

# Performance-Analyse von Oracle-Datenbanken mit Panorama



# Otto Group Solution Provider (OSP) GmbH



**Gründung:**

März 1991

**Muttergesellschaft:**

OTTO Group

**Standorte:**

Dresden, Hamburg, Burgkunstadt, Madrid, Taipeh, Bangkok

**Mitarbeiterzahl:**

Ca. 300

**Geschäftsführer:**

Dr. Stefan Borsutzky, Norbert Gödicke, Jens Gruhl

**Website:**

<https://www.osp.de>

# **zur Person**



**Peter Ramm**

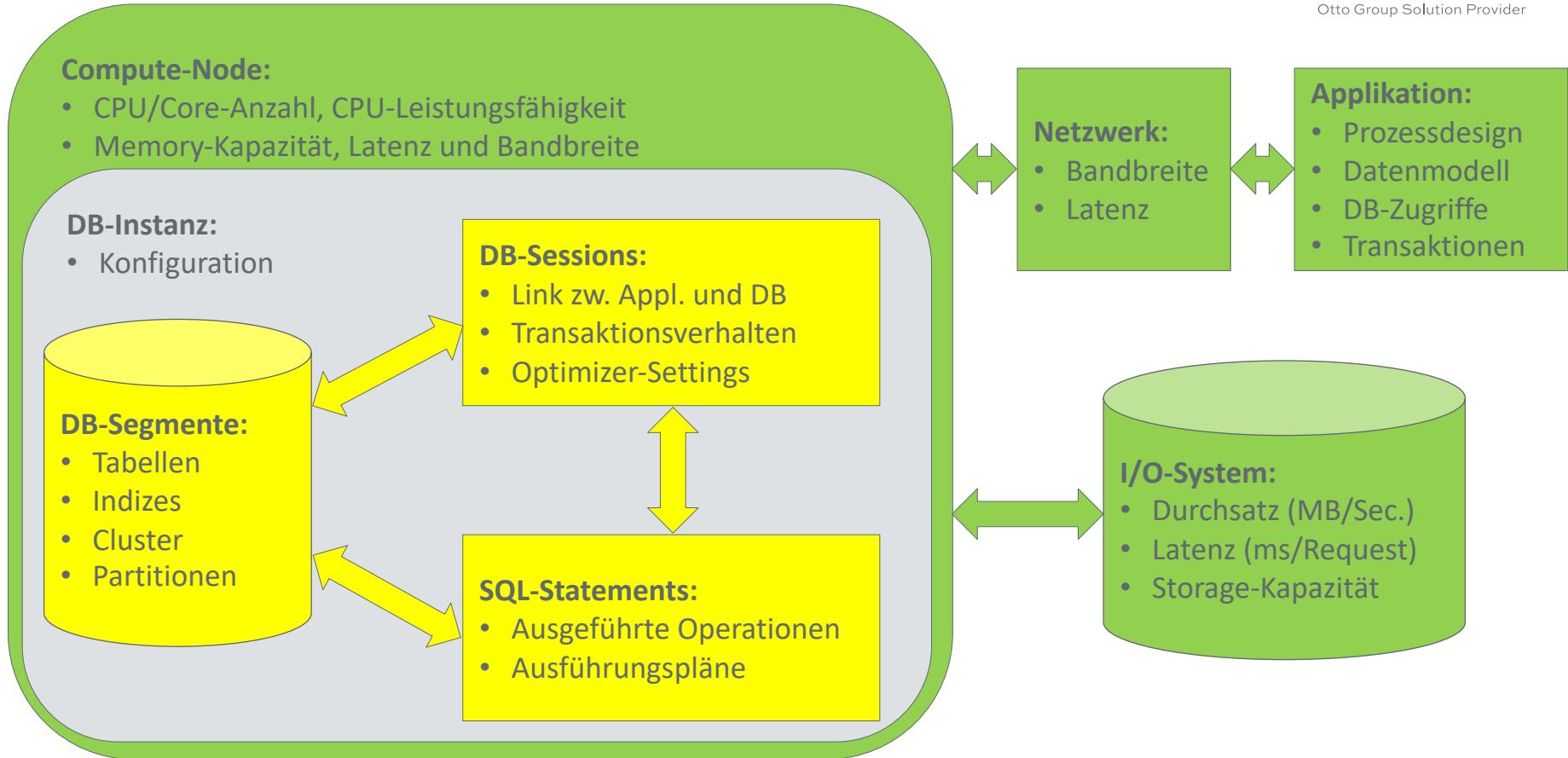
Teamleiter strategisch-technische Beratung bei OSP Dresden

> 30 Jahre Historie in IT-Projekten

Schwerpunkte:

- Entwicklung von OLTP-Systemen auf Basis von Oracle-Datenbanken
- Architektur-Beratung bis Trouble-Shooting
- Performance-Optimierung bestehender Systeme

# Einflussfaktoren auf Performance in DB-Nutzung



# Agenda

- Vorstellung des verwendeten Tools „Panorama“
- Top/Down-Analyse der Session-Aktivitäten
- Ausgeführte SQL-Statements mit Detail-Analyse
- Segment-Statistiken: Aufzeichnung von Kennwerten für Objekte (Tabellen, Indizes ...)
- Kapazität und Auslastung von CPU, Memory und I/O-System, Konfiguration der Instance
- Rasterfahndung: Systematischer Scan des Systems auf Performance-Antipattern

# Agenda

- Vorstellung des verwendeten Tools „Panorama“
- Top/Down-Analyse der Session-Aktivitäten
- Ausgeführte SQL-Statements mit Detail-Analyse
- Segment-Statistiken: Aufzeichnung von Kennwerten für Objekte (Tabellen, Indizes ...)
- Kapazität und Auslastung von CPU, Memory und I/O-System, Konfiguration der Instance
- Rasterfahndung: Systematischer Scan des Systems auf Performance-Antipattern

# Panorama for Oracle databases



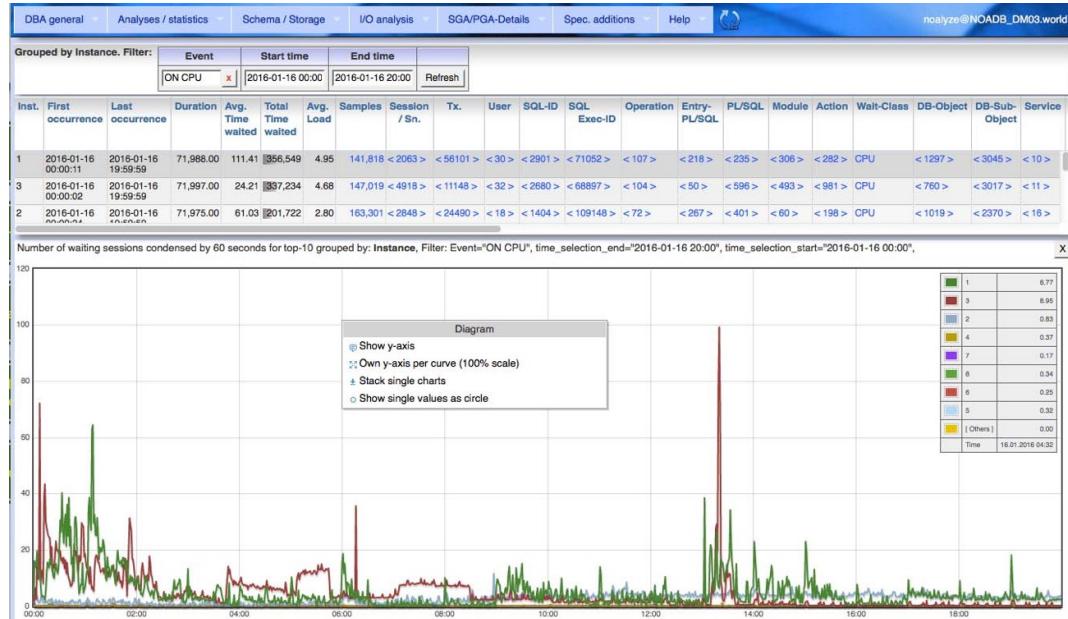
Freies verfügbares Werkzeug für Performance-Analyse von Oracle-DB

Basierend u.a. auf AWR-Daten oder eigenem Sampling (auch mit SE bzw. ohne Diagnostics Pack)

Docker-Image oder selbststartendes war-File

Panorama greift nur rein lesend auf Ihre DB zu und installiert keine eigenen PL/SQL-Objekte. Sie können die Funktionen also ohne Risiko testen.

Beschreibung von Panorama inkl. Download-Link:  
Blog zum Oracle-Performance-Analyse:  
Diverse Slides zur Anwendung von Panorama:

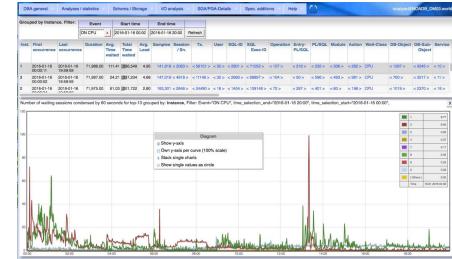


<https://rammpeter.github.io/panorama.html>  
<https://rammpeter.blogspot.com>  
<https://www.slideshare.net/PeterRamm1>

