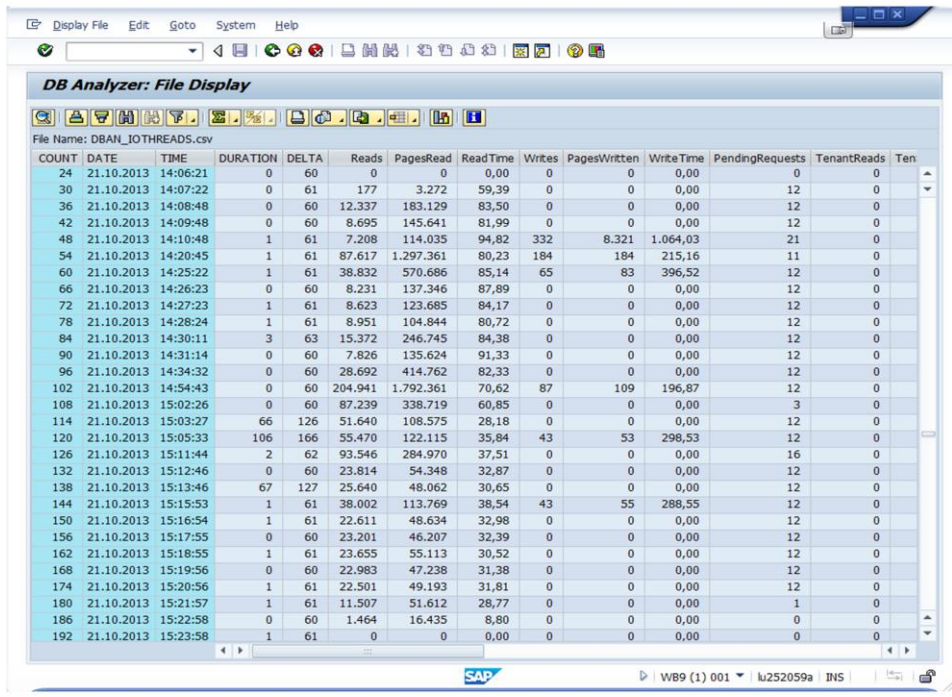


This slide shows an *.prt example for statistic values in the expert analysis - DBAN_USER_TASK_ACTIVITIES.prt
The DBAN_USER_TASK_ACTIVITIES.prt contains information about user tasks that were active between two intervals.
It does not give information about User Tasks in status connect wait.
The Database Analyzer uses the system view information (just like ‘x_cons show active’) to check if the region access count or the dispatch count has been changed related to the last interval. If this is true the task information will be logged in the new snapshot as well.
This shows us that the task is really working in the system. If a task is not shown anymore in the file DBAN_USER_TASK_ACTIVITIES.prt the task is doing nothing anymore on database level.

Statistics about servertasks and other special tasks are not listed here.

File DBAN_USER_TASK_ACTIVITIES.prt is available as of version 7.9. In versions < 7.9 use file DBAN_USER_TASKS_CMDS_EXECUTED.prt instead.

3.2.1. Display Collected Statistics with SAP GUI: Expert Analysis (3)



COUNT	DATE	TIME	DURATION	DELTA	Reads	PagesRead	ReadTime	Writes	PagesWritten	WriteTime	PendingRequests	TenantReads	Ten
24	21.10.2013	14:06:21	0	60	0	0	0,00	0	0	0,00	0	0	
30	21.10.2013	14:07:22	0	61	177	3.272	59,39	0	0	0,00	12	0	
36	21.10.2013	14:08:48	0	60	12.337	183.129	83,50	0	0	0,00	12	0	
42	21.10.2013	14:09:48	0	60	8.695	145.641	81,99	0	0	0,00	12	0	
48	21.10.2013	14:10:48	1	61	7.208	114.035	94,82	332	8.321	1.064,03	21	0	
54	21.10.2013	14:20:45	1	61	87.617	1.297.361	80,23	184	184	215,16	11	0	
60	21.10.2013	14:25:22	1	61	38.832	570.686	85,14	65	83	396,52	12	0	
66	21.10.2013	14:26:23	0	60	8.231	137.346	87,89	0	0	0,00	12	0	
72	21.10.2013	14:27:23	1	61	8.623	123.685	84,17	0	0	0,00	12	0	
78	21.10.2013	14:28:24	1	61	8.951	104.844	80,72	0	0	0,00	12	0	
84	21.10.2013	14:30:11	3	63	15.372	246.745	84,38	0	0	0,00	12	0	
90	21.10.2013	14:31:14	0	60	7.826	135.624	91,33	0	0	0,00	12	0	
96	21.10.2013	14:34:32	0	60	28.692	414.762	82,33	0	0	0,00	12	0	
102	21.10.2013	14:54:43	0	60	204.941	1.792.361	70,62	87	109	196,87	12	0	
108	21.10.2013	15:02:26	0	60	87.239	338.719	60,85	0	0	0,00	3	0	
114	21.10.2013	15:03:27	66	126	51.640	108.575	28,18	0	0	0,00	12	0	
120	21.10.2013	15:05:33	106	166	55.470	122.115	35,84	43	53	298,53	12	0	
126	21.10.2013	15:11:44	2	62	93.546	284.970	37,51	0	0	0,00	16	0	
132	21.10.2013	15:12:46	0	60	23.814	54.348	32,87	0	0	0,00	12	0	
138	21.10.2013	15:13:46	67	127	25.640	48.062	30,65	0	0	0,00	12	0	
144	21.10.2013	15:15:53	1	61	38.002	113.769	38,54	43	55	288,55	12	0	
150	21.10.2013	15:16:54	1	61	22.611	48.634	32,98	0	0	0,00	12	0	
156	21.10.2013	15:17:55	0	60	23.201	46.207	32,39	0	0	0,00	12	0	
162	21.10.2013	15:18:55	1	61	23.655	55.113	30,52	0	0	0,00	12	0	
168	21.10.2013	15:19:56	0	60	22.983	47.238	31,38	0	0	0,00	12	0	
174	21.10.2013	15:20:56	1	61	22.501	49.193	31,81	0	0	0,00	12	0	
180	21.10.2013	15:21:57	1	61	11.507	51.612	28,77	0	0	0,00	1	0	
186	21.10.2013	15:22:58	0	60	1.464	16.435	8,80	0	0	0,00	0	0	
192	21.10.2013	15:23:58	1	61	0	0	0,00	0	0	0,00	0	0	

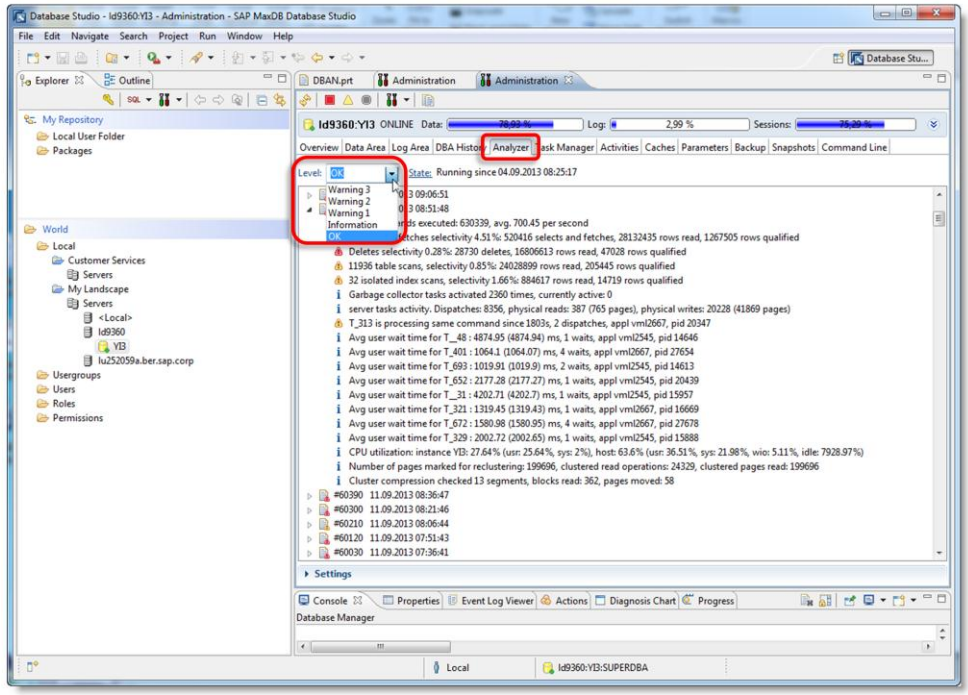
This slide shows an *.csv example for statistic values in the expert analysis - DBAN_IOTHREADS.csv

By default User Tasks do not execute the I/O itself, the I/O request is put in a queue and processed by the I/O thread. To analyze
the I/O performance the DBAN_IOTHREADS.csv is used.
Scalability of asynchroneous I/O has been improved in version 7.7 significantly. In older versions the I/O threads were directly associated with the volumes. As of version 7.7 I/O threads can send their requests to different volumes. There is a configurable number of queues per volume. It is possible to assign priorities to I/O requests. Tasks don‘t have to wait for the result of the I/O but can send the request asynchroneously and continue their work.

In this example we see that we have
- huge number of read IO (*PagesRead*)
- very bad I/O times for reading (*ReadTime* in ms)
- Write I/O especially every 10 minutes (*PagesWritten*) – could be savepoint
- Very bad I/O times for writing (*WriteTime* in ms)
- Dev threads got a bottleneck (*PendingRequests* > 0), the Dev threads could not write/read the data fast enough to avoid any wait situation in the DEV threads.

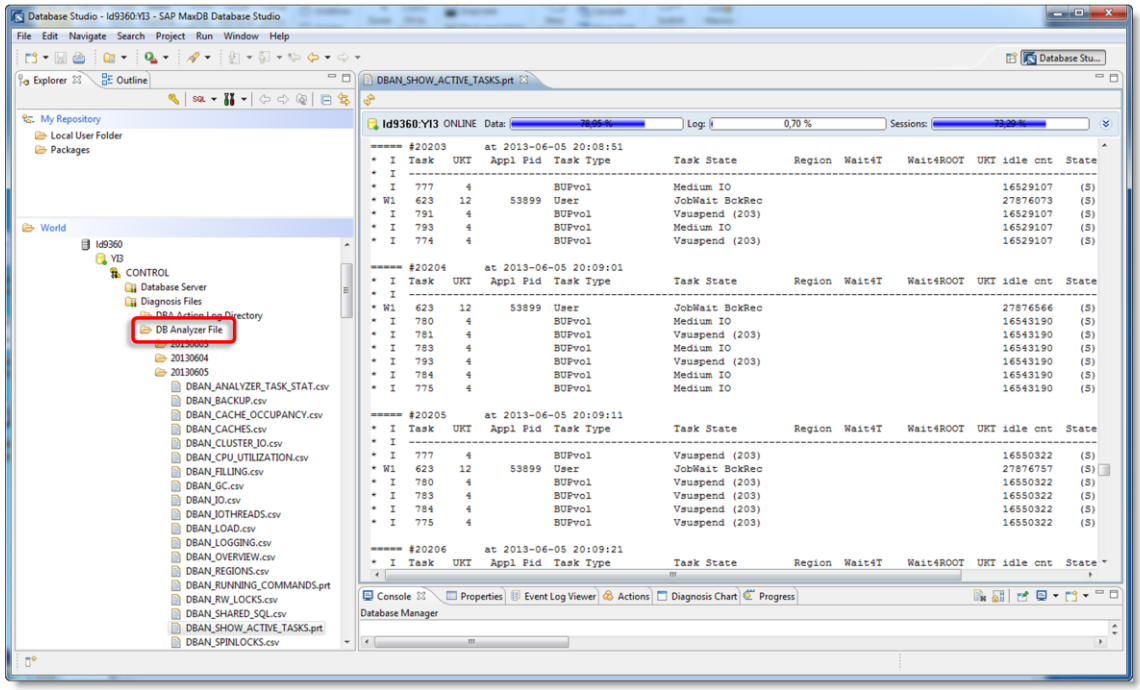
- In this example we should check the database disk configuration and the disk performance on hardware level.

3.2.2. Display Collected Statistics with DB Studio GUI (1)



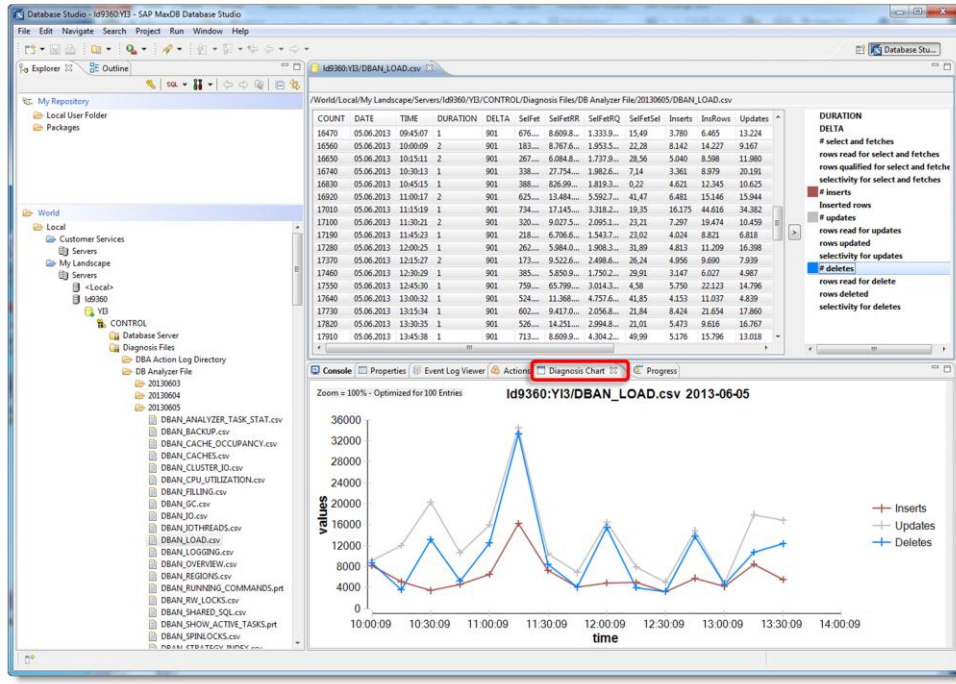
On tab *Analyzer* of the administration editor the displayed warning levels can be selected and are highlighted with different icons.
This information is based on file dban.prt and corresponds to the DBACockpit/LC10 menue *Bottlenecks*.

3.2.2. Display Collected Statistics with DB Studio GUI (2)



All Database Analyzer log files DBAN_*.prt and DBAN_*.csv are available over the Explorer tree entry *DB Analyzer File* (you have to select context menu *Extended File List* of *Diagnosis Files* to see this entry). Only DBAN_*.prt files can be displayed in an usable way, this example DBAN_SHOW_ACTIVE_TASKS.prt.

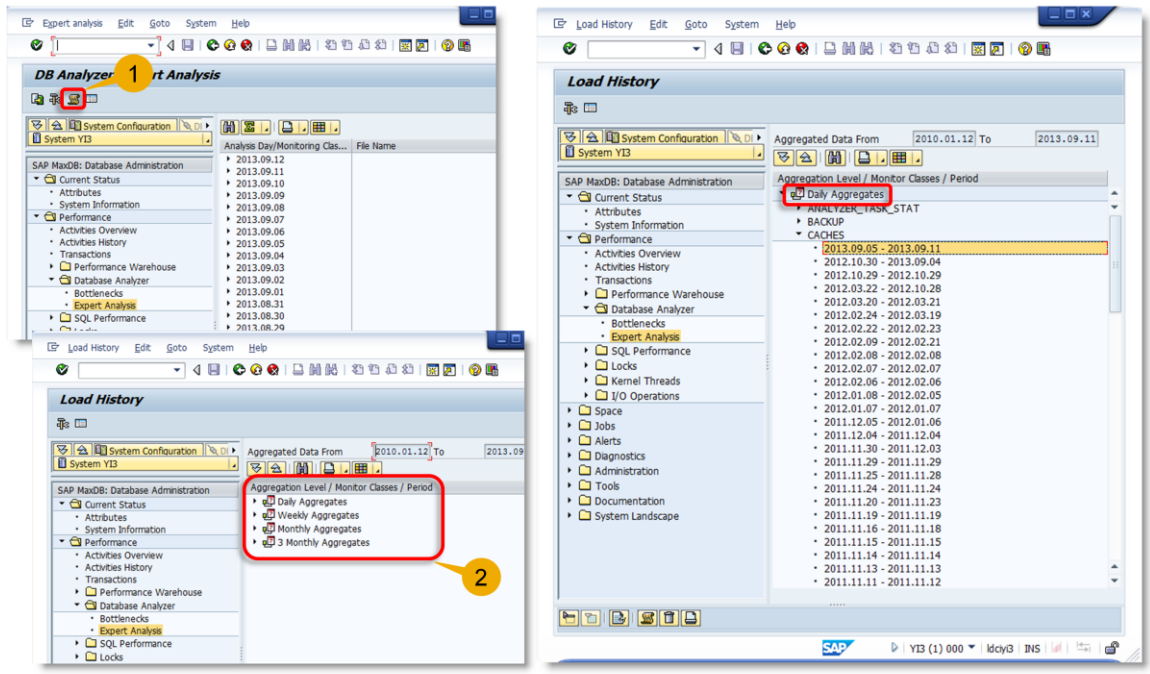
3.2.2. Display Collected Statistics with DB Studio GUI (3)



A new Database Studio GUI feature will allow to display DBAN_*.csv files in a graphical way. This new Database Studio GUI feature is in progress. Via *Diagnosis Charts* one or more columns of a loaded DBAN_*.csv file will be presentable in one of the next Database Studio GUI version.

In this example we see the changes done on database level (number of Inserts, Deletes and Updates) in a specified time period.
Such statistics are used to check the system load caused by application side. With this graphical layout you can easy see when was the peak and which SQL commands (in this case deletes and updates) caused the system load.

3.3. Statistics Aggregation (1)

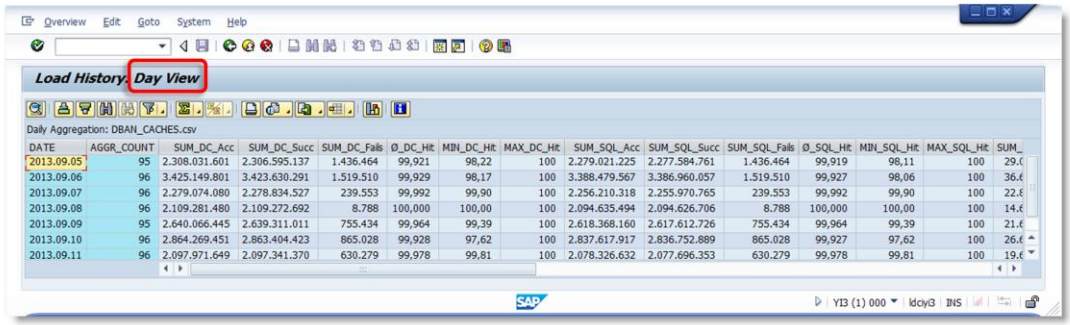


The statistics aggregation functionality allows to compare the statistic values of different days, weeks or months. This feature is implemented in ABAP stack only, not in Database Studio GUI.
The SAP system must be configured that the statistic values are aggregated. How to do this can be found in chapter 3.4.

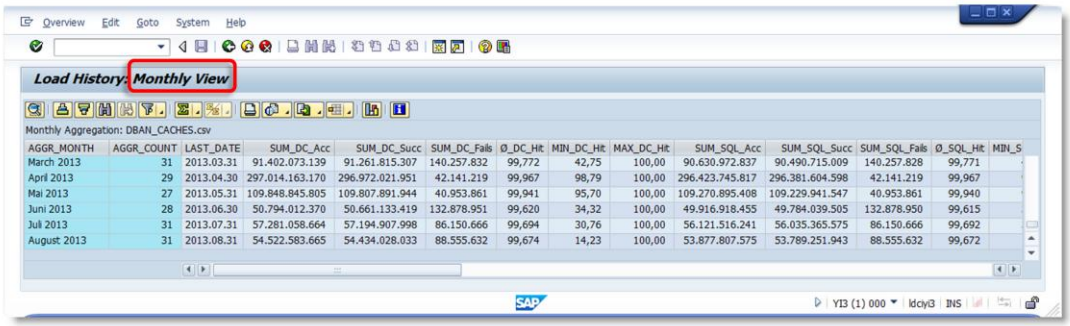
Via button *Aggregated Performance Data (1)* different aggregation levels can be selected (2).

You use this functionality if the system response time has changed after some adiministrative tasks have been done e.g after a software upgrade, after a hardware change, etc.

3.3. Statistics Aggregation (2)



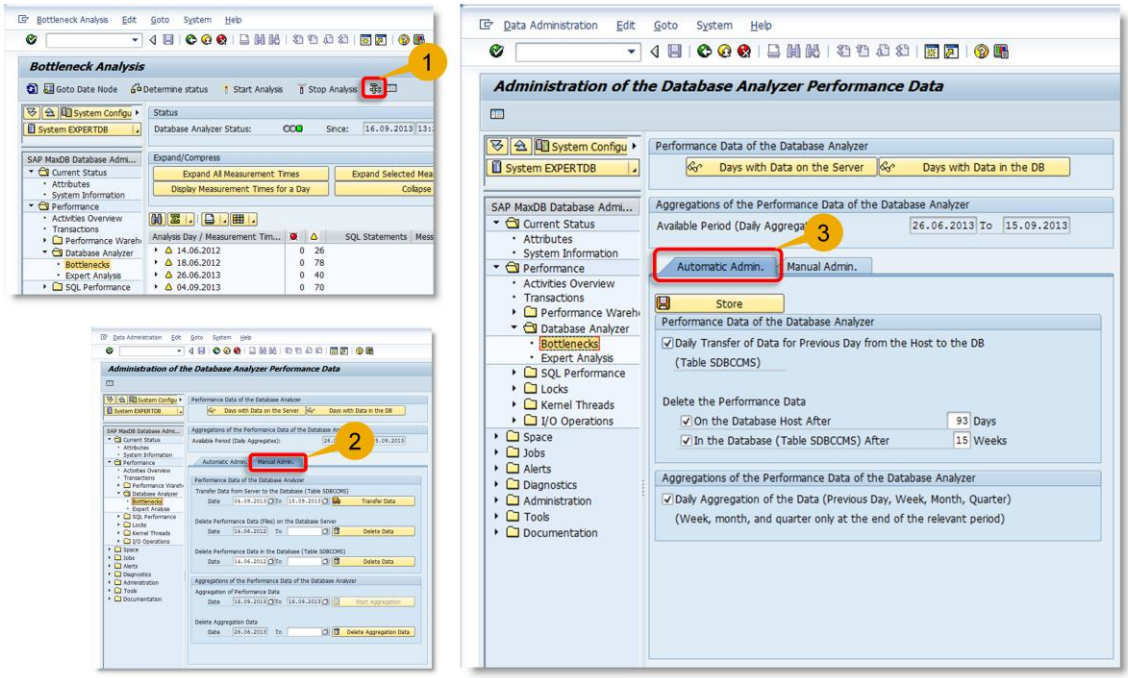
DATE	AGGR_COUNT	SUM_DC_Acc	SUM_DC_Succ	SUM_DC_Fails	Ø_DC_Hit	MIN_DC_Hit	MAX_DC_Hit	SUM_SQL_Acc	SUM_SQL_Succ	SUM_SQL_Fails	Ø_SQL_Hit	MIN_SQL_Hit	MAX_SQL_Hit	SUM_
2013.09.05	95	2.308.031.601	2.306.595.137	1.436.464	99,921	98,22	100	2.279.021.225	2.277.584.761	1.436.464	99,919	98,11	100	29.(
2013.09.06	96	3.425.149.801	3.423.630.291	1.519.510	99,929	98,17	100	3.388.479.567	3.386.960.057	1.519.510	99,927	98,06	100	36.(
2013.09.07	96	2.279.074.080	2.278.834.527	239.553	99,992	99,90	100	2.256.210.318	2.255.970.765	239.553	99,992	99,90	100	22.(
2013.09.08	96	2.109.281.480	2.109.272.692	8.788	100,000	100,00	100	2.094.635.494	2.094.626.706	8.788	100,000	100,00	100	14.(
2013.09.09	95	2.640.066.445	2.639.311.011	755.434	99,964	99,39	100	2.618.368.160	2.617.612.726	755.434	99,964	99,39	100	21.(
2013.09.10	96	2.864.269.451	2.863.404.423	865.028	99,928	97,62	100	2.837.617.917	2.836.752.889	865.028	99,927	97,62	100	26.(
2013.09.11	96	2.097.971.649	2.097.341.370	630.279	99,978	99,81	100	2.078.326.632	2.077.696.353	630.279	99,978	99,81	100	19.(



AGGR_MONTH	AGGR_COUNT	LAST_DATE	SUM_DC_Acc	SUM_DC_Succ	SUM_DC_Fails	Ø_DC_Hit	MIN_DC_Hit	MAX_DC_Hit	SUM_SQL_Acc	SUM_SQL_Succ	SUM_SQL_Fails	Ø_SQL_Hit	MIN_S
March 2013	31	2013.03.31	91.402.073.139	91.261.815.307	140.257.832	99,772	42,75	100,00	90.630.972.837	90.490.715.009	140.257.828	99,771	
April 2013	29	2013.04.30	297.014.163.170	296.972.021.951	42.141.219	99,967	98,79	100,00	296.423.745.817	296.381.604.598	42.141.219	99,967	
Mai 2013	27	2013.05.31	109.848.845.805	109.807.891.944	40.953.861	99,941	95,70	100,00	109.270.895.408	109.229.941.547	40.953.861	99,940	
Juni 2013	28	2013.06.30	50.794.012.370	50.661.133.419	132.878.951	99,620	34,32	100,00	49.916.918.455	49.784.039.505	132.878.950	99,615	
Juli 2013	31	2013.07.31	57.281.058.664	57.194.907.998	86.150.666	99,694	30,76	100,00	56.121.516.241	56.035.365.575	86.150.666	99,692	
August 2013	31	2013.08.31	54.522.583.665	54.434.028.033	88.555.632	99,674	14,23	100,00	53.877.807.575	53.789.251.943	88.555.632	99,672	

One example of statistics aggregation on the daily and monthly base (DBAN_CACHES.csv).

3.4. Statistics Administration



The administration of the Database Analyzer statistics is implemented via button *Administration of Performance Data (1)* which is available in both menues *Bottlenecks* and *Expert Analysis*. Manual (2) or automatic administration (3) can be selected via tab.

The automatic adminsitration

- defines if the performance data is additionally stored in the database (to speed up the reading time of files)
- how long these data stay in the database and on the database host
- activates the aggregation of data

The Manual administration allows in more detailes when data will be aggregated and deleted.

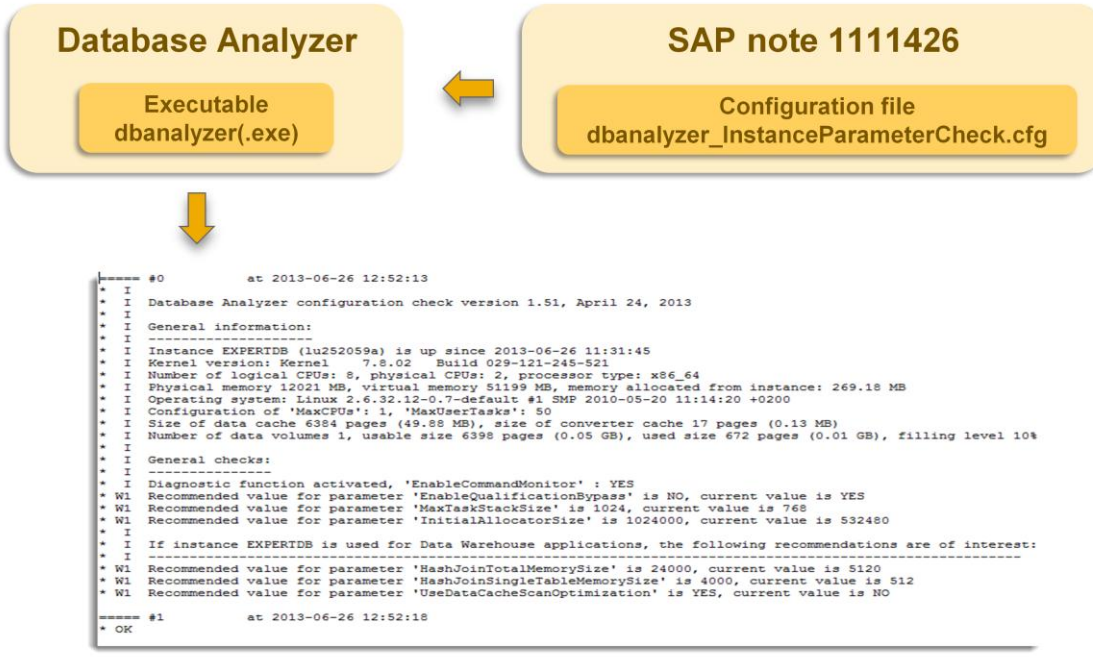
SAP recommends to activate the aggregation of statistics via automatic administration.

This functionality is not available in Database Studio GUI.

Agenda

1. Introduction
2. Functional Chain
3. Ways to Manage Database Analyzer
4. Parameter Check with Database Analyzer
5. Expert Analysis
6. Additional Useful Information

4. Parameter Check with Database Analyzer (1)



MaxDB/liveCache parameter check is embedded into the Database Analyzer. Use this Database Analyzer feature to check if the configuration of your MaxDB/liveCache database corresponds to the current SAP recommendations.

The parameter check should be executed after each MaxDB/liveCache software upgrade. Different recommendations may be relevant for different database versions.

The parameter check uses a **special Database Analyzer configuration file** (only one file for all MaxDB/liveCache versions). This special configuration file is attached to **SAP note 1111426**. As this file is regulary updated, you must download it always before a new check.

4. Parameter Check with Database Analyzer (2)

```
===== #0          at 2013-06-26 12:52:13
*  I
*  I  Database Analyzer configuration check version 1.51, April 24, 2013
*  I
*  I  General information:
*  I  --------------------
*  I  Instance EXPERTDB (lu252059a) is up since 2013-06-26 11:31:45
*  I  Kernel version: Kernel    7.8.02   Build 029-121-245-521
*  I  Number of logical CPUs: 8, physical CPUs: 2, processor type: x86_64
*  I  Physical memory 12021 MB, virtual memory 51199 MB, memory allocated from instance: 269.18 MB
*  I  Operating system: Linux 2.6.32.12-0.7-default #1 SMP 2010-05-20 11:14:20 +0200
*  I  Configuration of 'MaxCPUs': 1, 'MaxUserTasks': 50
*  I  Size of data cache 6384 pages (49.88 MB), size of converter cache 17 pages (0.13 MB)
*  I  Number of data volumes 1, usable size 6398 pages (0.05 GB), used size 672 pages (0.01 GB), filling level 10%
*  I
*  I  General checks:
*  I  ---------------
*  I  Diagnostic function activated, 'EnableCommandMonitor' : YES
* W1  Recommended value for parameter 'EnableQualificationBypass' is NO, current value is YES
* W1  Recommended value for parameter 'MaxTaskStackSize' is 1024, current value is 768
* W1  Recommended value for parameter 'InitialAllocatorSize' is 1024000, current value is 532480
*  I
*  I  If instance EXPERTDB is used for Data Warehouse applications, the following recommendations are of interest:
*  I  ------------------------------------------------------------------------------------------------------------
* W1  Recommended value for parameter 'HashJoinTotalMemorySize' is 24000, current value is 5120
* W1  Recommended value for parameter 'HashJoinSingleTableMemorySize' is 4000, current value is 512
* W1  Recommended value for parameter 'UseDataCacheScanOptimization' is YES, current value is NO

===== #1          at 2013-06-26 12:52:18
* OK
```


The database instance must be in operational state ONLINE when you start the parameter check tool. Perform the automatic check as SYSDBA user (e.g. dbadmin)

dbanalyzer –d EXPERTDB –n <server> –u dbadmin,<password>
–f dbanalyzer_instanceParametercheck.cfg -o <temp_directory> -i –c 1 –t 1,1

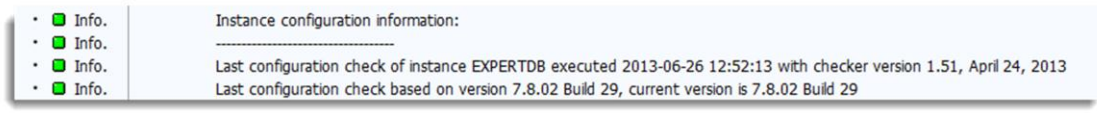
With parameter
-i the output directory will be cleaned up
-c output will be send to screen as well
-t 1,1 only 1 snapshot in an interval of one second

Analyze the screen output or the file *<temp_directory>/<YYYYMMDD>/DBAN.prt*. Important are all messages that are marked with "* W1 to * W3"

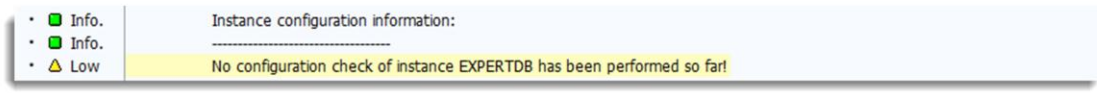
The following checks are executed:
- general parameters
- parameters which influence the I/O performance
- optimizer parameters
- special liveCache parameters
- additional checks
  - do corrupt indexes exist?
  - is the database kernel trace activated?
  - do tables exist which do not have any file directory counters?
  - is logging activated and autooverwrite deactivated?
  - does the size of the IO Buffer Cache correspond to the SAP recommendation, which is 2% of the configured volume size for UNICODE systems and 1% for NON-UNICODE systems?