# Panorama für Oracle: Motivation

## Schwerpunkt dabei:

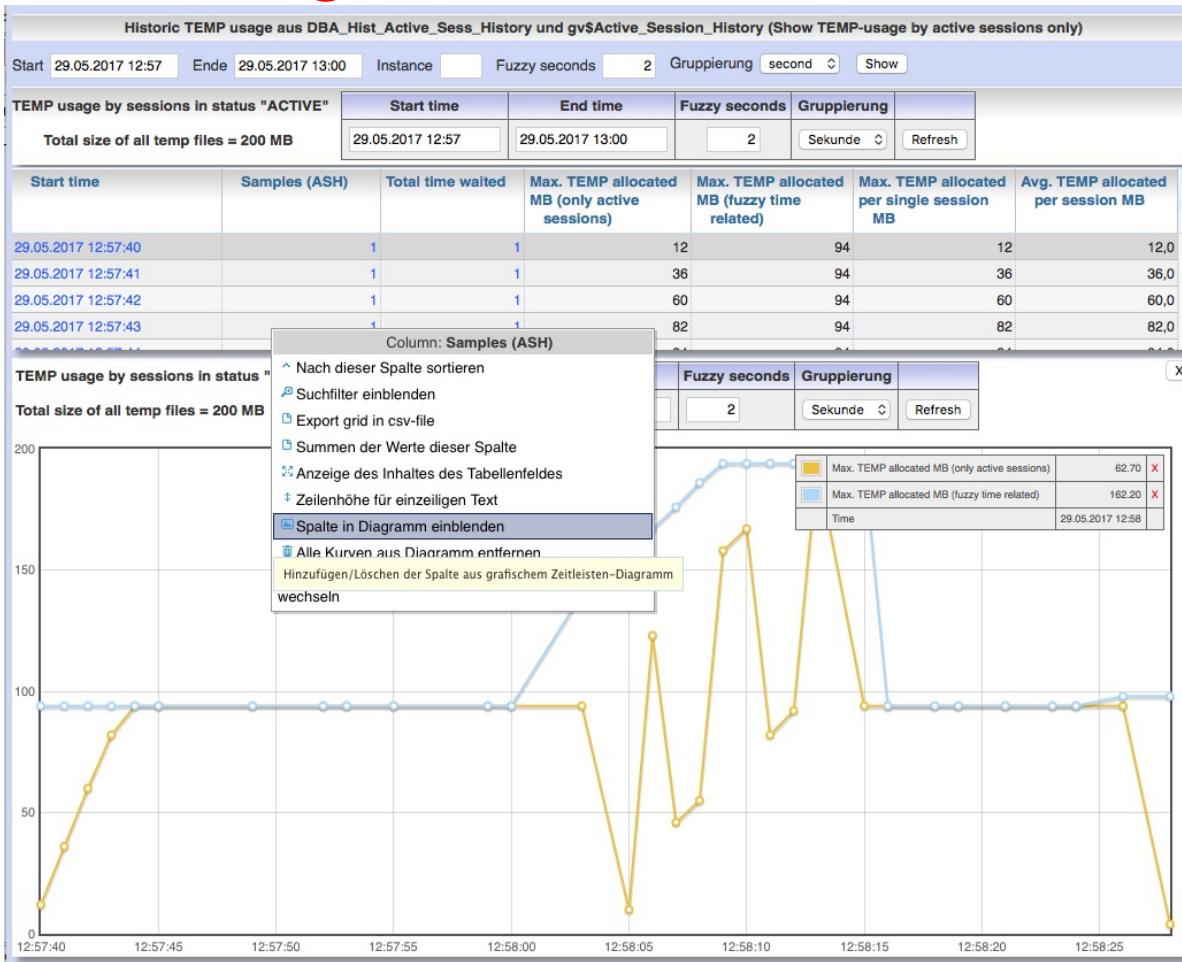
- Aufbereitung komplexer Zusammenhänge der DB ohne tiefes Insider-Wissen
- Unterstützung eines Analyse-Workflow durch Verknüpfung der einzelnen Schritte auf Web-GUI als Alternative zu liebevoller Sammlung einzelner SQL-Scripte
- Drilldown im Ursachenermittlung ausgehend von konkreten Problempunkten
- Offline-Analyse mit zeitlicher Distanz zum zu untersuchenden Problem
- Senken der Hemmschwelle, Problemen tatsächlich detailliert bis auf den Grund zu gehen



## Abgrenzung zu etablierten Monitoring-Tools:

- Panorama hat nicht den Anspruch, alle Facetten des Monitoring und der Visualisierung von Interna der Oracle-DB abzudecken
- Funktionen fanden i.d.R. Aufnahme in Panorama, wenn sie:
  - von den etablierten Tools nicht bzw. nur unzureichend angeboten werden
  - Existierende Werkzeuge für Normalanwender nicht erreichbar sind (z.B. wegen Kosten)
- Eine sinnvolle Anwendung erfolgt nicht anstatt, sondern in Kombination mit z.B. EM Cloud Control etc.

# Darstellungsformen



Workflow in Browser-Seite fortlaufend nach unten

Ergebnisse oft als Tabellen

Zeitbezogene Werte sind generell auch als Graphen darstellbar indem Spalten der Tabellen ein- und ausgeblendet werden

Visualisierung in Graphen über Kontext-Menü (rechte Maustaste)

Environment-Variable  
PANORAMA\_LOG\_LEVEL=debug  
zeigt ausgeführte SQLs auf Konsole

# Agenda

- Vorstellung des verwendeten Tools „Panorama“
- Top/Down-Analyse der Session-Aktivitäten
- Ausgeführte SQL-Statements mit Detail-Analyse
- Segment-Statistiken: Aufzeichnung von Kennwerten für Objekte (Tabellen, Indizes ...)
- Kapazität und Auslastung von CPU, Memory und I/O-System, Konfiguration der Instance
- Rasterfahndung: Systematischer Scan des Systems auf Performance-Antipattern

# Aktuell mit DB verbundene Sessions

Menü: „DBA Allgemein“ / „Sessions“

Anzeige Sessions

Nur Aktive  Parallel Query  Nur User  Nur DB-Link  Instance  Filter  Sessions anzeigen

Sessions																				
I	Con. ID	SID/SN	Status	SQL_ID	Wait event	Wait time	User	Proc	Machine	C-User	C- Proc	Program	Client- Info	Client ID	Module	Action	Service	Logon Time	Last Active Start	W
1	3	379, 62186	ACTIVE	45muknb7u586h	ON CPU	0,0	GRPAROBO	12780	OCNSVMEFI1034	OG-FKT- GROBO03	17256:1	pcselekt.exe	PCSEL	SID = 379, Serial# = 62186	pcselekt.exe		client.ov.otto.d	07.05.2021 08:13:06	07.05.2021 08:32:03	
1	3	1531, 14503	ACTIVE		SQL*Net message from	0,0	SUPPLIERCOM	3885	mercxfit1.ov.otto.de	buying	1234	JDBC Thin Client		SID = 1531, Serial# =	JDBC Thin Client		mercadohgv.o	07.05.2021 08:04:44	07.05.2021 08:33:50	

Details zu Session SID=379, Serial#=62186, Instance=1, CDB=3 (EKR30PS), 07.05.2021 08:34:01

Status	Client- Info	Module	Action	Username	Audit- SID	Sh. Proc.	PID	Machine	OSUser	Process	Program	Logon- time	Last active start	Tx. ID	Auth. -type	Client char set	Client connection	Client OCI lib	Client version	Client driver
ACTIVE	PCSEL	pcselekt.ex		GRPAROBO	195190559	12780	2561	OCNSVMEFI1034	OG-FKT- GROBO03	17256:17620	pcselekt.exe	07.05.2021 08:13:06	07.05.2021 08:32:04		DATABASE	UTF16	Heterogeneous	Home- based	11.2.0.1.0	

SQL-ID      C.      SQL exec start      SQL exec ID      SQL-Text

Aktuelles SQL-Statement	1zc46m5xpfxw	5 07.05.2021 08:33:57		28.375.751	SELECT TEXT FROM EINKAUF.II18N_BESCHREIBUNG IB WHERE IB.ID_II18N = :B3 AND IB.TEXT_KZ = :B2 AND RTRIM(IB.SPRACH_KZ) = :B1
Vorheriges SQL-Statement	1zc46m5xpfxw	5 07.05.2021 08:33:57		28.375.750	SELECT TEXT FROM EINKAUF.II18N_BESCHREIBUNG IB WHERE IB.ID_II18N = :B3 AND IB.TEXT_KZ = :B2 AND RTRIM(IB.SPRACH_KZ) = :B1

Process Memory (incl. PQ-Server) from GV\$Process\_Memory

Category	Allocated	Used	Max Allocated
SQL	11.928.776	0	587.463.016
Other	96.181.509		96.181.509
PL/SQL	169.960	0	178.312
Freeable	8.257.536	0	

Wait-Status   Locks   Temp-Usage   41 offene Cursoren   Objects accessed   Active Session History   Session-Statistics   Audit Trail   Optimizer Env.   SQL-Monitor (4)

Anlistung von DB-Sessions mit optionalem Filter und Drilldown in Session- und SQL-Details

# Rückbetrachtende Analyse des Workload

- Die Bewertung und Analyse des interessierenden Workload einer DB ist selten live möglich
  - I.d.R. erfolgt eine Eskalation mehr oder weniger zeitversetzt zum auslösenden Ereignis
  - Aus der aktuellen Lastsituation einer DB lässt sich nicht auf die Vergangenheit schließen
- 
- Einiges über ausgeführte SQLs lässt sich noch aus der SGA ermitteln (v\$SQL ...) sowie z.B. über StatsPack

## Oracle's builtin-Lösung: „Active Workload Repository“ (AWR) und „Active Session History“ (ASH)

- Historisierung diverser Betriebszustände out of the box
- AWR: Aufzeichnung im Stundentakt mit Aufbewahrung für 7 Tage (anpassbar)
- ASH: Aufzeichnung aktiver Sessions im Sekundentakt mit Verdichtung auf 10 Sekunden
- Verfügbar nur für Enterprise Edition mit zusätzlicher Lizensierung des „Oracle Diagnostics Pack“
- Vorsicht: Auch zugreifbar ohne Lizenz aber Zugriff bedeutet Lizenzverletzung

# Und was ist mit Standard Edition bzw. ohne Diagnostics Pack?

Panorama bietet eine eigene Aufzeichnung des Workload (Panorama-Sampler):

- Strukturidentisch zu Oracle's AWR- und ASH-Tabellen
- Nutzbar ohne EE / Diagnostics Pack auch für Standard Edition bzw. Express Edition
- Aufzeichnung in Tabellen in lokalem Schema der DB
- Panorama nutzt transparent entweder AWR-Daten oder eigene Workload-Aufzeichnung
- Weitere Sampler-Funktionen die in AWR nicht enthalten sind:
  - Cache-Historie, Historie der Objektgrößen, Blocking Locks, Langzeittrend der Lasten

- Integriert in Panorama
- Aktivieren der Aufzeichnung per GUI
- Beliebige Anzahl DB-Instanzen