4. Parameter Check with Database Analyzer (3)

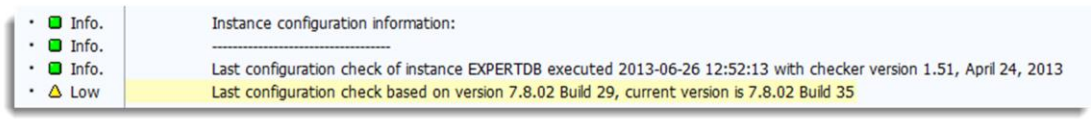
a) Check has already been performed for the current version:



b) Check has never been performed:



c) Check has not yet been performed for the current version:

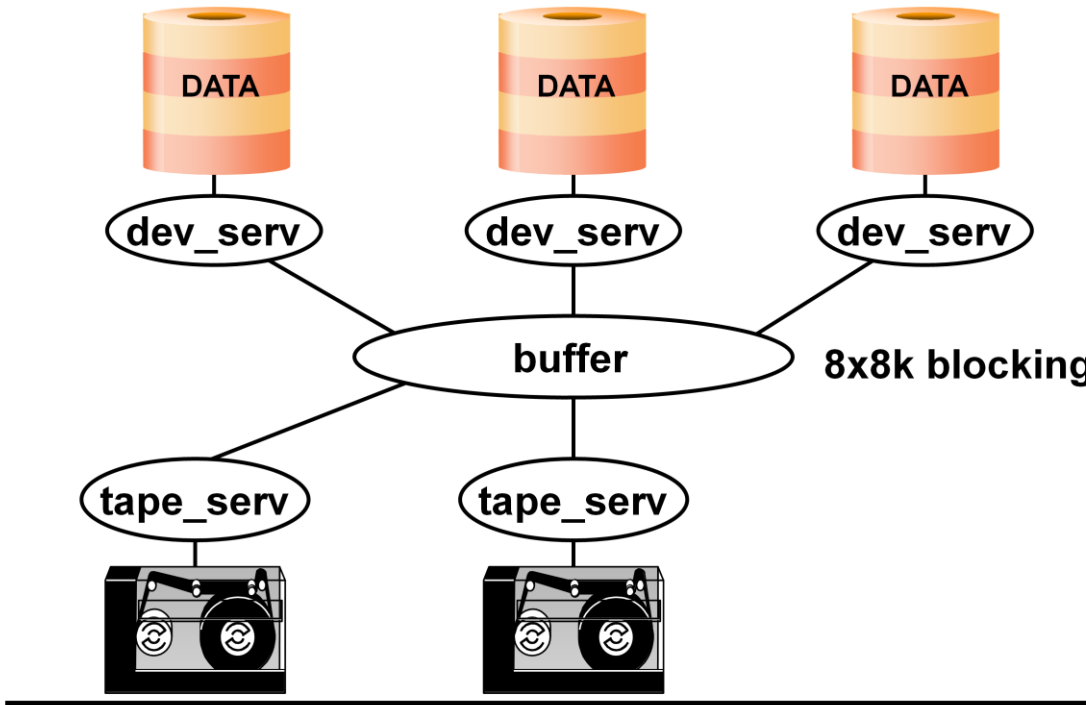


The Database Analyzer analyzes whether the configuration of your database was checked. If this is the case, the Database Analyzer also determines when the check was performed and which version was used to do this.

Agenda

1. Introduction
2. Functional chain
3. Ways to manage Database Analyzer
4. Parameter check with Database Analyzer
5. Expert analysis
6. Useful Information Resources

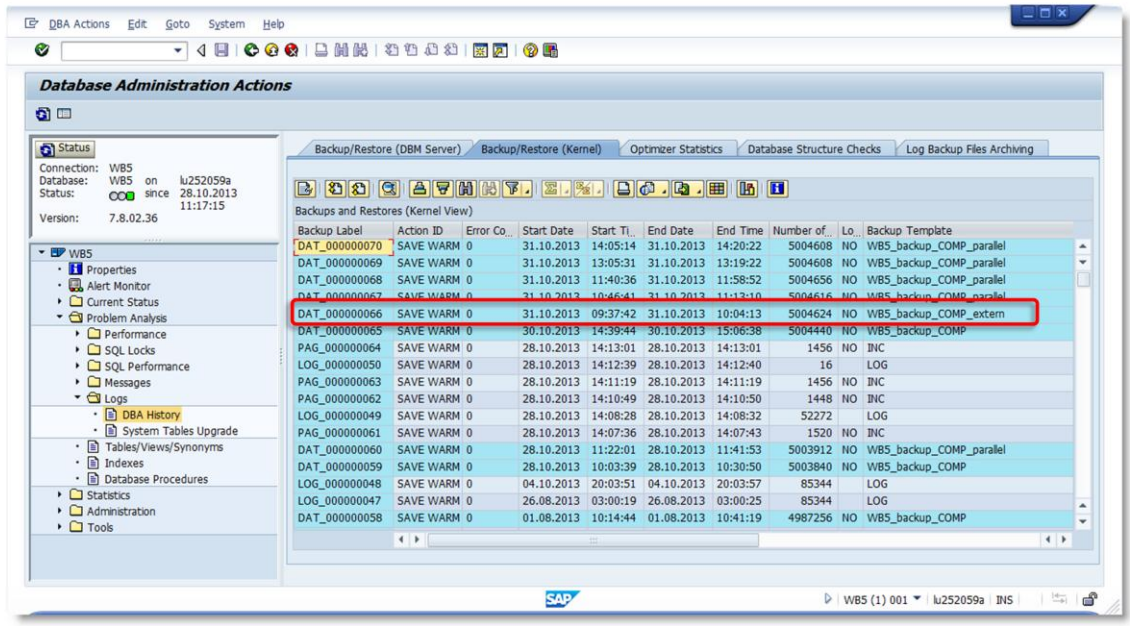
5.1. Optimize Runtime of Data Backup: Introduction



This illustration depicts a data transfer from the data volumes to the backup media. Each volume has a task that puts the 64 KB units into a buffer. One task per backup device reads the blocks from the buffer and stores them on the backup medium.

The limits of this process are posed either by the access speed of the data volumes, the writing performance of the backup devices or the transport layer (e.g. network) between the database server and the backup devices. As long as these limits are not reached, the process scales with any other backup device in parallel operation.

5.1. Optimize Runtime of Data Backup: Backup (1)



In this section the runtime of a data backup will be analyzed. It helps to recognize and remove bottlenecks.

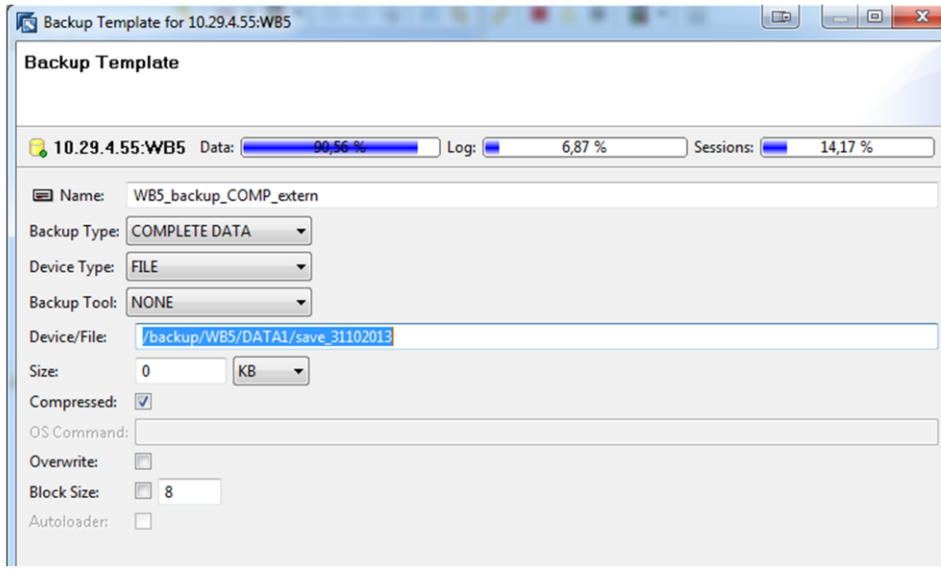
To be able to execute a runtime analysis of a backup, the Database Analyzer must be active during the backup. It's recommended to change the default setting of the Database Analyzer interval from 900 seconds to 60 seconds.

In addition, you have to activate the time measurement explicitly. This also applies to the SAP MaxDB version 7.8 or higher.

The measurement of time is activated using the DBA Cockpit as follows: DBA Cockpit (transaction DBACOCKPIT) -> *Performance -> Kernel Threads -> Task Manager -> Activate* the DB measurement of time.

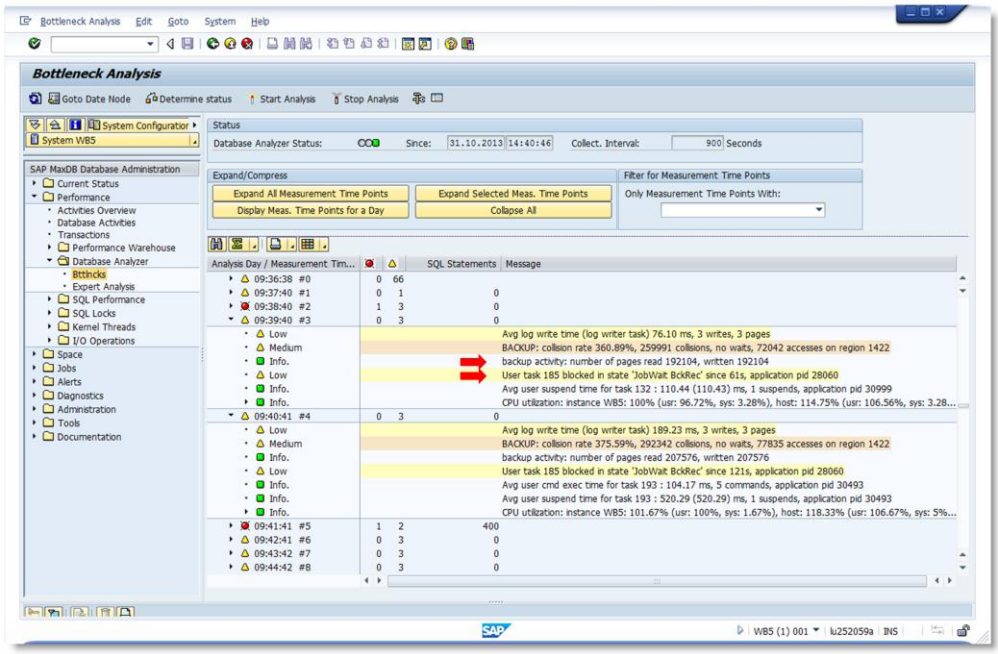
We check the runtime of a backup in DBACockpit -> *diagnostics -> Messages -> DBA History -> Backup/restore Kernel*. The Data Backup which is analyzed was executed at 31th of October from 9:37 am to 10:04 am -> duration 27 minutes.

5.1. Optimize Runtime of Data Backup: Single Medium



The backup was executed on backup template *WB5_backup_COMP_extern*.
The Template definition tells us that a compressed backup is created which writes into one file.

5.1. Optimize Runtime of Data Backup: Bottlenecks



The analysis starts with the Database Analyzer bottleneck analysis (file DBAN.prt).

Goto transaction DBACOCKPIT/LC10: *Performance → Database Analyzer → Bottlenecks*

Choose the date (here 31.10.2013) on which the backup to be analyzed was executed and go to the period in which the backup was active.
(from 9:37 am to 10:04 am)

As soon as a backup is active, you can find the following entry for the duration of the backup in the file DBAN.prt:

User task 185 blocked in state 'JobWait BckRec' since .....

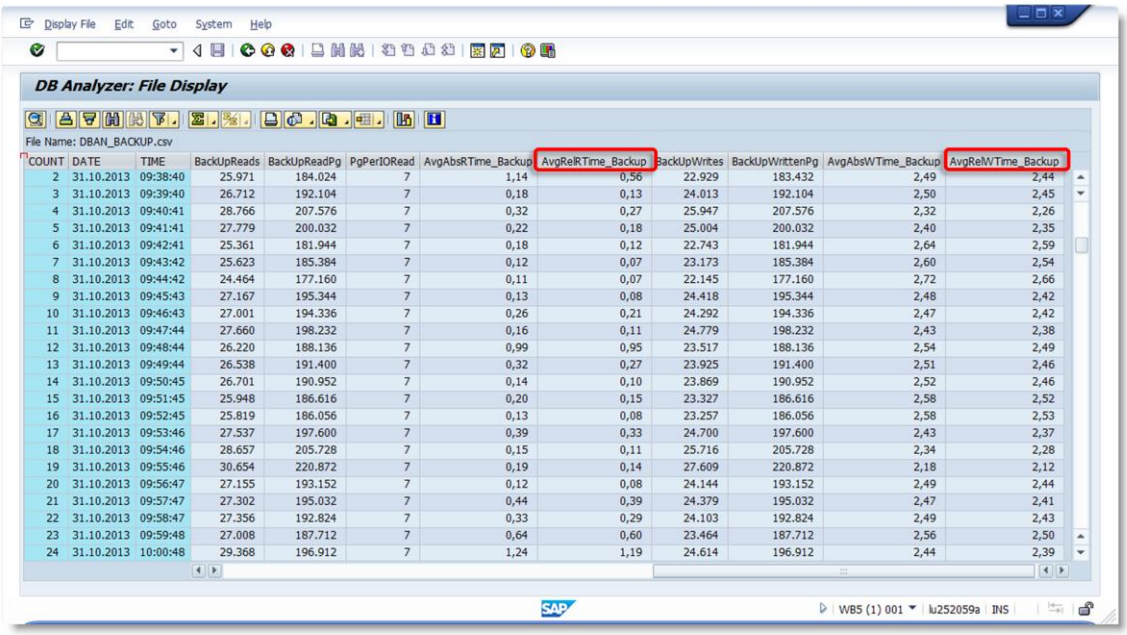
The UserTask itself does not execute the I/O reads – asynchronus I/O is done via ServerTasks.

During the backup, you find the Database Analyzer information of how many pages were read and written by the server tasks in each interval. This gives you a first hint concerning the throughput of the backup.