Database connect info	AWR and ASH	Object size history	DB-cache history	Blocking locks history	Long-term trend
Sample AWR and ASH	<input checked="" type="checkbox"/>				
Snapshot cycle (minutes)	60				
Snapshot retention (days)	30				
Limits for SQL-history	Min. executions	20	Min. runtime (ms)	200	
<input type="button" value="Test connection"/> <input type="button" value="Save"/> <input type="button" value="Cancel"/>					

Weitere Details zu Panorama-Sampler: [https://rammpeter.github.io/panorama\\_sampler.html](https://rammpeter.github.io/panorama_sampler.html)

# Funktionsumfang Panorama-Sampler

Für folgende AWR-Views mit Aufzeichnung der Historie existiert eine Entsprechung im Panorama-Sampler:

gv\$Active\_Session\_History  
DBA\_Hist\_Active\_Sess\_History  
DBA\_Hist\_Cache\_Advice  
DBA\_Hist\_Datafile  
DBA\_Hist\_Enqueue\_Stat  
DBA\_Hist\_FileStatXS  
DBA\_Hist\_IOStat\_Detail  
DBA\_Hist\_IOStat\_Filetype  
DBA\_Hist\_Log  
DBA\_Hist\_Memory\_Resize\_Ops  
DBA\_Hist\_OSStat

DBA\_Hist\_OSStat\_Name  
DBA\_Hist\_Parameter  
DBA\_Hist\_PGAStat  
DBA\_Hist\_Process\_Mem\_Summary  
DBA\_Hist\_Resource\_Limit  
DBA\_Hist\_Seg\_Stat  
DBA\_Hist\_Service\_Name  
DBA\_Hist\_Snapshot  
DBA\_Hist\_SQL\_Bind  
DBA\_Hist\_SQL\_Plan  
DBA\_Hist\_SQLStat  
DBA\_Hist\_SQLText

DBA\_Hist\_StatName  
DBA\_Hist\_Sysmetric\_History  
DBA\_Hist\_Sysmetric\_Summary  
DBA\_Hist\_System\_Event  
DBA\_Hist\_SysStat  
DBA\_Hist\_Tablespace  
DBA\_Hist\_Tempfile  
DBA\_Hist\_TempStatXS  
DBA\_Hist\_TopLevelCall\_Name  
DBA\_Hist\_UndoStat  
DBA\_Hist\_WR\_Control

# Warum Spuren hinterlassen für Analyse

Die Eskalation fachlicher Probleme erfolgt i.d.R. mit einer gewissen Latenz.  
Standard-Anfrage: Prozess XY lief gestern Nacht 3 x so lange wie üblich! Warum?

**Beispiel: Die Applikation arbeitet mit Java JEE-Application-Server und Session-Pooling:**

- Für jede Transaktion wird erneut willkürlich eine DB-Session aus dem Pool geholt
- Mit Rücksicht auf den OLTP-Charakter des Systems sind Transaktionen sehr kurz
- Ein Prozess skaliert parallel und nutzt dabei eine variable Anzahl von DB-Sessions

**Wie sollen hier die Spuren in der Datenbank-Historie dem Prozess zuordnet werden?**

**Äußerst hilfreich für rückbetrachtende Analyse ist das Taggen von DB-Sessions mit fachlichen Kontextinformationen zum Prozess am Beginn einer Transaktion.**

Setzen der Kontext-Info erfolgt durch Ausführung der PL/SQL-Funktion:

**DBMS\_Application\_Info.Set\_Module(<Module>, <Action>)**

Diese Info wird mit in diversen Spuren der DB aufgezeichnet und erlaubt später die Zuordnung der DB-Aktivitäten zum auslösenden Prozess

# Active Session History

Menü "Analysen/Statistiken" / „Session Waits“ / „Historisch“

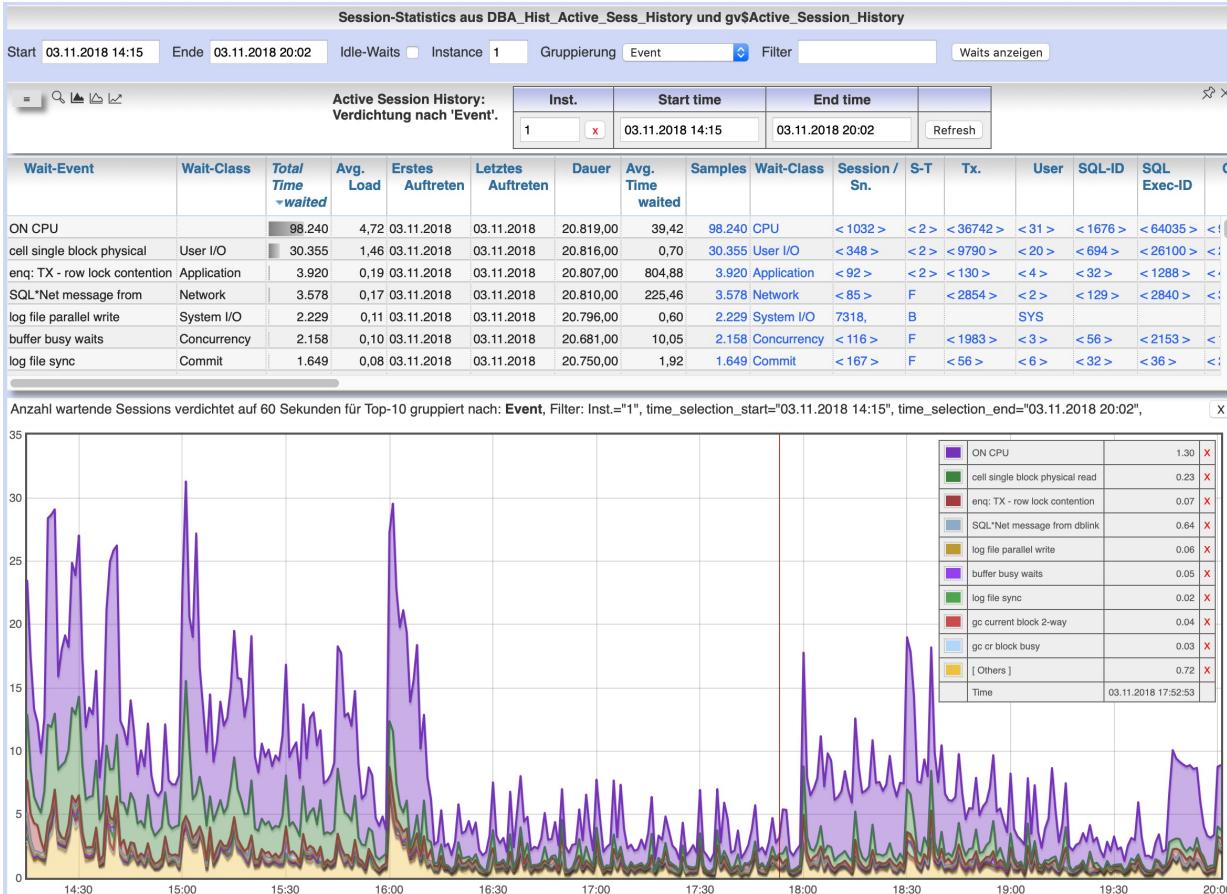


- Für mich der größte Schritt in den Analyse-Funktionen der Oracle-DB
- Eingeführt mit Oracle 10g, stark erweitert in 11g, noch etwas in 12c
- Historisierung der Daten aller aktiven Sessions der DB aus V\$Session ++
- Ablage je Sekunde in SGA-Memory, Abfrage per View V\$Active\_Session\_History
- Persistierung jeder 10. Sekunde im Zyklus der AWR-Snapshots in AWR-Tabelle, Abfrage per View DBA\_Hist\_Active\_Sess\_History
- In AWR-Tabelle aufbewahrt analog der weiteren AWR-Daten, Default 7 Tage
- Auf dieser Datenbasis lassen sich sehr exakt Betriebszustände der DB auch noch nach langer Zeit rekonstruieren

Nutzung erfordert Enterprise Edition +  
Lizensierung der Option ‚Diagnostics Pack‘

# Active Session History

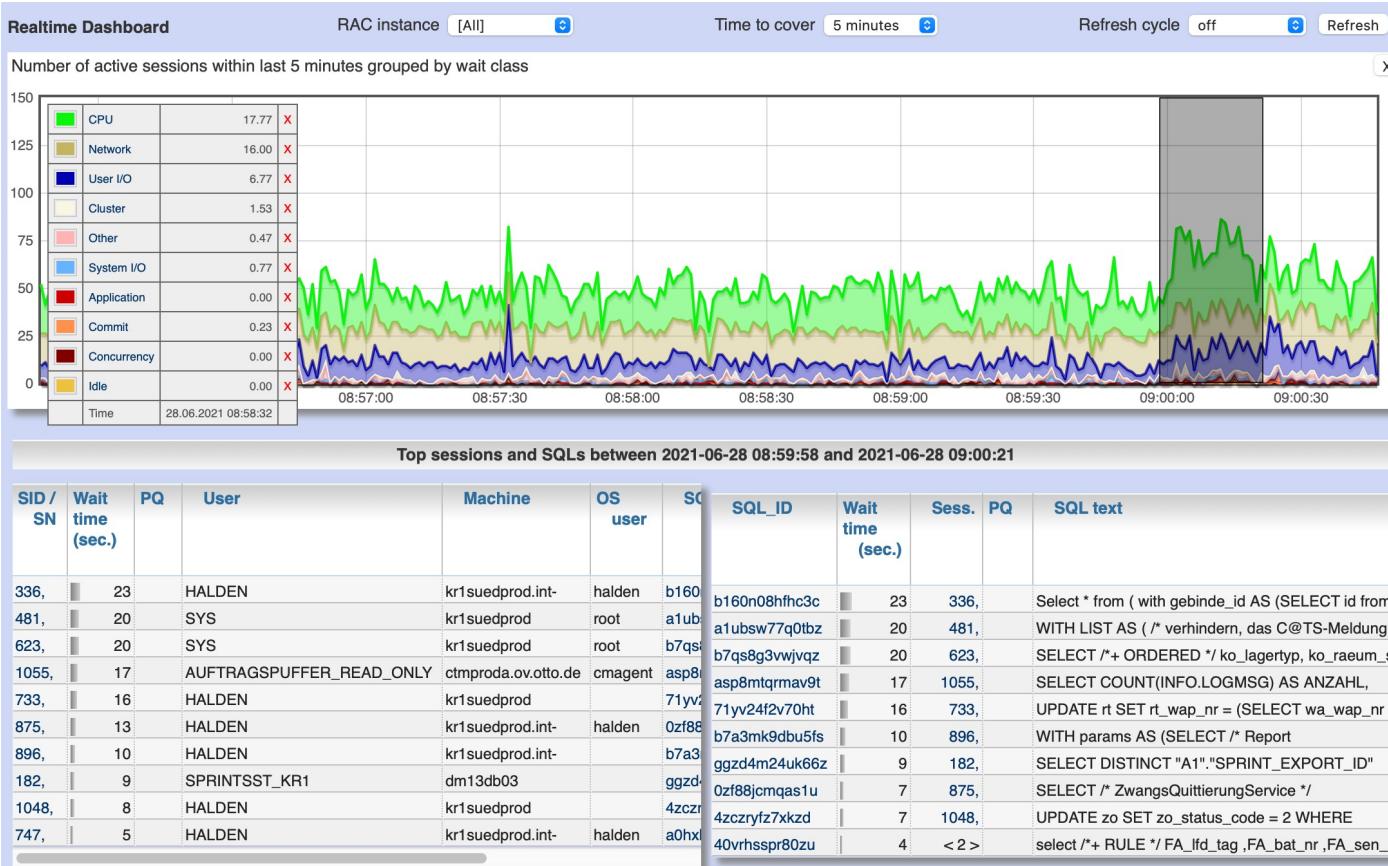
Menü "Analysen/Statistiken" / „Session Waits“ / „Historisch“



- Auswahl Zeitraum und Einstiegsgruppierung
- Sukzessiver Drilldown nach diversen Kriterien
- „< xxx>“ zeigt Anzahl unterschiedlicher Werte der Kategorie-Spalte in aktueller Selektion
- Click auf „< xxx>“ gruppiert die Werte der aktuellen Zeile nach dieser Kategorie
- Drilldown bis auf einzelne ASH-Sample-Records

# ASH: Real-time Dashboard

Menu “DBA Allgemein” / „Dashboard“



- Real-time Monitoring der aktiven Sessions nach Wait Class
- Automatischer Refresh im gewählten Zyklus
- Top SQLs und Top Sessions für den gewählten Zeitraum
- Drilldown über Wait Class (Klick in Legende), Session oder SQL

# ASH: Rückblickende Analyse blockierender Locks

Menü „DBA Allgemein“ / „DB-Locks“ / „Blocking Locks historisch aus ASH“

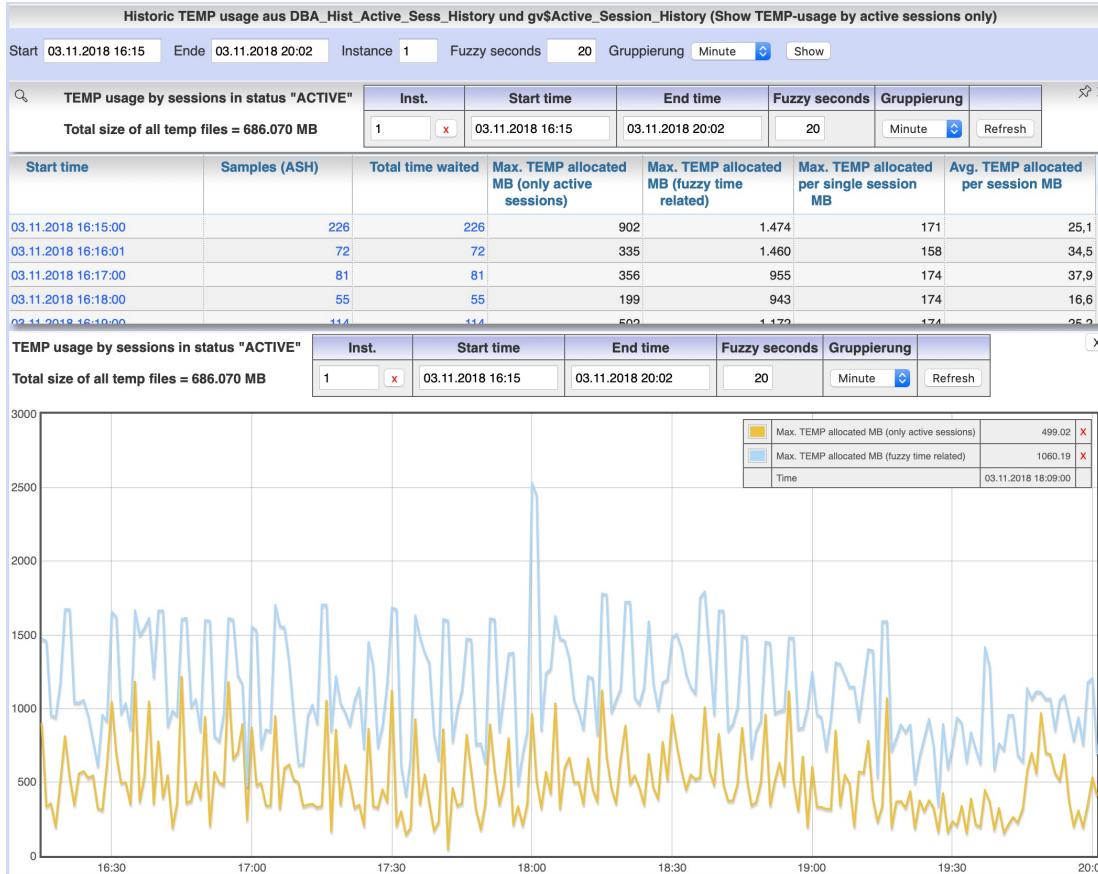


Blocking Locks aus DBA_Hist_Active_Sess_History																	
Start	03.11.2018 16:15	Ende	03.11.2018 20:02	Blocking Locks anzeigen													
Blocking Locks zwischen 03.11.2018 16:15 und 03.11.2018 20:02 hierarchisch gruppiert ausgehend von Root-Blockern																	
First occ.	Last occ.	Dead lock	Samples direct	B. I.	B.SID	B.User	B.SQL-ID	B.Event	Blocking Module / Action / Program	Direct Blocked	Total Blocked	Max. depth	Total Wait (sec.)	Waiting (sec.)	Blocking Object		
03.11.2018	03.11.2018		7.424	GLOBAL				INACTIVE		< 1795 >	< 1809 >	11	54,3	7.678	< 27 >,< 416 >		
03.11.2018	03.11.2018		1.156	1	6846	SYS		< 5 >	oracle@dm10db01	< 123 >	< 123 >	2	0,1	1.157	< 10 >,< 30 >		
03.11.2018	03.11.2018		1.136	2	6846	SYS		< 4 >	oracle@dm10db02	< 57 >	< 57 >	1	0,0	1.136	< 9 >,< 24 >		
03.11.2018	03.11.2018	Y	1.069	4	6451	SYS		< 4 >	oracle@dm10db04	< 55 >	< 56 >	2	0,0	1.073	< 8 >,< 23 >		
03.11.2018	03.11.2018	Y	262	1	4109	NOA	< 11 >	< 6 >	< 3 >< 2 > JDBC	< 45 >	< 51 >	3	30,7	287	< 4 >,< 8 >		
03.11.2018	03.11.2018		267	4	1549			INACTIVE		< 4 >	< 4 >	1	0,2	267			
03.11.2018	03.11.2018	Y	134	1	6410	NOA	< 5 >	< 6 >	< 2 > Store Statistics	< 18 >	< 26 >	10	90,8	203	< 5 >,< 5 >		
03.11.2018	03.11.2018	Y	178	1	4978	NOA	< 33 >	< 6 >	< 5 >< 2 > JDBC	< 41 >	< 43 >	4	50,1	185	< 5 >,< 7 >		
03.11.2018	03.11.2018	Y	149	1	5541	NOA	< 25 >	< 9 >	< 3 >< 4 > JDBC	< 44 >	< 52 >	5	35,7	177	< 6 >,< 9 >		
03.11.2018	03.11.2018		152	1	5058	NOA	< 10 >	< 6 >	< 5 >< 4 > JDBC	< 28 >	< 47 >	3	30,3	175	< 4 >,< 7 >		
03.11.2018	03.11.2018	Y	121	1	5032	NOA	< 3 >	< 5 >	< 3 >< 3 > JDBC	< 24 >	< 43 >	5	30,4	174	< 3 >,< 3 >		
03.11.2018	03.11.2018	Y	135	1	4539	NOA	< 10 >	< 7 >	< 6 >< 3 > JDBC	< 35 >	< 43 >	7	28,2	159	< 5 >,< 7 >		
03.11.2018	03.11.2018	Y	99	1	6378	NOA	< 11 >	< 8 >	< 5 >< 4 > JDBC	< 25 >	< 27 >	3	54,9	158	< 5 >,< 8 >		
03.11.2018	03.11.2018	Y	123	1	6965	NOA	< 12 >	< 5 >	< 3 >< 4 > JDBC	< 28 >	< 42 >	5	30,3	142	< 4 >,< 8 >		
03.11.2018	03.11.2018		121	1	5003	NOA	< 15 >	< 7 >	< 4 >< 4 > JDBC	< 25 >	< 34 >	6	20,4	136	< 5 >,< 11 >		
03.11.2018	03.11.2018		85	1	5059	NOA	< 4 >	< 5 >	Module, Action und Programm der blockenden Session Module = < 3 > Action = < 4 > Program = JDBC Thin Client					3	30,4	133	< 4 >,< 4 >
03.11.2018	03.11.2018	Y	103	1	6384	NOA	< 14 >	< 7 >				5	18,7	131	< 4 >,< 9 >		

- Zeigt eine Zeile je Blocking-Kaskade auslösende Session (root blocker)
  - Sortiert nach Wartezeit aller direkt oder indirekt geblockten Sessions
  - Drilldown in:
    - Geblockte Sessions
    - ASH der blockierenden Session
    - ASH der blockierten Sessions
    - Blockendes Objekt bis auf Primary Key des Records

# ASH: Rückblickende Analyse der TEMP-Nutzer

Menü „Schema / Storage“ / „TEMP-usage“ / „Historisch aus ASH“



Erkennen der Nutzer von TEMP-Tablespace inkl. Verursacher eines evtl. “unable to extend temp segment“

- Darstellung der TEMP-Nutzung über Zeit in Diagramm
- Ermitteln des relevanten Zeitpunktes
- Wechsel in ASH-Analyse zur Ermittlung der konkreten Sessions nach TEMP-Verbrauch zum Zeitpunkt

Siehe auch: [Blog-Post zum Thema](#)

# Agenda

- Vorstellung des verwendeten Tools „Panorama“
- Top/Down-Analyse der Session-Aktivitäten
- Ausgeführte SQL-Statements mit Detail-Analyse
- Segment-Statistiken: Aufzeichnung von Kennwerten für Objekte (Tabellen, Indizes ...)
- Kapazität und Auslastung von CPU, Memory und I/O-System, Konfiguration der Instance
- Rasterfahndung: Systematischer Scan des Systems auf Performance-Antipattern

# SQL-Historie

Menü "SGA/PGA Details" / „SQL-Area“ / ..