In this example:

In an interval of 60 seconds - *backup activity: number of pages read 192104, written 192104*

5.1. Optimize Runtime of Data Backup: DBAN_BACKUP.csv



COUNT	DATE	TIME	BackUpReads	BackUpReadPg	PgPerIORead	AvgAbsRTime_Backup	AvgRelRTime_Backup	BackUpWrites	BackUpWrittenPg	AvgAbsWTime_Backup	AvgRelWTime_Backup
2	31.10.2013	09:38:40	25.971	184.024	7	1,14	0,56	22.929	183.432	2,49	2,44
3	31.10.2013	09:39:40	26.712	192.104	7	0,18	0,13	24.013	192.104	2,50	2,45
4	31.10.2013	09:40:41	28.766	207.576	7	0,32	0,27	25.947	207.576	2,32	2,26
5	31.10.2013	09:41:41	27.779	200.032	7	0,22	0,18	25.004	200.032	2,40	2,35
6	31.10.2013	09:42:41	25.361	181.944	7	0,18	0,12	22.743	181.944	2,64	2,59
7	31.10.2013	09:43:42	25.623	185.384	7	0,12	0,07	23.173	185.384	2,60	2,54
8	31.10.2013	09:44:42	24.464	177.160	7	0,11	0,07	22.145	177.160	2,72	2,66
9	31.10.2013	09:45:43	27.167	195.344	7	0,13	0,08	24.418	195.344	2,48	2,42
10	31.10.2013	09:46:43	27.001	194.336	7	0,26	0,21	24.292	194.336	2,47	2,42
11	31.10.2013	09:47:44	27.660	198.232	7	0,16	0,11	24.779	198.232	2,43	2,38
12	31.10.2013	09:48:44	26.220	188.136	7	0,99	0,95	23.517	188.136	2,54	2,49
13	31.10.2013	09:49:44	26.538	191.400	7	0,32	0,27	23.925	191.400	2,51	2,46
14	31.10.2013	09:50:45	26.701	190.952	7	0,14	0,10	23.869	190.952	2,52	2,46
15	31.10.2013	09:51:45	25.948	186.616	7	0,20	0,15	23.327	186.616	2,58	2,52
16	31.10.2013	09:52:45	25.819	186.056	7	0,13	0,08	23.257	186.056	2,58	2,53
17	31.10.2013	09:53:46	27.537	197.600	7	0,39	0,33	24.700	197.600	2,43	2,37
18	31.10.2013	09:54:46	28.657	205.728	7	0,15	0,11	25.716	205.728	2,34	2,28
19	31.10.2013	09:55:46	30.654	220.872	7	0,19	0,14	27.609	220.872	2,18	2,12
20	31.10.2013	09:56:47	27.155	193.152	7	0,12	0,08	24.144	193.152	2,49	2,44
21	31.10.2013	09:57:47	27.302	195.032	7	0,44	0,39	24.379	195.032	2,47	2,41
22	31.10.2013	09:58:47	27.356	192.824	7	0,33	0,29	24.103	192.824	2,49	2,43
23	31.10.2013	09:59:48	27.008	187.712	7	0,64	0,60	23.464	187.712	2,56	2,50
24	31.10.2013	10:00:48	29.368	196.912	7	1,24	1,19	24.614	196.912	2,44	2,39

The file DBAN_BACKUP.csv contains detailed information about the backup procedure. Go to the period in which the backup was active.

Here, you will find, among others, information about how many pages were read in an interval (*BackUpReadPg*), how many read I/O accesses (*BackUpReads*) were executed in an interval, how many pages were written in an interval (*BackUpWrittenPg*), and how many write I/Os (*BackUpWrites*) were executed.

If the measurement of time has been activated, you can read the time used for reading the data in the columns *AvgAbsRTime_Backup* and *AvgRelRTime_Backup*. The columns *AvgAbsWTime_Backup* and *AvgRelWTime_Backup* display the times that were required for writing to the backup medium.

- *AvgAbsRTime_Backup/AvgAbsWTime_Backup:*
  The absolute time is the total time that a task needs for the read/write until the CPU is assigned to the task again. In other words, even if the actual I/O has already been executed, this time can still increase when the task is in the runqueue and has to wait for the CPU assignment.

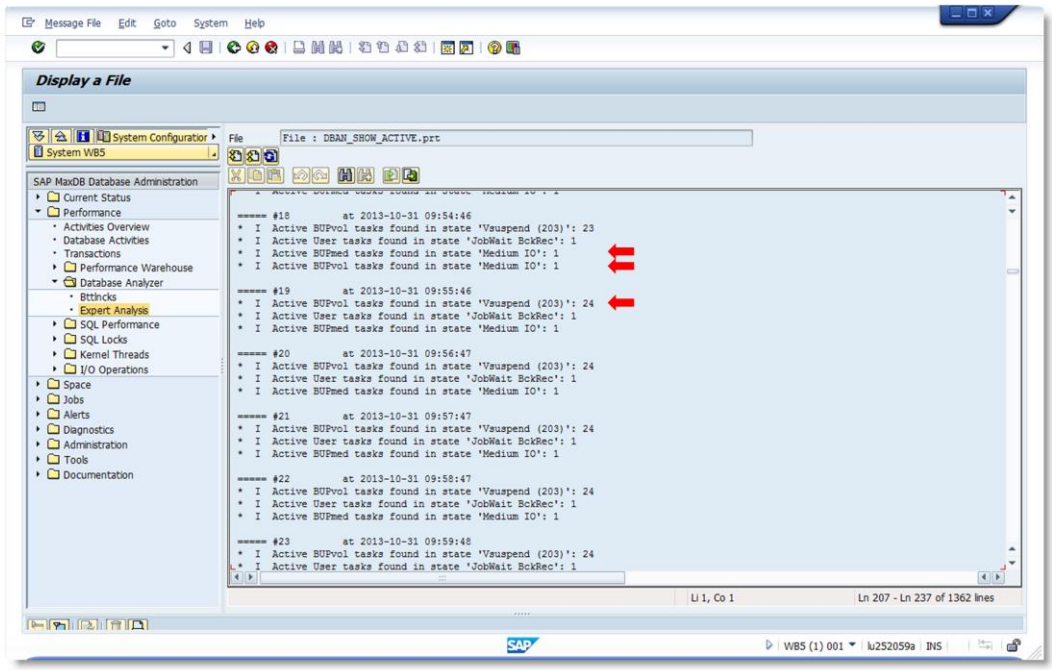
- *AvgRelRTime_Backup/AvgRelWTime_Backup:*
  The relative time is the time that a task spends exclusively waiting for the execution of the I/O. As a rule, the relative time is therefore lower than the absolute time.

A big discrepancy between the relative and the absolute times points to a bottleneck in the thread (UKT). In a thread, only one task can use the CPU at any one time. In this case, you can resolve the bottleneck by distributing the server tasks to different threads (UKTs). For detailed information, see SAP note 1672994.

In this case, the I/O times measured for reading the data from volumes (*AvgRelRTime_Backup*) and the values for writing the data to the backup medium (AvgRelWTime_Backup) are very good. For good I/O times, the I/O must be lower than 10 msec.

However, poor I/O times do not necessarily mean a bottleneck for the runtime of a backup. In general, during backup you have more data volumes from which data is read in parallel by the server tasks than backup media that are written to.

5.1. Optimize Runtime of Data Backup: DBAN_SHOW_ACTIVE.prt



In each case, you should also analyze the file DBAN_SHOW_ACTIVE.prt (> 7.8: DBAN_SHOW_ACTIVE_TASKS.prt). Go to the period in which the backup was active.

This file gives information about which tasks have been active in an interval. It only contains entries if the Database Analyzer interval is less than or equal 300 seconds.

In principle, several snapshots are used for the analysis of this file. This gives you an overview over the process of the backup.

BUPmed tasks - Activities of tasks that write to the backup medium

BUPvol tasks - Activities of tasks responsible for reading the data from the volumes

In the snapshots of 09:54:46 and 09:55:46 we can see that only one task is busy with a write I/O to the backup medium (*BUPmed*). This suggests that no parallel medium is defined here.

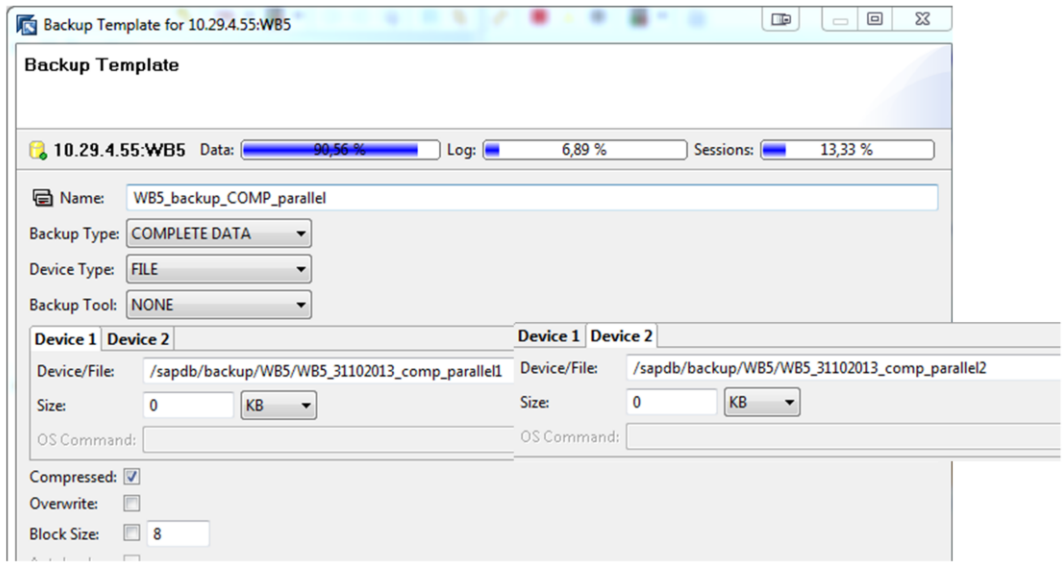
In the snapshot at 09:54:46, 1 server task is busy with reading from the data volumes (*BUPvol 'Medium IO'*).

The user task is waiting for the end of the backup (*User JobWait BckRec*).

The second snapshot at 09:55:46 is interesting if you concentrate on the server tasks that read from the data volumes. All server tasks (24) that have the read request are waiting (*Vsuspend*). The reason for the wait situation is that one write task is not fast enough to write the data from the ring buffer to the backup medium. Only once space becomes free again in the ring buffer the reading server task can continue reading the data.

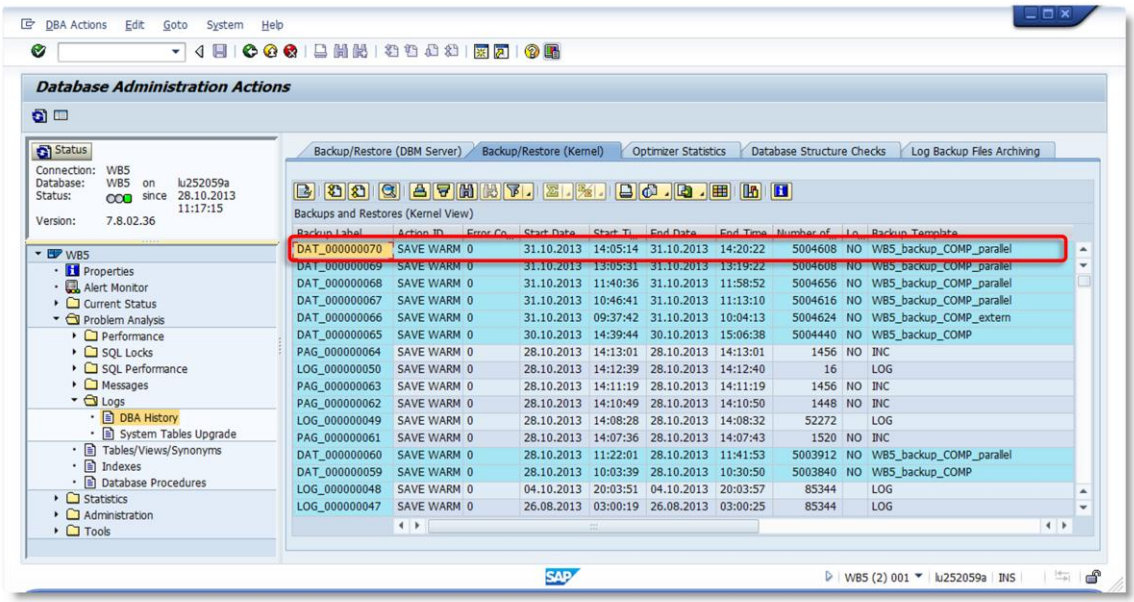
The bottleneck is not due to the I/O but due to the throughput of data written to the backup medium. The I/O times for this are good (less than 3 ms).
You can speed up the backup by installing a parallel backup medium.

5.1. Optimize Runtime of Data Backup: Parallel Medium



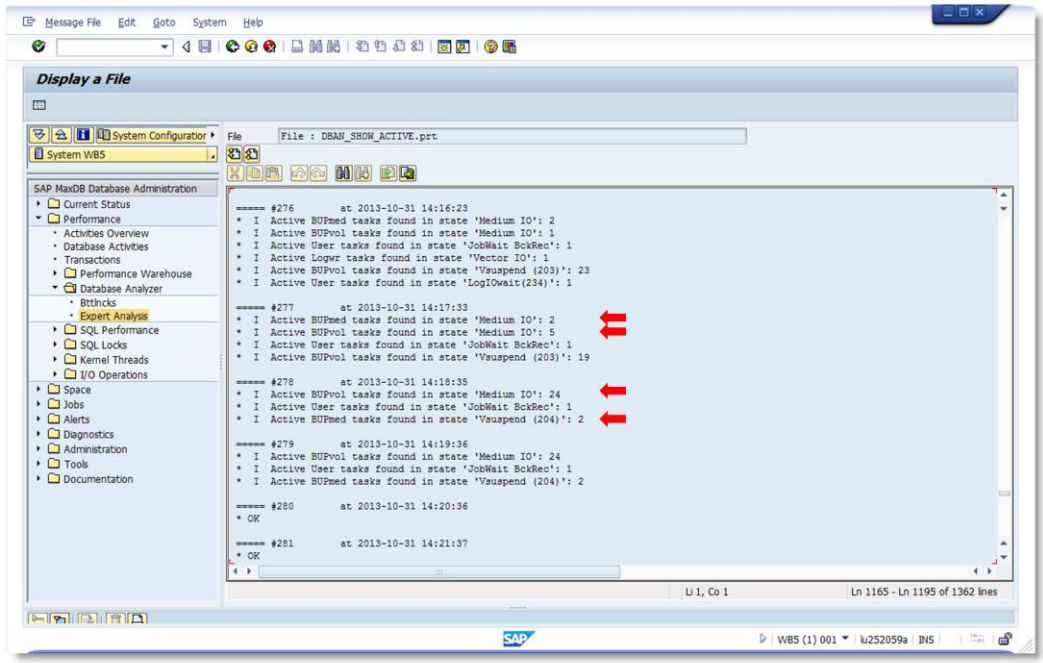
Backup template WB5_backup_COMP_parallel has been created. Backups using this template are compressed as well and write to 2 files in parallel. (Device 1 and Device 2)

5.1. Optimize Runtime of Data Backup: Backup (2)



Lets check now the runtime of the backup with parallel medium executed at 31.10.2013 from 2:05 pm to 2:20 pm (runtime is now only 15 minutes instead of 27 minutes)

5.1. Optimize Runtime of Data Backup: DBAN_SHOW_ACTIVE.prt



Let's have a look at snapshot 14:17:33:

Now 2 tasks in parallel are busy with a write I/O to the backup medium (*BUPmed*)

5 server tasks are busy with reading from the data volumes (*BUPvol 'Medium IO'*)

Snapshot 14:18:35:

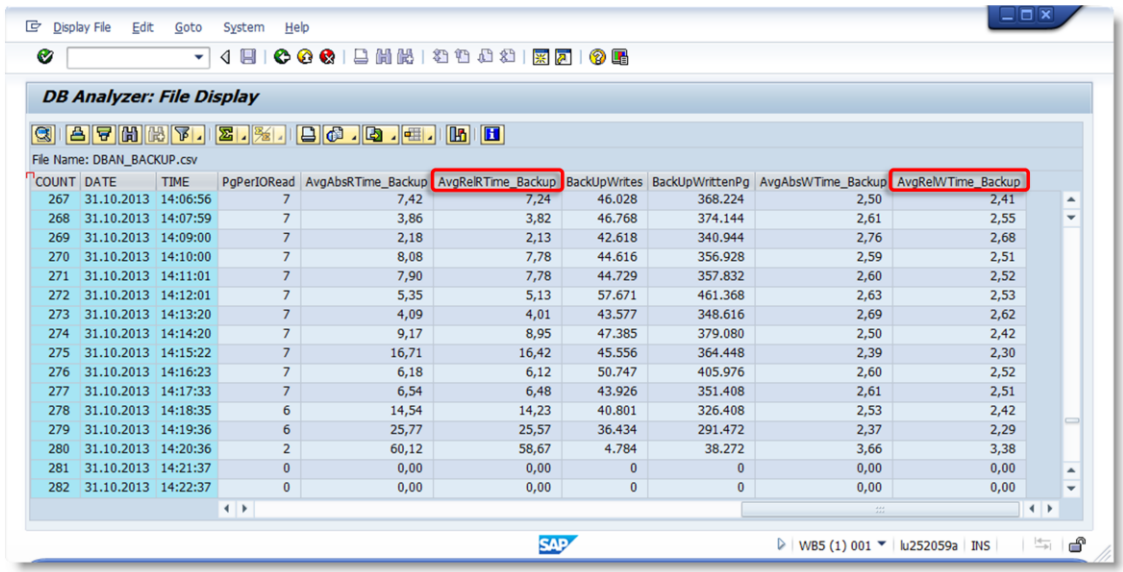
All 24 server tasks are busy with reading from the data volumes (*BUPvol 'Medium IO'*)

Both server tasks that have the write request are waiting (*BUPmed Vsuspend*).

The 2 servertasks which write the data to the external media are faster than only one task. So in this example the ring buffer was never
filled up completely. The server tasks which read the data from the volumes to the ring buffer have no wait (vsuspend) situations.
But we can see now wait situations on the servertasks which have to write to the external media.

Now the bottleneck is on server tasks side that have the read request.
Let's check now the throughput of the I/O in DBAN_backup.csv.

5.1. Optimize Runtime of Data Backup: DBAN_BACKUP.csv



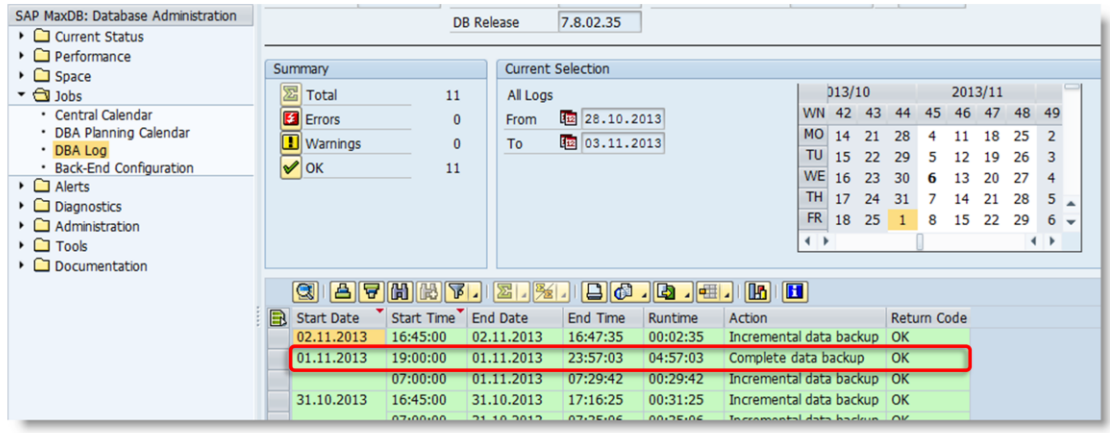
COUNT	DATE	TIME	PgPerIORead	AvgAbsRTime_Backup	AvgRelRTime_Backup	BackUpWrites	BackUpWrittenPg	AvgAbsWTime_Backup	AvgRelWTime_Backup
267	31.10.2013	14:06:56	7	7,42	7,24	46.028	368.224	2,50	2,41
268	31.10.2013	14:07:59	7	3,86	3,82	46.768	374.144	2,61	2,55
269	31.10.2013	14:09:00	7	2,18	2,13	42.618	340.944	2,76	2,68
270	31.10.2013	14:10:00	7	8,08	7,78	44.616	356.928	2,59	2,51
271	31.10.2013	14:11:01	7	7,90	7,78	44.729	357.832	2,60	2,52
272	31.10.2013	14:12:01	7	5,35	5,13	57.671	461.368	2,63	2,53
273	31.10.2013	14:13:20	7	4,09	4,01	43.577	348.616	2,69	2,62
274	31.10.2013	14:14:20	7	9,17	8,95	47.385	379.080	2,50	2,42
275	31.10.2013	14:15:22	7	16,71	16,42	45.556	364.448	2,39	2,30
276	31.10.2013	14:16:23	7	6,18	6,12	50.747	405.976	2,60	2,52
277	31.10.2013	14:17:33	7	6,54	6,48	43.926	351.408	2,61	2,51
278	31.10.2013	14:18:35	6	14,54	14,23	40.801	326.408	2,53	2,42
279	31.10.2013	14:19:36	6	25,77	25,57	36.434	291.472	2,37	2,29
280	31.10.2013	14:20:36	2	60,12	58,67	4.784	38.272	3,66	3,38
281	31.10.2013	14:21:37	0	0,00	0,00	0	0	0,00	0,00
282	31.10.2013	14:22:37	0	0,00	0,00	0	0	0,00	0,00

The runtime of the backup was less than before, but the I/O times measured for reading from volumes (*AvgRelRTime_Backup*) is increased, sometimes over 10 msec. Why?

In the configuration of this database all 24 Data Volumes are located on the same disk. This is not the configuration SAP is recommending.
Because more server tasks get read requests more often and in parallel the bottleneck now is the Read IO.

To solve this bottleneck the database configuration has to be changed -> distribute the data volumes on different disks.

5.1. Optimize Runtime of Data Backup: Backup (3)

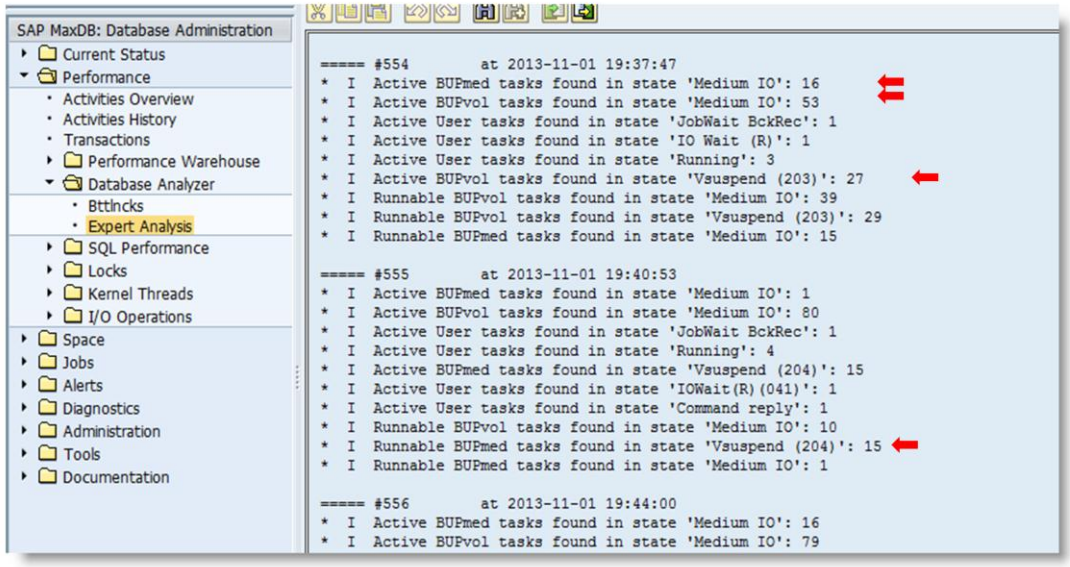


Now we have a backup analysis of a 4 Tbyte customer system.
This customer configuration has 80 Data volumes distributed on several disks.

Let's have a closer look to this backup performance to see if we can optimize here as well.

A complete data backup was executed at 1st of November between 7:00 pm and 11:57 pm – around 5 hours runtime of backup.

5.1. Optimize Runtime of Data Backup: DBAN_SHOW_ACTIVE.prt



Let's have a look at snapshot 07:37:47 pm:

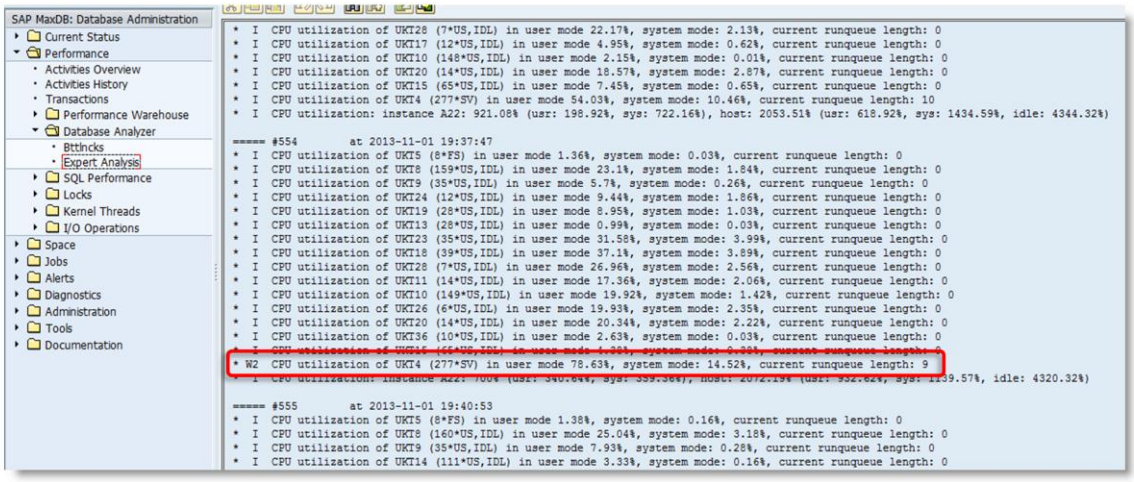
Interesting are the BUPmed and BUPvol tasks.

There are 16 tasks in parallel busy with write I/O to the backup medium (*BUPmed*) – So now we know this is a parallel backup template with 16 parallel devices.

53 server tasks are busy with reading from the data volumes (*BUPvol 'Medium IO'*)

27 server tasks that have the read request are waiting *(BUPmed 'Vsuspend')*

5.1. Optimize Runtime of Data Backup: DBAN_UKT_CPU_UTILIZATION.prt



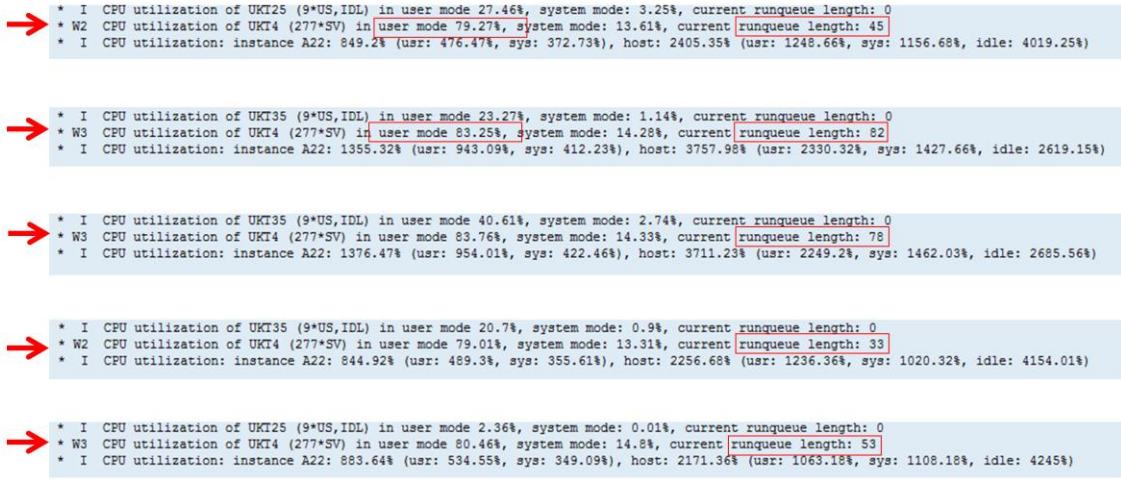
File DBAN_UKT_CPU_UTILIZATION.prt contains the detailed information which UKT (user-kernel thread) is responsible for which CPU consumption.

The only thread that is of interest for the runtime analysis of a data backup is the thread in which the server tasks are configured.
-> Thread 4 – during this backup the most CPU load was in thread no 4 (~74% in User and 14,5% in System)

Note as well that the runqueue has 9 elements which are ready to use CPU but cannot get them because the CPU is used by another task in the same thread.

In this case all server tasks (read and write request) are in UKT 4. The CPU consumption of the UKT4 is high (78%) and the current runqueue length (9) points to a CPU bottleneck during the backup.

5.1. Optimize Runtime of Data Backup: DBAN_UKT_CPU_UTILIZATION.prt



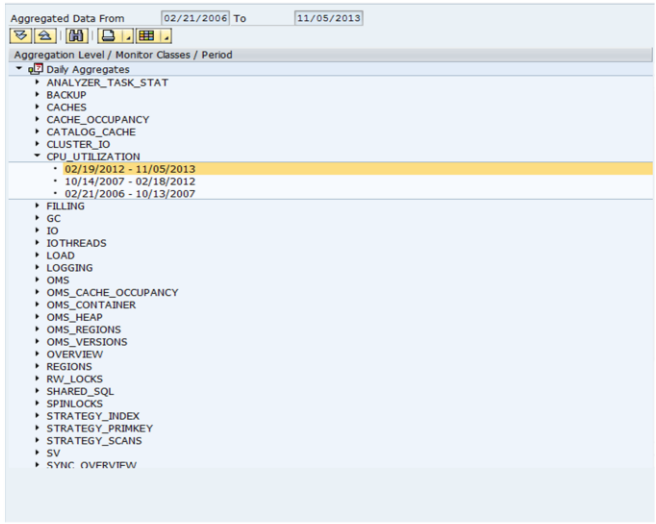
We focus on the CPU Utilization of the UKT no 4. UKT no 4 includes all server tasks.
In the displayed snapshots the CPU consumption of the UKT4 is between 79% and 83% and the runqueue length increase up to 82 entries!

This pints to a CPU bottleneck which appears during the total backup time.
Server task distribution to several UKTs could be the solution for such a CPU bottleneck.
With Kernel parameter configuration the server tasks can be distributed on different Threads. You can decide if the ServerTasks are in a user seperated Thread or together with the Users Tasks in the Threads. For more information, see Note 1672994.

Be careful – this configuration change should never be done directly on the productive system.
Notice that server tasks are used for backup, Check Data, create index, read ahead and during savepoints.

5.2. Aggregation Analysis (1)

The Database Analyzer aggregated values are used to compare statistic values



e.g.