SQL of current SGA from GV\$SQLArea, grouped by SQL-ID

I	SQL-ID	SQL-Text	C	V	P	Last active	User	Parse	Execs	Elapsed	Ela./Ex.	CPU	Disk Reads	Disk/Ex.	Buffer Gets	Buffer/Ex.	Rows proc
1	6wg9h22z83d0y	SELECT ID, ID_DISCRIMINATOR,	1	2	1	2018-11- EKS	EKS		24,458,911	155,832	0.0064	134,901	437,947	0	117,746,970	5	10,51
1	9tpccq6wqc6s7	SELECT S.SID FROM V\$SESSION S	1	3	1	2018-11- DBAITEWKD	DBAITEWKD		63,563,724	150,815	0.0024	131,043	71	0	2,215	0	63,56
1	1tyjz7ucuuq5	BEGIN :1 :=	1	1	1	2018-11- EKS	EKS		246,232	123,718	0.5024	107,186	205,302	1	1,117,368,294	4,538	241

Statement details of current SGA from GV\$SQL: Instance = 1, SQL-ID = '6wg9h22z83d0y', Child-No. = 0

```
/* single line SQL-text formatted by Panorama */
SELECT ID, ID_DISCRIMINATOR, ID_EDITORIALNODE, TSUPDATE, ID_LOGICALPRODUCT, OBJECTID
FROM BUYING.EDITORIALLINK
WHERE ((OBJECTID = :1) AND (ID_DISCRIMINATOR = :2))
```

Parsing schema name	EKS	
Plan-Hash-Value	2489798738	
Child-address	0000003ED664198	
Optimizer Env Hash-Value	2384849091	
Parsing module	JDBC Thin Client	
Parsing action		
Object status	VALID	
PL/SQL program / line	:1 :0	
First Load Time	2018-10-14 14:00:20	
Last Load Time	2018-11-09 05:00:25	
Last Active Time	2018-11-09 11:57:05	
Buffer cache hit ratio	99.63	
Metric-Name	Total	Per Exec.
Executions	24,459,973	-
Parse calls	2,543,252	0.10
Fetches	24,453,828	1.00
Sorts	0	0.00
Rows processed	10,515,132	0.43
Buffer gets	118,494,428	4.84
Disk reads	437,957	0.02
Elapsed Time (sec), incl. parallel Query Slaves	155,837.21	0.0064
CPU-Time (sec)	134,906.57	0.0055
Application Wait Time (sec)	0.00	0.0000
Concurrency Wait Time (sec)	16.66	0.0000
Cluster Wait Time (sec)	0.00	0.0000
User-IO Wait Time (sec)	1,031.38	0.0000
PL/SQL Exec Time (sec)	0.00	0.0000

Execution-Plan (1) Bind variables (2) Objects (2) Full history Cursor Sharing (1 versions) Active Session History Open Cursor (81) DBMS\_XPLAN SQL-Monitor (0) SQL-Patch

Explain Plan of Child=0 parsed at 2018-11-09 05:00:25, Optimizer-Mode=ALL\_ROWS, Executions=24,462,287, first ASH-Sample in SGA from 2018-11-09 07:54:19

ID	R.	Object-name	Rows	MB	Cost	Card.	Parallel	Access	Filter	Temp est.	Temp max.	DB time	CPU	Waits	I/O	IC	Dist	PGA max.	Proj.	Starts	Rows	Cost
0	3									3	0.9	0.9	1.0	1.0					23			
1	2	BUYING.EDITORIALLINK	3,959,392	183	1,308	1				3	0.2	0.1	36.4	0.2	0.2				23	"ID"		
2	1	BUYING.IX_EDTLINK_LOGPK	3,959,392	155	1,307	1		"ID DISCRIMINATOR"	"(OBJECTID" = :1 AND	3	98.9	99.0	63.6	98.9	98.9				25	"EDITORIALLIN		

Auswertung der aktuell in SGA befindlichen SQLs sowie der Historie aus AWR

Anlistung der SQLs sortiert nach diversen Kriterien

Details je SQL inkl.

- Execution Plan
- Bindevariablen
- Child-Cursoren
- Komplette Historie aller AWR-Snapshots des SQL
- Gründe für multiple Cursoren
- SQL-Monitor Aufzeichnungen
- Erstellung Baseline / SQL-Patch

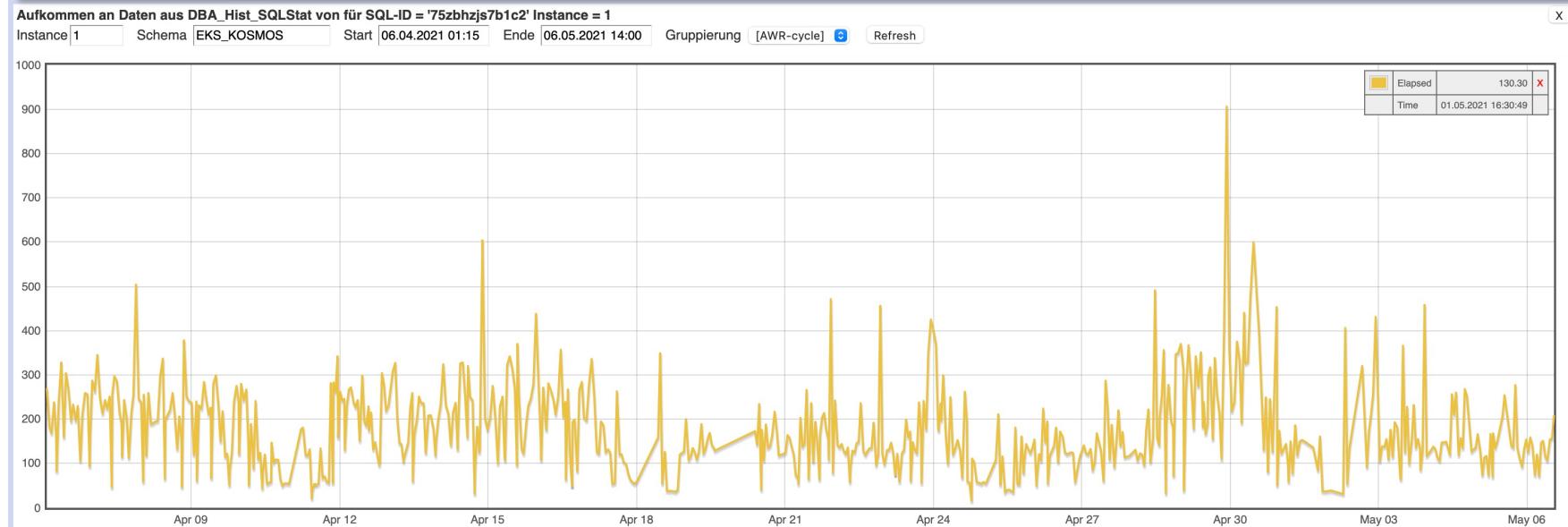
# Details zu SQL-Statement: Execution Plan