- before and after an MaxDB software upgrade
- performance problems since a known date
- hardware configuration changes

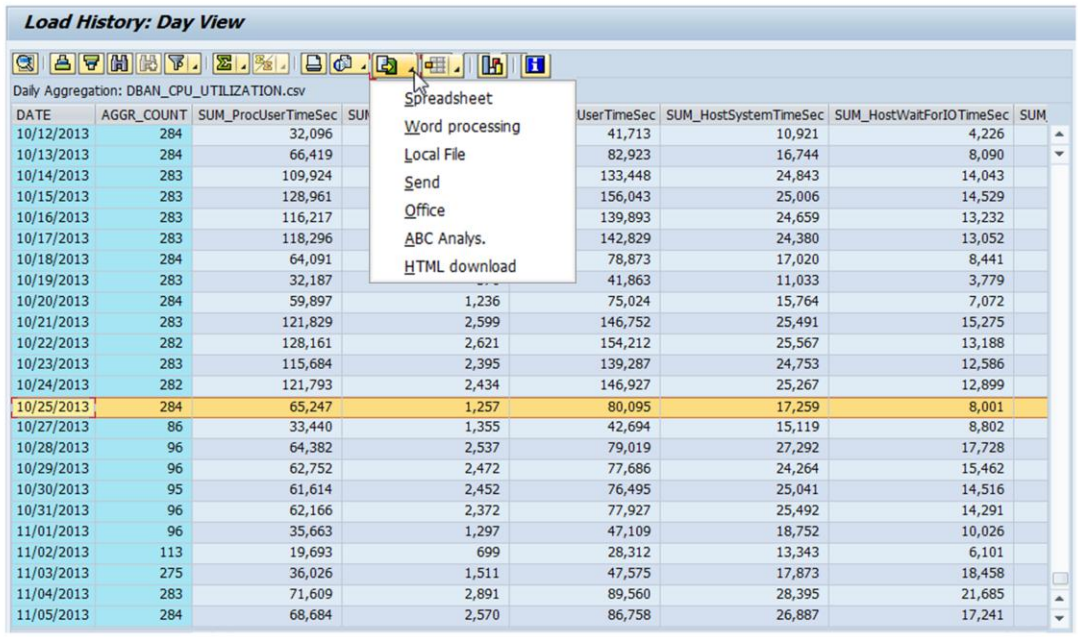
The aggregated Database Analyzer statistic values are used to compare a system e.g. before and after a software upgrade. If the customer detecs performance problems since a special date or hardware components on the database server have been changed.

The first example is based on a liveCache customer system which was upgraded from version 7.7.07.37 to 7.7.07.45 at 26th of October 2013.

We are using the daily aggregates to compare the system before and after the upgrade.

There are several time frames listed in the aggregation view if the structure of the file has changed – like in CPU_UTILIZATION.

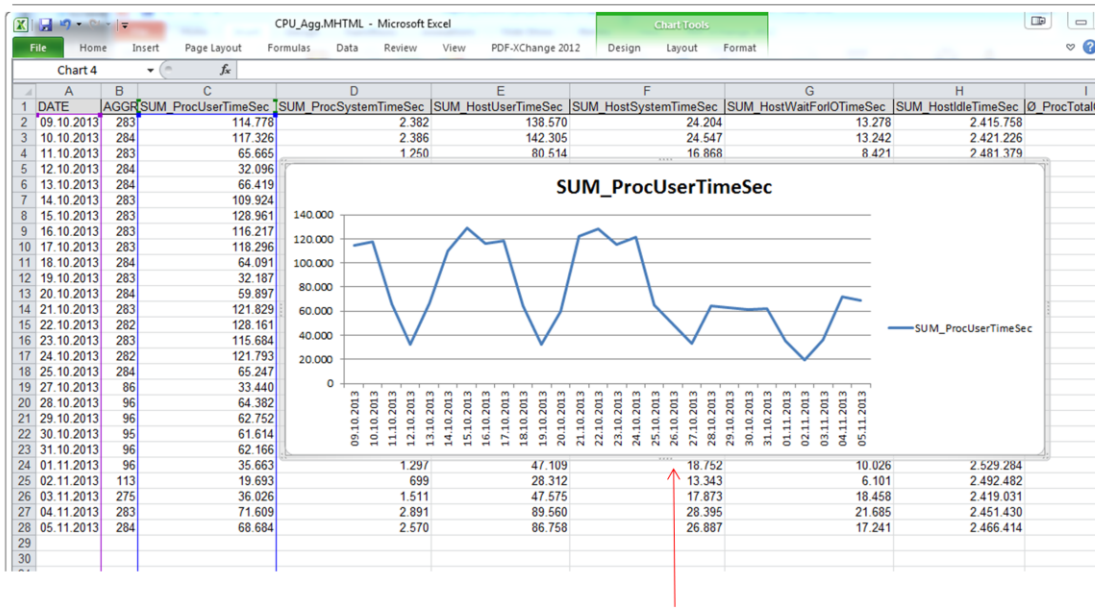
5.2. Aggregation Analysis – CPU_UTILIZATION (1)



DATE	AGGR_COUNT	SUM_ProcUserTimeSec	SUM	UserTimeSec	SUM_HostSystemTimeSec	SUM_HostWaitForIOTimeSec	SUM
10/12/2013	284	32,096		41,713	10,921	4,226	
10/13/2013	284	66,419		82,923	16,744	8,090	
10/14/2013	283	109,924		133,448	24,843	14,043	
10/15/2013	283	128,961		156,043	25,006	14,529	
10/16/2013	283	116,217		139,893	24,659	13,232	
10/17/2013	283	118,296		142,829	24,380	13,052	
10/18/2013	284	64,091		78,873	17,020	8,441	
10/19/2013	283	32,187		41,863	11,033	3,779	
10/20/2013	284	59,897	1,236	75,024	15,764	7,072	
10/21/2013	283	121,829	2,599	146,752	25,491	15,275	
10/22/2013	282	128,161	2,621	154,212	25,567	13,188	
10/23/2013	283	115,684	2,395	139,287	24,753	12,586	
10/24/2013	282	121,793	2,434	146,927	25,267	12,899	
10/25/2013	284	65,247	1,257	80,095	17,259	8,001	
10/27/2013	86	33,440	1,355	42,694	15,119	8,802	
10/28/2013	96	64,382	2,537	79,019	27,292	17,728	
10/29/2013	96	62,752	2,472	77,686	24,264	15,462	
10/30/2013	95	61,614	2,452	76,495	25,041	14,516	
10/31/2013	96	62,166	2,372	77,927	25,492	14,291	
11/01/2013	96	35,663	1,297	47,109	18,752	10,026	
11/02/2013	113	19,693	699	28,312	13,343	6,101	
11/03/2013	275	36,026	1,511	47,575	17,873	18,458	
11/04/2013	283	71,609	2,891	89,560	28,395	21,685	
11/05/2013	284	68,684	2,570	86,758	26,887	17,241	

In the CPU-Utilization we can see that the sum of CPU time in User Mode (database kernel software) decreased after the upgrade.
The are huge number of values to compare. It is much easier to have a graphical view.
Choose Spreadsheet to download the content into an excel sheet.

5.2. Aggregation Analysis – CPU_UTILIZATION (2)



DATE	AGGR	SUM_ProcUserTimeSec	SUM_ProcSystemTimeSec	SUM_HostUserTimeSec	SUM_HostSystemTimeSec	SUM_HostWaitForIOTimeSec	SUM_HostIdleTimeSec	Ø_ProcTotalC
09.10.2013	283	114.778	2.382	138.570	24.204	13.278	2.415.758	
10.10.2013	284	117.326	2.386	142.305	24.547	13.242	2.421.226	
11.10.2013	283	65.665	1.250	80.514	16.868	8.421	2.481.379	
12.10.2013	284	32.096						
13.10.2013	284	66.419						
14.10.2013	283	109.924						
15.10.2013	283	128.961						
16.10.2013	283	116.217						
17.10.2013	283	118.296						
18.10.2013	284	64.091						
19.10.2013	283	32.187						
20.10.2013	284	59.897						
21.10.2013	283	121.829						
22.10.2013	282	128.161						
23.10.2013	283	115.684						
24.10.2013	282	121.793						
25.10.2013	284	65.247						
27.10.2013	86	33.440						
28.10.2013	96	64.382						
29.10.2013	96	62.752						
30.10.2013	95	61.614						
31.10.2013	96	62.166						
01.11.2013	96	35.663	1.297	47.109	18.752	10.026	2.529.284	
02.11.2013	113	19.693	699	28.312	13.343	6.101	2.492.482	
03.11.2013	275	36.026	1.511	47.575	17.873	18.458	2.419.031	
04.11.2013	283	71.609	2.891	89.560	28.395	21.685	2.451.430	
05.11.2013	284	68.684	2.570	86.758	26.887	17.241	2.466.414	

The aggregated values can only be transferred into graphics with Excel. The Database Studio does not support this feature.

We can easily see that the CPU_USAGE of the database in user Mode decreases since 26.10.2013.
The reason for this could be Code change in liveCache kernel, code change in LCAPPS, or other reasons.

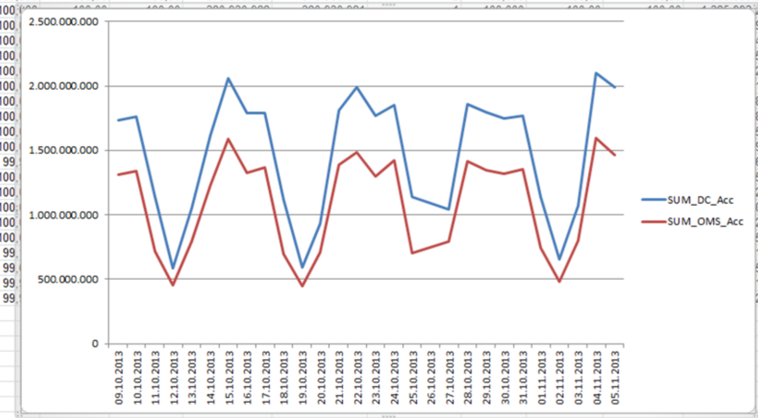
The database Analyzer expert view allows to get more detailed information about the liveCache ressource usage.
Notice that the liveCache kernel and the LCAPPS are running in the same process.
Currently we do not have detailed information about LCAPPS statistics in the database Analyzer expert analysis.

To confirm that the CPU usage has really decreased it is important to check if the application load before and after the upgrade has been the same. Here you have several expert analysis aggregation views.

5.2. Aggregation Analysis – CACHES

Cache_AGG.MHTML - Microsoft Excel

	C	D	E	F	G	H	I	J	K	L	M	N	O
1	SUM_DC_Acc	SUM_DC_Succ	SUM_DC_Fails	Ø_DC_Hit	MIN_DC_Hit	MAX_DC_Hit	SUM_SQL_Acc	SUM_SQL_Succ	SUM_SQL_Fails	Ø_SQL_Hit	MIN_SQL_Hit	MAX_SQL_Hit	SUM_OMS_Acc
2	1.733.142.833	1.733.142.246	587	100,000	100,00	100,00	336.977.049	336.977.042	7	100,000	100,00	100,00	1.310.431.931
3	1.759.843.772	1.759.843.327	445	100,000	100,00	100,00	334.627.326	334.627.323	3	100,000	100,00	100,00	1.338.068.661
4	1.140.721.596	1.140.721.267	329	100,000	100,00	100,00	381.530.973	381.530.973	0	100,000	100,00	100,00	714.794.799
5	587.418.949	587.418.895	54	100,000	100,00	100,00	117.142.514	117.142.514	0	100,000	100,00	100,00	451.524.594
6	1.049.517.672	1.049.517.479	193	100,000	100,00	100,00	204.711.356	204.711.356	0	100,000	100,00	100,00	793.196.006
7	1.618.852.716	1.618.852.372	344	100,000	100,00	100,00	307.814.154	307.814.152	2	100,000	100,00	100,00	1.228.412.179
8	2.056.436.993	2.056.436.640	353	100,000	100,00	100,00	375.566.653	375.566.651	2	100,000	100,00	100,00	1.586.089.238
9	1.792.565.334	1.792.565.037	297	100,									208
10	1.786.997.145	1.786.996.874	271	100,									509
11	1.116.162.493	1.116.162.372	121	100,									426
12	594.462.502	594.462.462	40	100,									527
13	928.551.157	928.551.067	90	100,									217
14	1.812.665.891	1.812.665.703	188	100,									120
15	1.991.651.996	1.991.651.815	181	100,									843
16	1.769.561.018	1.769.560.853	165	100,									874
17	1.854.956.626	1.854.956.420	206	100,									570
18	1.139.156.305	1.139.156.123	182	100,									598
19	1.040.835.390	1.040.785.791	49.599	99,									661
20	1.857.641.550	1.857.636.723	4.827	100,									506
21	1.794.958.962	1.794.958.334	628	100,									278
22	1.745.996.697	1.745.996.523	174	100,									001
23	1.771.562.662	1.771.562.580	82	100,									239
24	1.141.574.338	1.141.574.271	67	100,									522
25	653.547.656	652.492.246	1.055.410	99,									119
26	1.066.930.977	1.065.793.066	1.137.911	99,									548
27	2.099.645.723	2.099.540.340	105.383	99,									198
28	1.989.923.198	1.989.893.472	29.726	99,									210

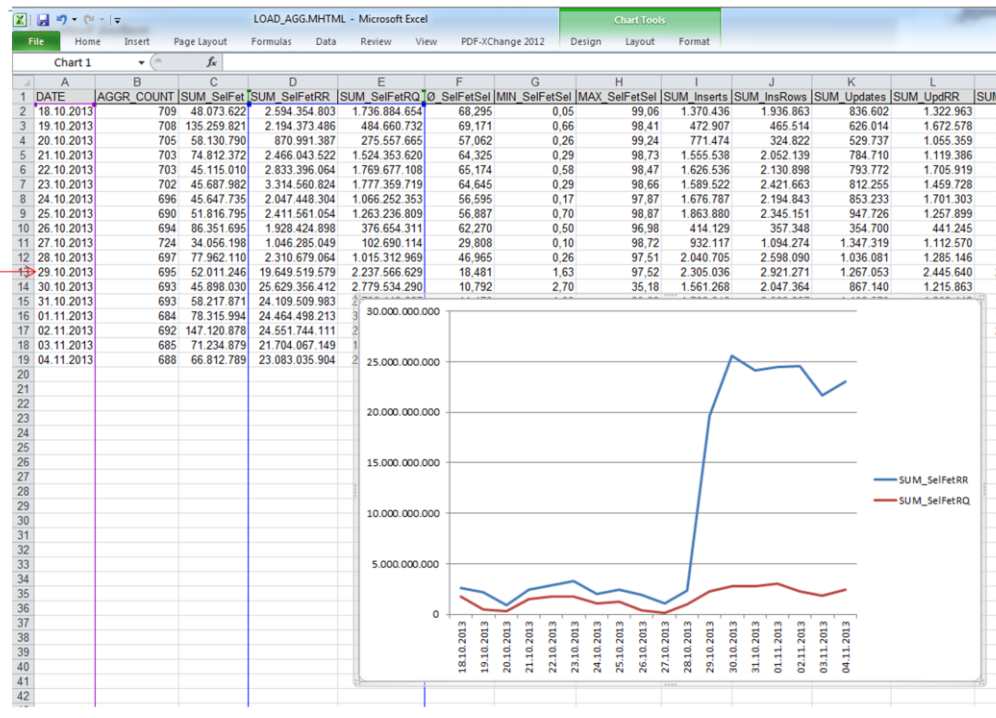


 Public 53

We can check the aggregated statistic values of Cache Accesses to check if the system load related to Data Cache Accesses (blue) have been the same than before the upgrade. Because this is a liveCache application the Sum of OMS Accesses will be interesting as well.

To get closer to the reason for the lower CPU usage a detailed analysis must follow and detailed expert knowledge about liveCache application and liveCache architecture is necessary. This analysis is done at SAP with the colleagues of active global support for Max Attention customers and in team work with experts of liveCache application and liveCache kernel.