Explain Plan of Child=13 parsed at 02.05.2021 09:53:19, Plan_Hash_Value = 4146904004, Optimizer-Mode=ALL_ROWS First ASH-Sample in SGA is from 06.05.2021 09:58:34																				
Operation	ID	R.	Object-name	Rows	MB	Cost	Card.	Parallel	Access	Filter	Temp est.	Temp max.	DB time	CPU	Waits	I/O	IC	PGA max.	Proj.	
SELECT STATEMENT	0	41				1.165.531														
HASH UNIQUE	1	40				1.165.531	1		Parallelization (from Other_Tag)			0	0,3	1,1		0,5	0,0	60 "MS"		
NESTED LOOPS SEMI	2	39				1.165.530	1					0	0,5	1,7		0,6	0,0	61 "MS"		
HASH JOIN RIGHT SEMI	3	36				1.111.661	17.221		"IO"."ID_STATEATTRIBUTEQUALIT			0	0,2	0,6		0,2	0,0	60 (#key)		
TABLE ACCESS BY INDEX ROWID BATCHED	4	2	SYSPEKS.STATEATTRIBUTEQUALITY	11	0	2	10									"IO_E				
INDEX RANGE SCAN	5	1	SYSPEKS.IX_ATTRQUALTY_CODE	11	0	1	10		"IO_SAQ"."CODE">>1							"IO_E				
HASH JOIN RIGHT SEMI	6	35				1.111.659	17.395		"IO"."ID_PROCESSINGSYSTEMCO			0	0,2	0,6		0,3	0,0	60 (#key)		
VIEW	7	7	SYSPEKS.INDEX\$_JOINS_012			1	3		("IO_PROC"."CODE" = 5 OR							(rows				
HASH JOIN	8	6							ROWID = ROWID							(#key				
INLIST ITERATOR	9	4														ROW				
INDEX UNIQUE SCAN	10	3	SYSPEKS.IX_PROCSYSCOD_LKEY	14	0	0	3		("IO_PROC"."CODE" = 5 OR							ROW				
INDEX FAST FULL SCAN	11	5	SYSPEKS.PKEY_PROCSYSCOD	14	0	1	3									ROW				
NESTED LOOPS	12	34				1.111.658	52.191								"MS"					
NESTED LOOPS	13	32				1.111.658	169.288					0	0,2	0,6		0,3	0,0	60 "MS"		
HASH JOIN SEMI	14	30				1.034.872	24.184		"IC"."ID_COUNTRY" = "C"."ID" AND			23	0	1,5		5,7		2,8	0,2	61 (#key)
HASH JOIN	15	28				981.763	210.867		"MS"."ID" =			19							(#key	
HASH JOIN RIGHT SEMI	16	26				962.611	211.434		"IEP"."ID_SHOPDOMAIN" =			17							(#key	
TABLE ACCESS FULL	17	8	BUYING.ITEMPERMISSION	76.783.942	3.840	134.796	738.307		(IP"."ID_SHOPDOMAIN" = 83			0	20,1	9,2		24,0		6,8	5,1	14 (rows
HASH JOIN RIGHT SEMI	18	25				823.735	811.492		"I"."ID_STATEATTRIBUTEQUALITY"										(#key	
TABLE ACCESS BY INDEX ROWID BATCHED	19	10	SYSPEKS.STATEATTRIBUTEQUALITY	11	0	2	10									"I_SA				
INDEX RANGE SCAN	20	9	SYSPEKS.IX_ATTRQUALTY_CODE	11	0	1	10		"I_SAQ"."CODE">>1							"I_SA				
HASH JOIN	21	24				823.727	819.689		"IE"."ID_ITEM" = "I"."ID"			45							(#key	
HASH JOIN RIGHT SEMI	22	22				759.174	819.690		"IE"."ID_PROCESSINGSYSTEMCO										(#key	
VIEW	23	15	SYSPEKS.INDEX\$_JOINS_016			1	3		(IE_PROC"."CODE" = 5 OR							(rows				
HASH JOIN	24	14							ROWID = ROWID							(#key				
INLIST ITERATOR	25	12														ROW				
INDEX UNIQUE SCAN	26	11	SYSPEKS.IX_PROCSYSCOD_LKEY	14	0	0	3		("IE_PROC"."CODE" = 5 OR							ROW				
INDEX FAST FULL SCAN	27	13	SYSPEKS.PKEY_PROCSYSCOD	14	0	1	3									ROW				
HASH JOIN	28	21				759.153	3.005.528		"IEP"."ID_ITEMEFFORT" = "IE"."ID"			92	0	2,4		9,2	0,1	0,1	68 (#key)	
NESTED LOOPS	29	19				510.319	3.005.528								"C_1"					
TABLE ACCESS BY INDEX ROWID	30	17	SYSP.COUNTRY	248	0	1	1									"C_1.R				
INDEX UNIQUE SCAN	31	16	SYSP.IND_COUNTRY_CODEA2_UNIQUE	248	0	0	1		"C"."CODEA2" = 'GB'							"C_1.R				
TABLE ACCESS FULL	32	18	BUYING.ITEMEFFORTPERMISSION	312.574.93	14.526	510.318	3.005.528		UPPER("CODEA2") = 'GB'							"IEP"				
INDEX FAST FULL SCAN	33	20	BUYING.IX_ITEMEFFORT_BSKAT	85.175.000	5.307	180.601	43.031.10		(IEP"."ID_SHOPDOMAIN" = 83			0	30,9	33,9		29,8		27,4	87,7	40 "IEP"
TABLE ACCESS FULL	34	23	BUYING.ITEM	5.674.191	1.664	58.232	2.606.693		"IE"."ID_STATEDELETED" = 1			0	3,5	13,2		0,3	2,5	68 "IE"		
									"I"."ID_STATEATTRIBUTEQUALIT							0,1	0,0	0,0	84 (rows	

# Details zu SQL-Statement: AWR-Historie

		Aufkommen an Daten aus DBA_Hist_SQLStat von für SQL-ID = '75zbxzjs7b1c2' Instance = 1																								
Instance   1			Schema   EKS_KOSMOS			Start   06.04.2021 01:15			Ende   06.05.2021 14:00			Gruppierung   [AWR-cycle]			Refresh											
-Start	Erstes Auftreten	Letztes Auftreten	Plans	Plan hash value	Envs.	Opt. env hash value	Execs	Elapsed	Ela./Ex.	Ela./Row	CPU	App.	Con.	Clu.	I/O	PL/SQL	Disk Reads	Disk / Ex.	ms/Disk Read	Hit Ratio	Buffer Gets	Buffer / Ex.	Buffer / Row	Rows proc.	Rows / Ex.	Parses
06.04.2021 02:00:33	06.04.2021 02:00:33	06.04.2021 02:15:38	2	27993490	1	2313521473	1	270	269,8379	7	106	0	0	0	180	0	896.698	896.698,000	0,20	98,45	58.024.819	58.024.819,00	1.448,48	40.059	40.059,00	1
06.04.2021 03:15:58	06.04.2021 03:15:58	06.04.2021 03:30:03	2	27993490	1	2313521473	0	183	183,4556	5	82	0	0	0	112	0	592.396	592.396,000	0,19	98,94	56.114.660	56.114.660,00	1.400,80	40.059	40.059,00	0
06.04.2021 04:15:18	06.04.2021 04:15:18	06.04.2021 04:30:23	2	27993490	1	2313521473	1	169	168,6823	4	84	0	0	0	93	0	320.389	320.389,000	0,29	99,45	58.379.402	58.379.402,00	1.457,34	40.059	40.059,00	1
06.04.2021	06.04.2021	06.04.2021	2	27993490	1	2313521473	1	238	237,5469	6	94	0	0	0	155	0	663.102	663.102,000	0,23	98,86	57.935.113	57.935.113,00	1.450,48	39.942	39.942,00	1



# Details zu SQL-Statement: SQL-Monitor

Enterprise Manager Database Active Report

Real-time SQL Monitoring: Ou8czymn2gpar Status ✓

Time Zone: Browser (GMT+02:00) ▾

### Overview

**General**

SQL Text: `SELECT PCSELEKT.ART.BEST_WG_FT_KZ FT, PCSELEKT...`  
Execution Started: 06.05.2021 16:58:25 GMT+02:00  
Last Refresh Time: 06.05.2021 17:01:07 GMT+02:00  
Execution Id: 16777216  
User Name: THEINS@EKR30PS  
Fetch Calls: 1

**Time & Wait**

Metric	Value
Duration	2,7 m
Database Time	2,97 m
PL/SQL & Java	0 s
Activity	100 %

**I/O**

Metric	Value
Buffer Gets	1.009K
I/O Requests	334K
I/O Bytes	2,6 GB

### Details

Plan Statistics Plan Notes

Plan Hash Value: 1020049330 Plan Size: 20 Lines

Operation	Object	Informati	Line ID	Timeline	Execut	Est. Rows	Rows	Mem (Max)	Temp (Max)	I/O Requests	Activity
SELECT STATEMENT			0		1						
SORT ORDER BY			1		1	18K					
HASH GROUP BY			2		1	18K					
FILTER			3		1	18K					
HASH JOIN OUTER			4		1	1.552	1,8 MB			5,03 %	
FILTER			5		1	1.552					
HASH JOIN OUTER			6		1	18K	1.552	1,8 MB		3,77 %	
NESTED LOOPS			7		1	18K	1.552				
NESTED LOOPS			8		1	55K	1.552				
HASH JOIN			9		1	55K	1.552	2,6 MB		0,63 %	
TABLE ACCESS BY INDEX ROWID BATCHED	ART		10		1	35K	1.008		3		
INDEX RANGE SCAN	IND_ART_6		11		1	20K	2.000		12		
INLIST ITERATOR			12		1	29M					

# Details zu SQL-Statement: Plan-Stabilität

Menü „SGA/PGA Details“ / „SQL Plan Management“

SQL plan management directives										
SQL profiles from DBA_SQL_Profiles										
Name	Category	Created	Last modified	Description	Type	Status	Force matching	Usages in SGA	Usages in history	SQL
co_e_7g7nxtsc978z3_1180579734	DEFAULT	28.02.2020 16:25:54	28.02.2020 16:25:54	co_e_7g7nxtsc978z3 1180579734 1684505828464809677, 1684505828464809677,	MANUAL	ENABLED	NO	1	1	SELECT ITEMcmp_id, ITEMcmptype
SYS_SQLPROF_0171e44aea53	DEFAULT	05.05.2020 12:04:07	05.05.2020 12:04:07		MANUAL	ENABLED	NO			INSERT INTO temp_ottoint WITH
SYS_SQLPROF_0171bb184c610	DEFAULT	27.04.2020 12:04:24	27.04.2020 12:04:24		MANUAL	ENABLED	NO			SELECT ABEST_WG_FT_KZ FT,
SYS_SQLPROF_014932983b35	DEFAULT	18.02.2020 15:02:05	18.02.2020 15:02:05		MANUAL	ENABLED	NO			SELECT TL.ID ,TL.LOGTYPE ,GETI

Existenz per SQL und  
als Übersicht :

- SQL-Profiles
- SQL Plan Baselines
- Stored Outlines
- SQL-Translations
- SQL-Patches

Für ausgeführte SQLs Generierung von PL/SQL-Schnipsel für Erzeugung von Direktiven:

- SQL Plan Baselines: Pinnen auf Basis bereits existierender Pläne per Plan\_Hash\_Value
- SQL-Patches: Injizieren von Optimizer-Hints in unveränderte SQL-Statements
- SQL-Translations: Vollständiges Ersetzen des SQL-Textes von unveränderte Statements

# Agenda

- Vorstellung des verwendeten Tools „Panorama“
- Top/Down-Analyse der Session-Aktivitäten
- Ausgeführte SQL-Statements mit Detail-Analyse
- Segment-Statistiken: Aufzeichnung von Kennwerten für Objekte (Tabellen, Indizes ...)
- Kapazität und Auslastung von CPU, Memory und I/O-System, Konfiguration der Instance
- Rasterfahndung: Systematischer Scan des Systems auf Performance-Antipattern