5.2. Aggregation Analysis – SQL Performance (1)

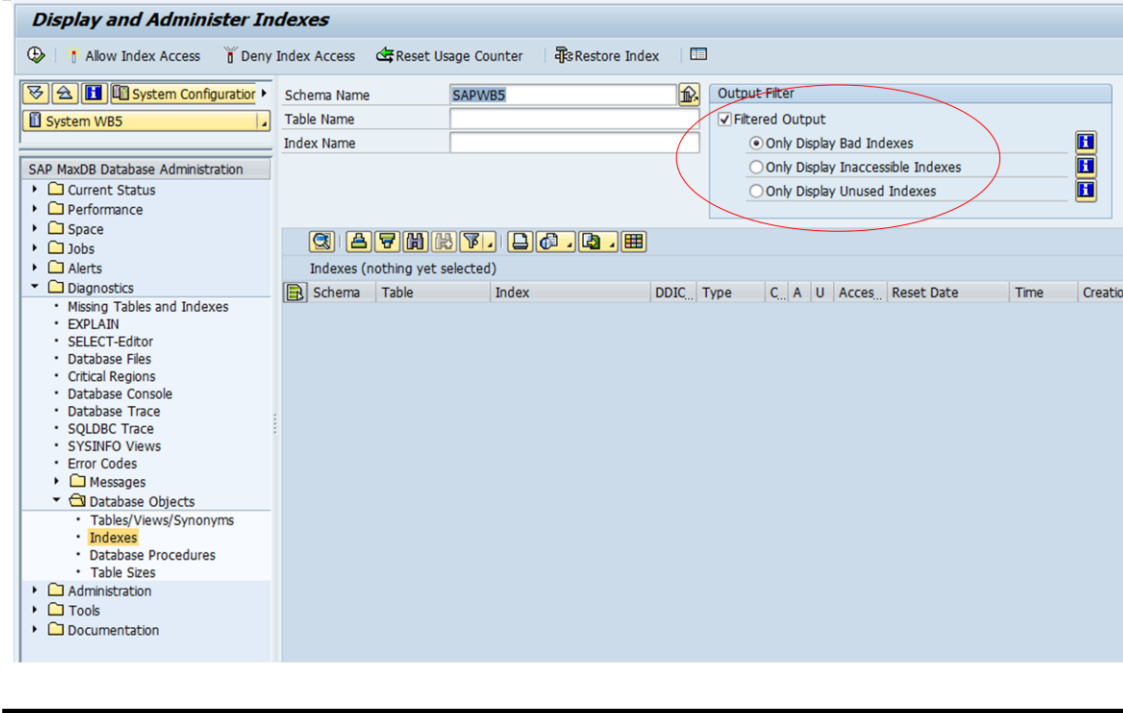


	A	B	C	D	E	F	G	H	I	J	K	L
1	DATE	AGGR_COUNT	SUM_SelFet	SUM_SelFetRR	SUM_SelFetRQ	Ø_SelFetSel	MIN_SelFetSel	MAX_SelFetSel	SUM_Inserts	SUM_InsRows	SUM_Updates	SUM_UpdRR
2	18.10.2013	709	48.073.622	2.594.354.803	1.736.884.654	68,295	0,05	99,06	1.370.436	1.936.863	836.602	1.322.963
3	19.10.2013	708	135.259.821	2.194.373.486	484.660.732	69,171	0,66	98,41	472.907	465.514	626.014	1.672.578
4	20.10.2013	705	58.130.790	870.991.387	275.557.665	57,062	0,26	99,24	771.474	324.822	529.737	1.055.359
5	21.10.2013	703	74.812.372	2.466.043.522	1.524.353.620	64,325	0,29	98,73	1.555.538	2.052.139	784.710	1.119.386
6	22.10.2013	703	45.115.010	2.833.396.064	1.769.677.108	65,174	0,58	98,47	1.626.536	2.130.898	793.772	1.705.919
7	23.10.2013	702	45.687.982	3.314.560.824	1.777.359.719	64,645	0,29	98,66	1.589.522	2.421.663	812.255	1.459.728
8	24.10.2013	696	45.647.735	2.047.448.304	1.066.252.353	56,595	0,17	97,87	1.676.787	2.194.843	853.233	1.701.303
9	25.10.2013	690	51.816.795	2.411.561.054	1.263.236.809	56,887	0,70	98,87	1.863.880	2.345.151	947.726	1.257.899
10	26.10.2013	694	86.351.695	1.928.424.898	376.654.311	62,270	0,50	96,98	414.129	357.348	354.700	441.245
11	27.10.2013	724	34.056.198	1.046.285.049	102.690.114	29,808	0,10	98,72	932.117	1.094.274	1.347.319	1.112.570
12	28.10.2013	697	77.962.110	2.310.679.064	1.015.312.969	46,965	0,26	97,51	2.040.705	2.598.090	1.036.081	1.285.146
13	29.10.2013	695	52.011.246	19.649.519.579	2.237.566.629	18,481	1,63	97,52	2.305.036	2.921.271	1.267.053	2.445.640
14	30.10.2013	693	45.898.030	25.629.356.412	2.779.534.290	10,792	2,70	35,18	1.561.268	2.047.364	867.140	1.215.863
15	31.10.2013	693	58.217.871	24.109.509.983								
16	01.11.2013	684	78.315.994	24.464.498.213								
17	02.11.2013	692	147.120.878	24.551.744.111								
18	03.11.2013	685	71.234.879	21.704.067.149								
19	04.11.2013	688	66.812.789	23.083.035.904								

The next example was a customer system which had performance problems since 29th of October. Again we are using the expert analysis aggregation view to compare the statistics data before and after the 29th of October.
The customer could not tell us the transaction or the report which is slow but the information that the batch process during the night is very very slow suddenly. The customer told us that nothing has been changed from application side.
In such cases we won't use the command monitor to catch the commands in a first step.
SAP first checks with the aggregated values.
In this example we check the system load – have it really changed.
You could start with DBAN_transactions to check if the sum of SQL commands has changed.
Here we check the daily aggregated values of DBAN_LOAD.csv
In DBAN_LOAD we get information about the number of Select and Fetches, the number of Insert, Updates and Deletes.
We get as well information about the Selectivity, which means how many rows have to be read and how many of these read rows have been qualified.
The selectivity gives a hint if there are missing or bad indexes and therefore a high system load.
We can see that the number of Select and fetches are the same for all days.
We can also see that the number of rows read (column D) to the number of rows qualified (Column E) was getting worse at 29th of October.
And this did not change until 4.11.2013
The Bad selectivity points to missing or BAD indexes.

So first we check if the are BAD indexes in the system -> DBACockpit -> Diagnostics -> database Objects -> Indexes

5.2. Aggregation Analysis – SQL Performance (2)



If there are bad indexes shown – those indexes cannot be used of the optimizer anymore and this could be the reason for the worse Selectivity.

Use ‘Restore Index’ to create those indexes again (Recreate Index) –but be careful only one create index should be active in the system. And indexes should be created when there is low system load.

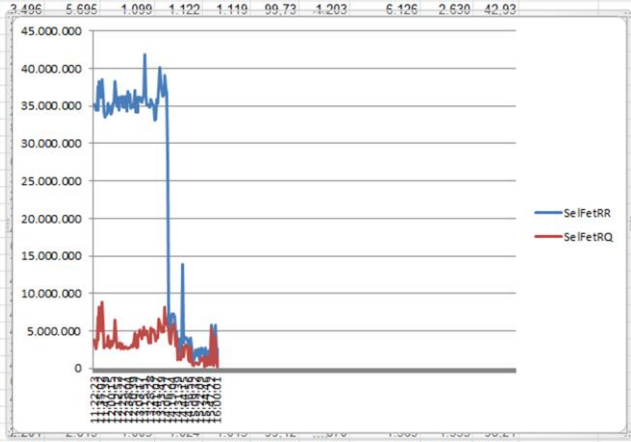
Another issue could be that really an index is missing to find the best strategy. But this index was missing before the 29th of october as well. May be the customer has changed the application coding? In these cases you use Command monitor to catch the SQL statement with the worse selectivity and create a new index.

If you detect such issues and you cannot solve this issue by your own please open a CSS ticket on BC-Db-SDB.
We help to find the root cause.

After the problem has been solved you can use the Database Analyzer expert analysis LOAD.csv as well to check the result.

5.2. Aggregation Analysis – SQL Performance (3)

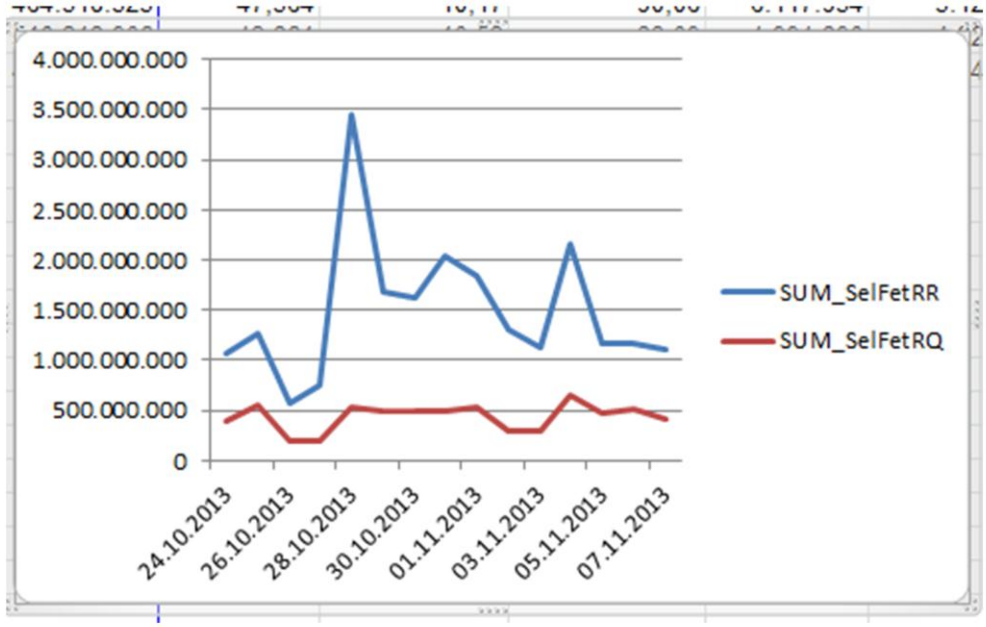
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
07.11.2013	13:20:05	6	126	70.653	36.360.217	4.987.883	13,72	4.748	5.408	1.359	1.247	1.247	100,00	2.476	3.059	3.057	99,93		
07.11.2013	13:22:11	7	126	119.015	35.102.457	4.969.914	14,16	5.142	6.381	1.815	1.683	1.677	99,64	1.952	2.857	2.856	99,96		
07.11.2013	13:24:18	4	127	93.997	35.194.705	4.344.899	12,35	6.052	8.895	1.275	1.303	1.299	99,69	2.634	4.384	4.350	99,22		
07.11.2013	13:26:22	6	124	64.879	34.904.010	3.426.268	9,82	2.648	3.291	816	917	915	99,78	1.470	2.362	2.361	99,96		
07.11.2013	13:28:28	6	126	76.003	34.852.947	3.796.307	10,89	2.274	3.155	915	913	902	98,80	588	1.489	1.396	93,75		
07.11.2013	13:30:34	6	126	63.952	35.272.546	3.379.869	9,58	1.901	2.542	820	987	971	98,38	547	1.052	1.051	99,90		
07.11.2013	13:32:40	6	126	109.044	35.886.777	5.407.364	15,07	4.099	5.119	1.209	1.446	1.331	92,05	1.120	2.092	2.076	99,24		
07.11.2013	13:34:46	7	126	79.642	35.557.851	5.254.641	14,78	3.496	5.695	1.099	1.122	1.119	99,73	1.203	6.126	2.630	42,93		
07.11.2013	13:36:53	6	127	88.487	35.182.680	4.855.837	13,80												
07.11.2013	13:38:59	8	126	99.216	35.017.256	5.115.666	14,61												
07.11.2013	13:41:07	8	128	123.188	33.111.462	4.492.506	13,57												
07.11.2013	13:43:15	6	128	121.661	33.295.300	3.702.932	11,12												
07.11.2013	13:45:21	7	126	72.198	35.859.160	4.341.491	12,11												
07.11.2013	13:47:28	7	127	101.374	35.320.569	4.070.838	11,53												
07.11.2013	13:49:35	7	127	76.504	37.655.981	6.576.353	17,46												
07.11.2013	13:51:42	7	127	95.479	40.114.565	6.004.160	14,97												
07.11.2013	13:53:49	6	127	81.686	38.518.547	5.722.047	14,86												
07.11.2013	13:55:55	6	126	56.381	37.255.956	4.844.692	13,00												
07.11.2013	13:58:01	6	126	60.360	36.257.009	4.835.627	13,34												
07.11.2013	14:00:07	6	126	99.145	36.442.873	4.911.548	13,48												
07.11.2013	14:02:14	6	127	138.415	38.191.813	8.161.087	21,37												
07.11.2013	14:04:20	7	126	120.300	39.032.814	7.476.790	19,16												
07.11.2013	14:06:27	6	127	116.594	37.416.400	6.580.626	17,59												
07.11.2013	14:08:33	5	126	96.723	36.638.530	5.965.741	16,28												
07.11.2013	14:10:38	6	125	58.622	27.861.894	5.625.905	20,19												
07.11.2013	14:12:44	6	126	78.384	5.911.031	5.087.547	86,07												
07.11.2013	14:14:50	7	126	90.908	4.600.521	3.510.485	76,31												
07.11.2013	14:16:57	7	127	81.636	4.986.470	3.228.012	64,74												
07.11.2013	14:19:04	5	127	72.029	7.186.305	4.997.416	69,54												
07.11.2013	14:21:09	7	125	112.290	7.221.989	5.511.907	76,32												
07.11.2013	14:23:16	6	127	91.808	7.274.566	5.904.143	81,16												
07.11.2013	14:25:22	6	126	98.121	6.793.100	5.327.285	78,42												
07.11.2013	14:27:28	5	126	85.097	4.687.623	3.001.867	64,04												
07.11.2013	14:29:33	6	125	66.290	4.802.710	4.219.011	87,85	2.968	3.271	873	865	865	100,00	1.183	1.519	1.519	100,00		
07.11.2013	14:31:39	6	126	71.044	2.582.112	1.863.283	72,16	10.746	11.035	812	846	842	99,53	11.504	9.415	9.414	99,99		
07.11.2013	14:33:45	6	126	100.961	2.209.391	1.153.249	52,20	3.139	3.659	1.191	1.292	1.287	99,61	1.137	1.840	1.826	99,24		
07.11.2013	14:35:51	6	126	78.483	2.157.130	1.236.350	57,31	2.631	3.551	1.363	1.411	1.411	100,00	1.156	20.338	1.951	9,59		
07.11.2013	14:37:57	5	126	59.875	3.957.812	1.516.982	38,33	2.530	3.298	945	935	929	99,36	1.500	48.625	2.432	5,00		
07.11.2013	14:40:02	7	125	53.834	1.766.082	1.140.866	64,60	2.345	2.948	779	853	852	99,88	383	45.961	1.165	2,53		
07.11.2013	14:42:09	6	127	109.352	2.516.107	1.701.086	67,61	2.647	3.163	1.143	1.160	1.155	99,57	858	11.962	1.448	12,10		
07.11.2013	14:44:15	7	126	75.862	13.816.384	2.844.597	20,59	3.743	4.666	1.081	1.082	1.064	98,34	941	1.821	1.742	95,66		



After the problem has been allocated and solved you can immediately easily check the result with the database Analyzer expert view.

For good system performance the rows read (blue) line and the rows qual (red line) should be as closely together as possible.

5.2. Aggregation Analysis – SQL Performance (4)



Next exxample is from a custoemr who told me that after the upgrade from 7.8 to 7.9 he has performance problems.

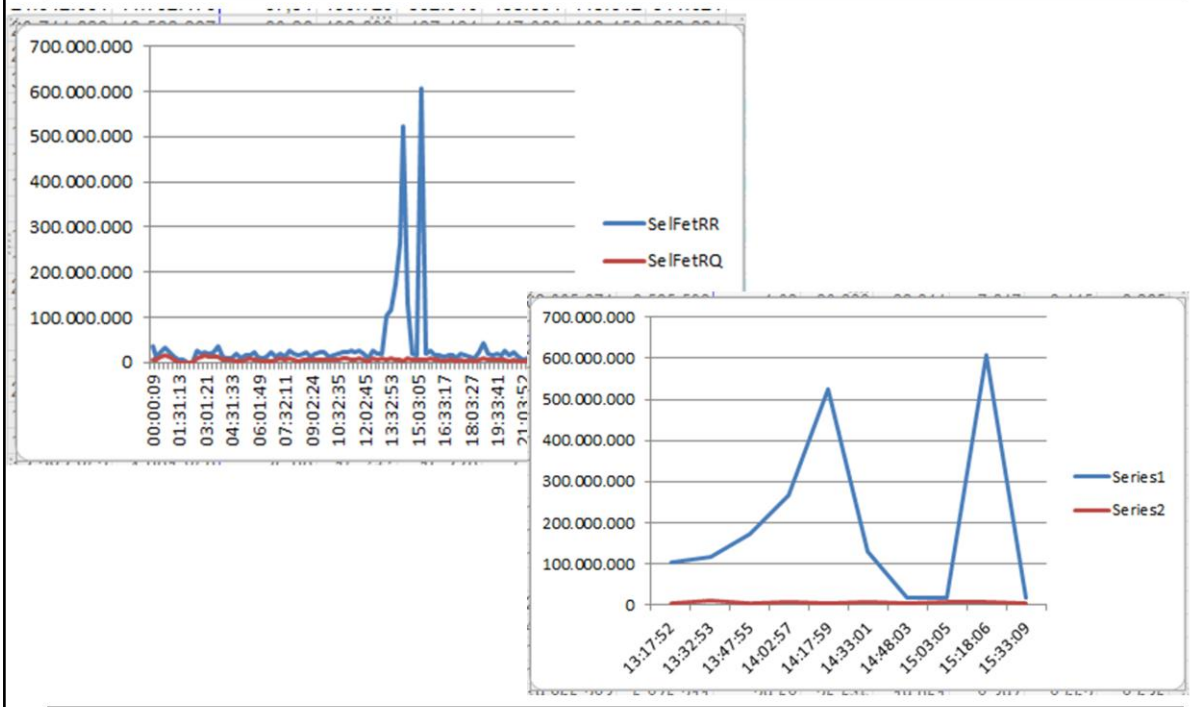
The upgrade has been done on Sunday 27.10.2013.