# Struktur-Info von Tabellen, Indizes, Packages etc.

Menü: „Schema/Storage“ / „Describe Object“

**Describe database object**

Object-Owner **cust** Object-Name **Customer** Object-Type [Alle] **Describe**

Description of TABLE CUST.CUSTOMER

Company dependent customer  
Komprimierung reduziert Footprint auf 60%

Columns of TABLE CUST.CUSTOMER

Col.Name	Type	Prec.	Sc.	N.	Def.	Distinct	Nulls	Avg. Len.	Density	Buckets	Histogram	Comments	LOB segment	EQ	EQJ	NEQ
ID	NUMBER	9	0	N		68.305.085	0	7	0,0000	1	NONE	Primary Key		24.729	84.737	0
ID_COMPANY	NUMBER	4	0	N		59	0	3	0,0000	59	FREQUENCY	Company to		48.698	37.917	0
ID_PERSON	NUMBER	9	0	N		68.305.085	0	7	0,0000	1	NONE	Person that		9.015	26.073	0
CREATIONDATE	DATE			N		43.819.008	0	8	0,0000	254	HYBRID	Date of		5.626	192	78
ID_LOCALE	NUMBER	4	0	N		13	0	3	0,0000	13	FREQUENCY	Locale		7.318	4.971	0
CUSTNO	VARCHAR2	10	Chars	N		39.424.000	0	10	0,0000	254	HYBRID	Company		12.343	1.714	0
CANCELLATIONDATE	DATE			Y		96.240	64.013.335	2	0,0000	254	HYBRID	Date when		7.322	0	0
ORDERCOUNTER	NUMBER	6	0	Y		2.533	12.500.973	3	0,0000	254	TOP-	Comments		119	0	0
LASTONLINEUPD	DATE			Y		46.915.584	7.174.628	8	0,0000	1	NONE	Letzte				
ID_SHOPDOMAIN	NUMBER	4	0	Y		88	44.031.316	3	0,0114	1	NONE	ShopDomain,				

Attributes of TABLE CUST.CUSTOMER

TS	Pct Free	Init. Trans	Initial extent (KB)	Rows	Size (MB) Table	Size (MB) Total	Extents	Blocks	Empty	Avg. Space	Chained	RowLen	Dg.	Cache	Part.	Sub-Part.	Created	Last Update
DATA02	10	8		68.305.085	4.659,75	18.598,94	598	596.448	0	0	0	56	1	N		2	08.03.2015 00:42:45	05.08.2015 08:08:08

6 Indexes 1 Primary Key 2 Check Constraints 4 References from 85 References to 12 Triggers 129 Dependencies 49 Grants DBMS\_METADATA DB-Cache

Sessions accessing SQLs Active Session History Segment statistics Size evolution

Detail-Info zu DB-Objekten mit Drilldown in Strukturen, Zugriffe, Abhängigkeiten etc.

# Überblick über Storage-Nutzung

Menü: „Schema/Storage“ / „Plattenspeicher-Nutzung“

Storage-Objekte für Datenbank NOADB\_PROCESS 10.05.2021 13:13

Summen total				
Typ	MB Total	MB Used	MB Free	% used
PERMANENT	10.984.443	8.798.588	2.185.855	80,1
TEMPORARY	904.921	901.246	3.675	99,6
UNDO	917.504	790.892	126.612	86,2
Redo-Logs outside FRA	196.608	196.608	0	100,0
Fast Recovery Area	9.216.000	4.619.981	4.596.019	50,1
<b>TOTAL</b>	<b>22.219.476</b>	<b>15.307.315</b>	<b>6.912.161</b>	<b>68,9</b>

Summen Netto in TS nach Segment-Typen

Segment-Type	MBytes
TABLE	4.646.224
INDEX	3.873.836
TYPE2 UNDO	790.940
LOBSEGMENT	265.753
SYSTEM STATISTICS	8.356
FLASHBACK LOG	4.005.273,60
CLUSTER	4.116
LOBINDEX	165
ROLLBACK	0

Usage of fast recovery area (FRA)  
total FRA size = 9.000,00 GB  
not reclaimable usage = 31,55 %

File type	MB used	% used	% recl.
FLASHBACK LOG	4.005.273,60	43,46	18,58
ARCHIVED LOG	614.707,20	6,67	0,00

Schema-Usage

Schema	Table	Index	Index IOT-PKey	TYPE2 UNDO	SYSTEM STATISTICS	ROLLBACK	Total MB
AUFTRAG	2.008.695	2.039.635	26.900				<b>4.075.230</b>
CUST	298.000	660.218	601.197				<b>1.559.414</b>
SYS	66.088	34.178	24	790.916	8.356	0	<b>899.563</b>
LOGISTIK	375.962	246.873	0				<b>622.835</b>
WHR	130.458	293.638	4.721				<b>428.817</b>
ITM	261.024	98.040	16.256				<b>407.120</b>

Tablespace-usage per schema

Schema	Tablespace	Used (MB)	Quota (MB)	Charged (MB)
AUFTRAG	INDEXSP01	2.032.607		
AUFTRAG	DATASP01	2.005.260		
CUST	DATA02	561.713		
CUST	INDEX02	376.437		
CUST	DATA01	310.975		
ITM	DATA02	269.253		
CUST	INDEX01	257.482		

Tablespace-usage total from DBA\_Tablespace

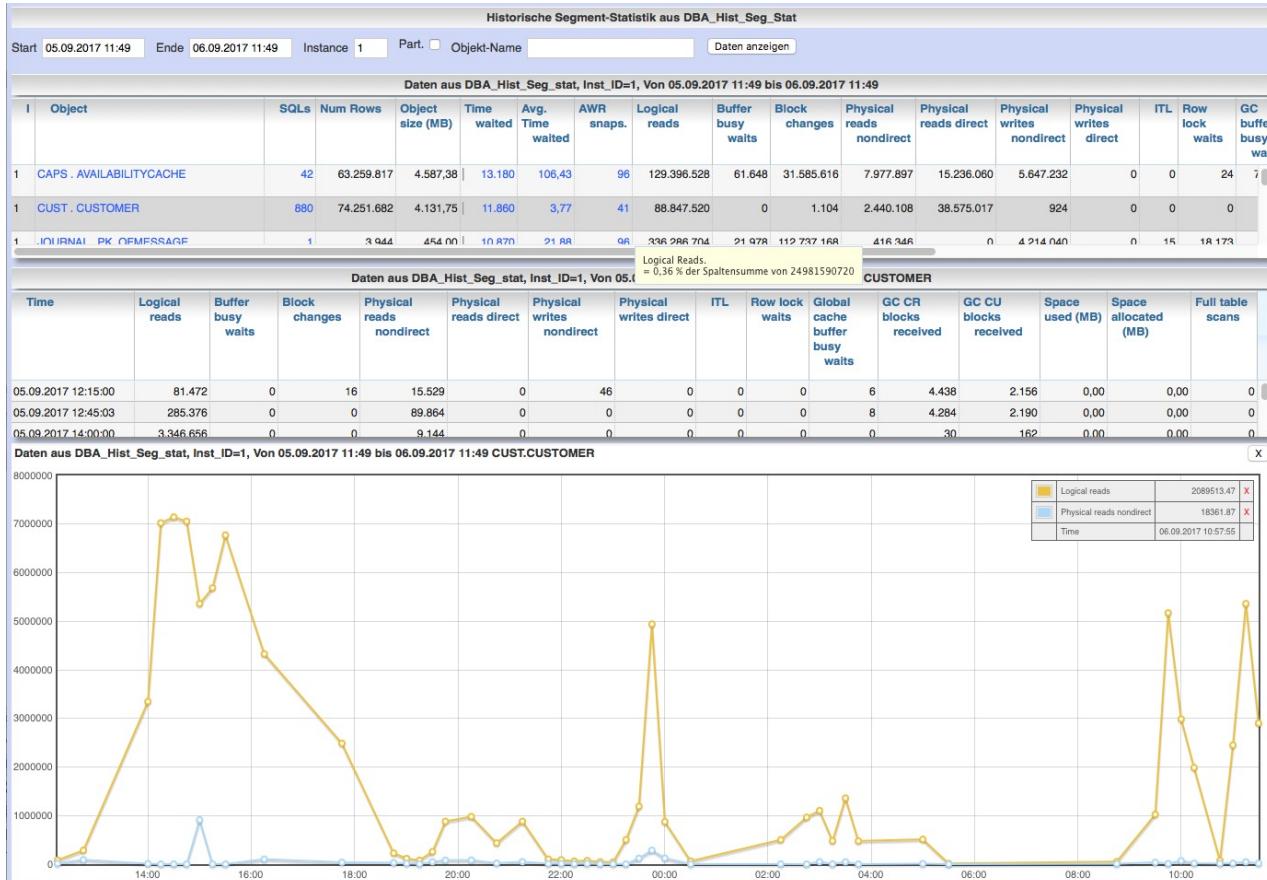
Tablespace	Contents	Block size	MB Total	MB Used	MB Free	% used	Auto ext.	MB max.	% used max.	Files	Status	Logging	Force log.	Extent mgmt.	Allocation Type	Pl. in	Segment Space Mgmt.	Def. table compress	Big file	Encrypt.	Def. in memory	Details
CRYPT_ALL	PERMANENT	8.192	307.200	<b>231.400</b>	<b>75.801</b>	75,3 NO				1	ONLINE LOGGING	NO	LOCAL	SYSTEM	NO AUTO	DISABLED	YES YES	DISABLE				
SYSAUX	PERMANENT	8.192	96.254	<b>74.096</b>	<b>22.158</b>	77,0 NO				3	ONLINE LOGGING	YES	LOCAL	SYSTEM	NO AUTO	DISABLED	NO NO	DISABLE	v\$SYSAUX_C			
TEMP_MOVEIT TEMPORARY	TEMPORARY	8.192	184.320	180.759	3.561	98,1 YES	524.288	34,5		1	ONLINE NOLOGGING	NO	LOCAL	UNIFORM	NO MANUAL	DISABLED	YES NO	DISABLE				
TEMP	TEMPORARY	8.192	720.601	720.487	114	100,0 NO				22	ONLINE NOLOGGING	NO	LOCAL	UNIFORM	NO MANUAL	DISABLED	NO NO	DISABLE				
LOG_DATASPO	PERMANENT	8.192	102.400	<b>5.153</b>	<b>97.247</b>	5,0 NO																
UNDOTBS1	UNDO	8.192	229.376	175.435	<b>53.941</b>	76,5 NO				1	ONLINE LOGGING	NO	LOCAL	SYSTEM	NO MANUAL	DISABLED	YES NO	DISABLE	Retention=NC			
DATA01	PERMANENT	8.192	2.007.020	<b>1.507.530</b>	<b>500.000</b>	99,2 NO				4	ONLINE LOGGING	NO	LOCAL	SYSTEM	NO AUTO	DISABLED	YES NO	DISABLE				