We use again the Database Analyzer aggregated statistics (daily) with DBAN_LOAD.csv and create a graphic view.
Blue line are the rows that were read and the red line shows the rows qualified.

With the graphical view we easily can see that there are several peaks between 28.10.2013 and 6.11.2013.
Much more data is read than qualified.

The next step is now zoom more detailed to the peaks. We use here the daily statistics values.

5.2. Aggregation Analysis – SQL Performance (5)



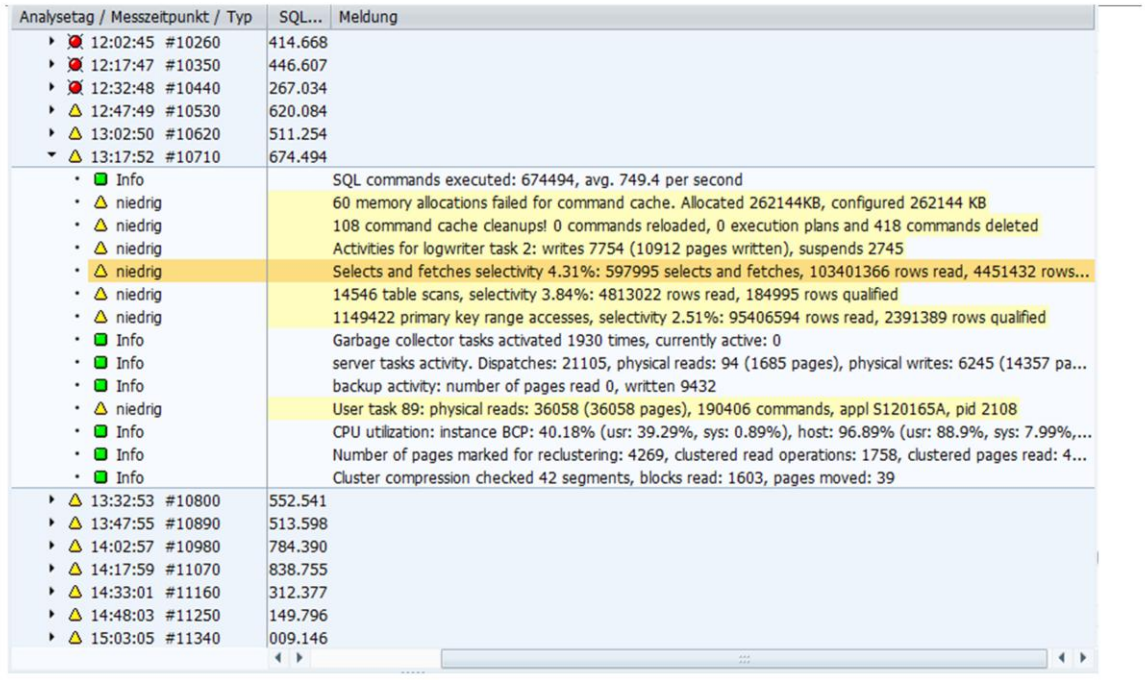
We start with the first peak of 28th of October.

We see now that the peak is between 12:02 and 4:00 pm. With the same created excel we can zoom further to the time frame.

We see that there must be an application active between 1:30 and 2:48, which has to be checked in more detail.
We have to find out which SQL statement was active during this time.

We use again the Database Analyzer Bottleneck analysis to get more details.

5.2. Aggregation Analysis – SQL Performance (6)

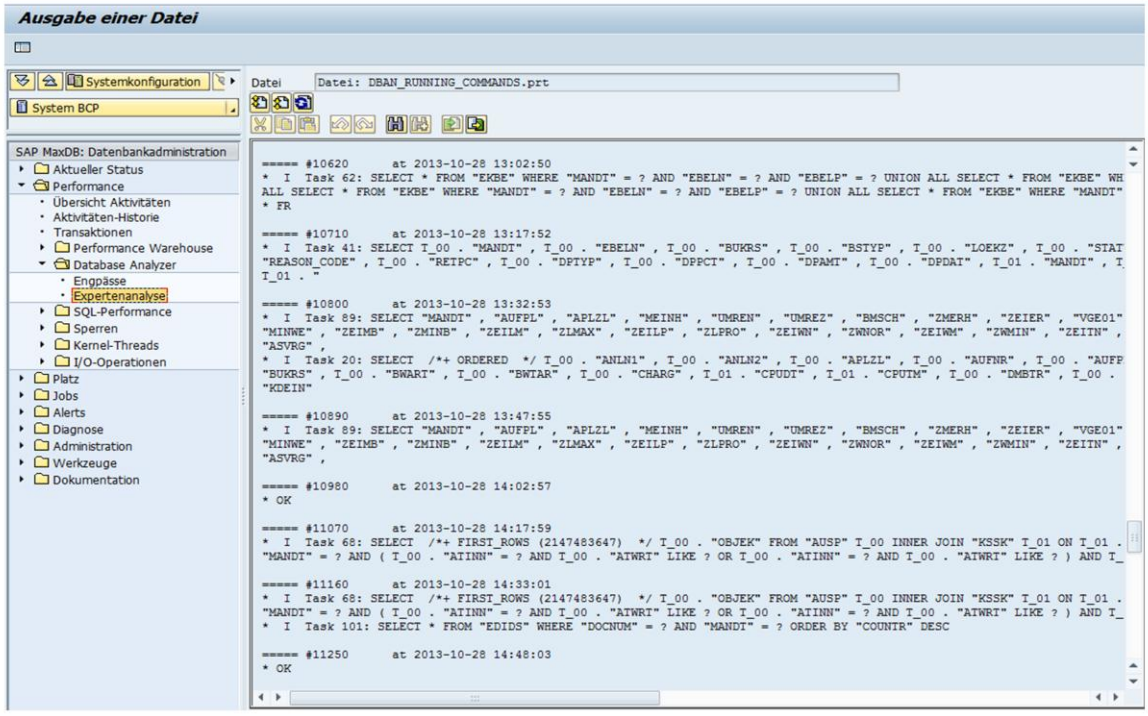


Analysetag / Messzeitpunkt / Typ	SQL...	Meldung
12:02:45 #10260	414.668	
12:17:47 #10350	446.607	
12:32:48 #10440	267.034	
12:47:49 #10530	620.084	
13:02:50 #10620	511.254	
13:17:52 #10710	674.494	
Info		SQL commands executed: 674494, avg. 749.4 per second
niedrig		60 memory allocations failed for command cache. Allocated 262144KB, configured 262144 KB
niedrig		108 command cache cleanups! 0 commands reloaded, 0 execution plans and 418 commands deleted
niedrig		Activities for logwriter task 2: writes 7754 (10912 pages written), suspends 2745
niedrig		Selects and fetches selectivity 4.31%: 597995 selects and fetches, 103401366 rows read, 4451432 rows...
niedrig		14546 table scans, selectivity 3.84%: 4813022 rows read, 184995 rows qualified
niedrig		1149422 primary key range accesses, selectivity 2.51%: 95406594 rows read, 2391389 rows qualified
Info		Garbage collector tasks activated 1930 times, currently active: 0
Info		server tasks activity. Dispatches: 21105, physical reads: 94 (1685 pages), physical writes: 6245 (14357 pa...
Info		backup activity: number of pages read 0, written 9432
niedrig		User task 89: physical reads: 36058 (36058 pages), 190406 commands, appl S120165A, pid 2108
Info		CPU utilization: instance BCP: 40.18% (usr: 39.29%, sys: 0.89%), host: 96.89% (usr: 88.9%, sys: 7.99%,...
Info		Number of pages marked for reclustering: 4269, clustered read operations: 1758, clustered pages read: 4...
Info		Cluster compression checked 42 segments, blocks read: 1603, pages moved: 39
13:32:53 #10800	552.541	
13:47:55 #10890	513.598	
14:02:57 #10980	784.390	
14:17:59 #11070	838.755	
14:33:01 #11160	312.377	
14:48:03 #11250	149.796	
15:03:05 #11340	009.146	

We get the information that User Task 89 did a lot of physical reads. User Task T89 was connected with application process id 2108 (Workprocess) of application server S1200165A.

To get more information which kind of SQL Statements have been executed we use again the Expert Analysis -> DBAN_RUNNING_COMMANDS.

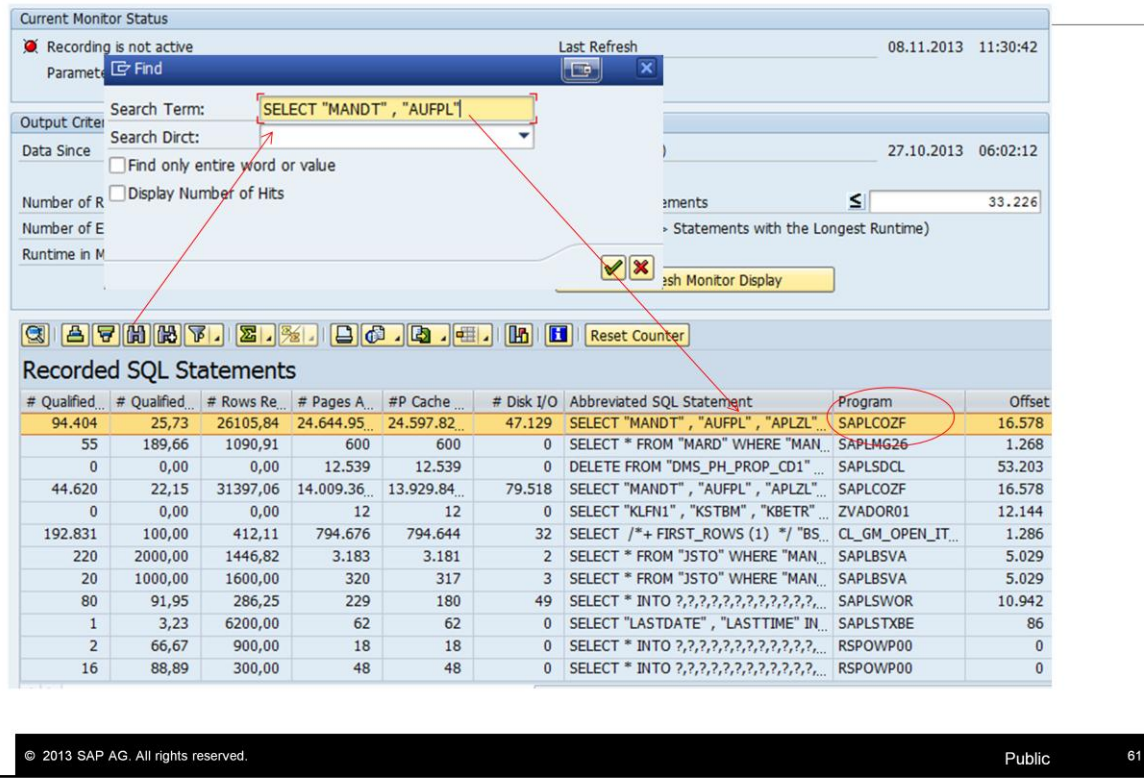
5.2. Aggregation Analysis – SQL Performance (7)



With the expert View of DBAN_running_commands we find out which SQL statements were active in the timeframe 1:02 pm to 2:48 pm.
The information of the bottleneck analysis that task T89 was active can be confirmed. T89 executed several SQL comamnds.

To find out which ABAP programs executed those commands you can use as of SAP MaxDB 7.9 the Resource Monitor.

5.2. Aggregation Analysis – SQL Performance (8)



As of SAP MaxDB version 7.9 the ressource monitor is always active.
You can search for the SQL command logged in RUNNING_COMMANDS.
Please notice that the DBACockpit does not display all commands
Of SharedSQL. Please check first the number of commands stored in SharedSQL with
Select count (*) from commandstatistics and insert the result into the resource monitor *Number of Statements* and *Refresh Monitor Display* first before you search for the SQL command string.
When the command string can be found you 'll get the Report name where this command has been executed the first time.

Agenda

1. Introduction
2. Functional chain
3. Ways to manage Database Analyzer
4. Parameter check with Database Analyzer
5. Expert analysis
6. Useful Information Resources

5. Useful Information Resources

SAP MaxDB documentation:

- http://maxdb.sap.com

  following ➡ “Documentation” ➡ “Version 7.8”

  ➡ “Glossary” ➡ “Database Analyzer”

SAP notes:

- 1423935 “FAQ: SAP MaxDB Database Analyzer”
- 1111426 “Parameter check for liveCache/MaxDB instances”
- 1680854 “Database Analyzer: LOGQUE<no.>: collision rate”
- 1676903 “SAP MaxDB: Runtime analysis of data backup”
- 978027 “Overview note of Database Analyzer problems”

SAP Community Network (SCN):

- http://wiki.sdn.sap.com/wiki/display/MaxDB/MaxDB+Database+Analyzer

  (SAP MaxDB Database Analyzer)

Questions

SAP® MaxDB™ Database Analyzer

SAP® MaxDB™ – Expert Sessions Learning Map (1)

SAP® MaxDB ™ Features	SAP® MaxDB ™ Administration	SAP® MaxDB ™ Problem Analysis
Session 1: Low TCO with the SAP MaxDB Database	Session 2: Basic Administration with Database Studio	Session 5: SAP MaxDB Data Integrity
Session 6: New Features in SAP MaxDB Version 7.7	Session 3: CCMS Integration into the SAP System	Session 14: SAP MaxDB Tracing
Session 8: New Features in SAP MaxDB Version 7.8	Session 11: SAP MaxDB Backup and Recovery	Session 12: Analysis of SQL Locking Situations
	Session 13: Third-Party Backup Tools	
	Session 19: SAP MaxDB Kernel Parameter Handling	

SAP® MaxDB ™ Installation/Upgrade
Session 7: SAP MaxDB Software Update Basics

All Expert Sessions (recording and slides) are available for download
http://maxdb.sap.com/training/

SAP® MaxDB™ – Expert Sessions Learning Map (2)

SAP® MaxDB™ Architecture	SAP® MaxDB ™ Performance
Session 18: Introduction MaxDB Database Architecture	Session 4: Performance Optimization with SAP MaxDB
Session 15: SAP MaxDB No-Reorganization Principle	Session 9: SAP MaxDB Optimized for SAP BW
Session 17: SAP MaxDB Shadow Page Algorithm	Session 16: SAP MaxDB SQL Query Optimization (Part 1)
Session 12: Analysis of SQL Locking Situations	Session 16: SAP MaxDB SQL Query Optimization (Part 2)
Session 10: SAP MaxDB Logging	Session 22: SAP MaxDB Database Analyzer
Session 20: SAP MaxDB Remote SQL Server	
Session 21: SAP MaxDB DBM Server	

All Expert Sessions (recording and slides) are available for download
http://maxdb.sap.com/training/

Thank You!
Bye, Bye – And Remember Next Session

Feedback and further information:
http://www.sdn.sap.com/irj/sdn/maxdb

Next Sessions:
follow in 2014

Thank you

Contact information:

Christiane Hienger
Development Expert IMS
Christiane.Hienger@sap.com

Bettina Laidler
Senior Developer IMS
Bettina.Laidler@sap.com