Indicates whether the In-Memory Column Store (IM column store) is by default enabled (ENABLED) or disabled (DISABLED) for tables in this tablespace

- Storage-Nutzung nach Typen, Schemata, Tablespaces
- Drilldown bis zu Objekten und Partitionen
- Kalkulation des durch Reorganisation freizugebenden Platzes
- Darstellung des real verfügbaren Storage unter Berücksichtigung der Fragmentierung

# Segment-Statistiken

Menü „Analysen / Statistiken“ / „Segment Statistics“



Kennwerte je Table/Index  
für Zeitraum

Sortierbar nach einzelnen  
Kennwerten

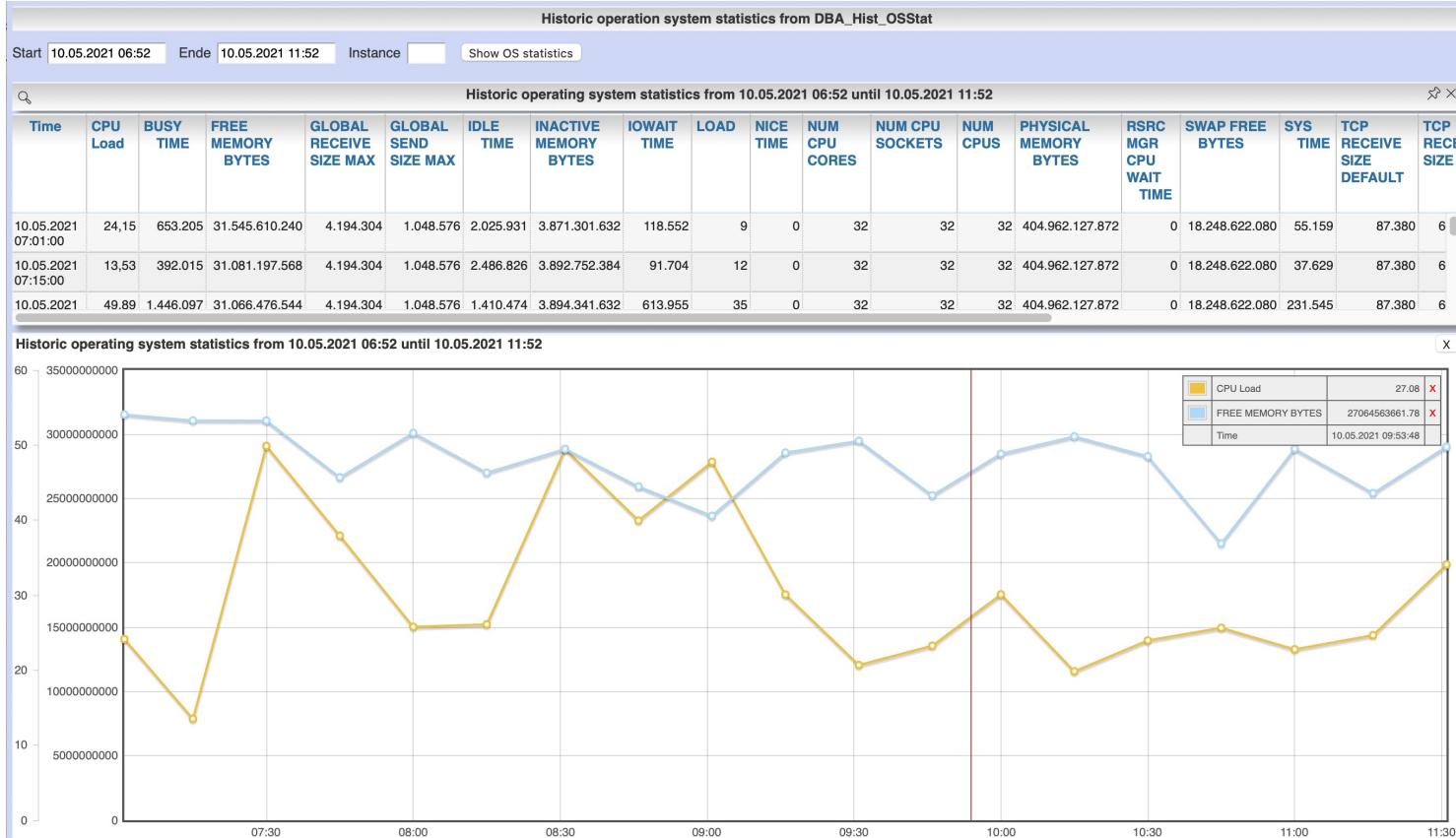
Zeitliche Entwicklung  
innerhalb des Zeitraums

# Agenda

- Vorstellung des verwendeten Tools „Panorama“
- Top/Down-Analyse der Session-Aktivitäten
- Ausgeführte SQL-Statements mit Detail-Analyse
- Segment-Statistiken: Aufzeichnung von Kennwerten für Objekte (Tabellen, Indizes ...)
- Kapazität und Auslastung von CPU, Memory und I/O-System, Konfiguration der Instance
- Rasterfahndung: Systematischer Scan des Systems auf Performance-Antipattern

# OS-Statistiken zu CPU und Memory

Menü: „Analysen/Statistiken“ / „OS Statistiken“



- CPU/Core-Anzahl
- CPU-Last der Maschine total
- Physischer Memory gesamt
- Physischer Memory noch verfügbar
- Swap verfügbar
- Historie in Auflösung des AWR-Zyklus

# I/O-Statistiken

Menü: „I/O-Analyse“ / „I/O-Historie nach Files“

I/O History from DBA\_Hist\_Filestatxs (Data and Temp files, without Redo-Logs)

Start 10.05.2021 06:52 Ende 10.05.2021 12:00 Instance Gruppierung FileType Show I/O history

Verdichtung nach FileType. Filter: Start time End time  
10.05.2021 06:52 Refresh 10.05.2021 12:00 Refresh

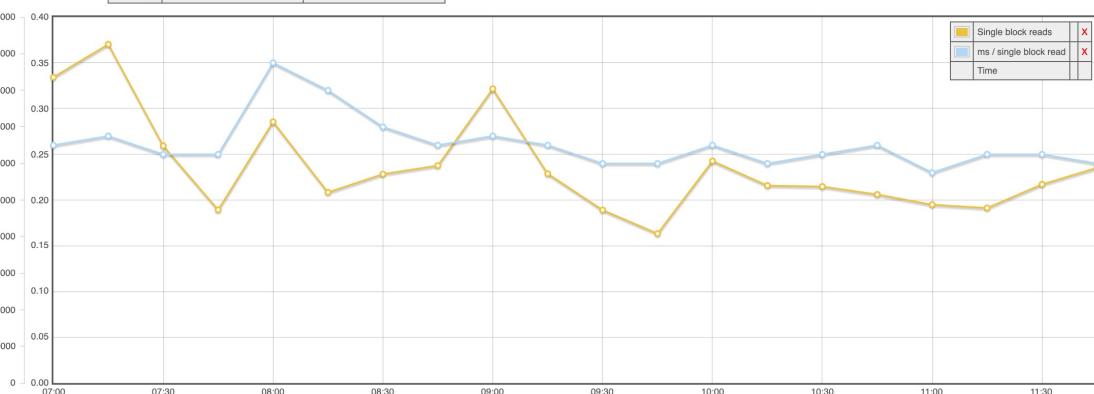
FileType	Erstes Auftreten	Letztes Auftreten	Dauer	DB	Inst.	Tablespace	Filename	Samples	Phys. reads	ms / phys. read	Blocks / phys. read	Phys. writes	ms / phys. write	Blocks / phys. write	Single block reads	ms / single block read	Blocks / multi block read	Total I/O per second	Read I/O per second	Write I/O per second
DATA	10.05.2021 07:00:01	10.05.2021 12:00:12	18.011	COR01PX	< 4 >	< 20 >	< 24 >	20	-3.696.468.899	-0,05	-0,21	56.528.705	1.300,52	1,85	591.784.930	0,26	0,0	-202.095	-205.234	3.139
TEMP	10.05.2021 07:00:01	10.05.2021 12:00:12	18.011	COR01PX	< 4 >	< 2 >	< 23 >	20	5.777.391	9,04	15,55	12.451.900	43.784,75	15,55	1.372.537	2,85	20,1	1.012	321	691

Avg. total I/O-operations per second within sample time

Sample-Records. Filter: FileType Start time End time  
DATA 10.05.2021 06:52 Refresh 10.05.2021 12:00 Refresh

Sample-Time	Phys. reads	ms / phys. read	Blocks / phys. read	Phys. writes	ms / phys. write	Blocks / phys. write	Single block reads	ms / single block read	Blocks / multi block read	Total I/O per second	Read I/O per second	Write I/O per second
10.05.2021 07:00:00	42.229.513	0,27	1,43	3.905.480	162,34	2,68	41.783.676	0,26	41,3	50.642	46.355	4.287
10.05.2021 07:15:00	46.722.063	0,28	1,32	6.188.553	179,33	2,36	46.289.290	0,27	35,6	57.449	50.730	6.720
10.05.2021 07:30:00	32.909.889	0,27	1,51	6.187.230	902,84	1,97	32.407.982	0,25	34,7	42.870	36.085	6.784

Sample-Records. Filter: FileType Start time End time  
DATA 10.05.2021 06:52 Refresh 10.05.2021 12:00 Refresh



- Drei Quellen für I/O-Historie aus AWR
- Aufzeichnung verschiedener Kennwerte
- unterschiedliche Gruppierung nach Filletypen bis zu Filenamen

- Die dritte Quelle DBA\_Hist\_Filestatxs wird durch AWR nicht mehr gefüllt ab 18c

# Memory-Konfiguration

Menü: „SGA/PGA-Details“ / „SGA-Memory“ / „SGA-Komponenten“

Anzeige der Memory-Verbraucher der SGA						
Instance	1	Anzeige der Memory-Komponenten				
I	Pool	Bytes	MBytes	GBytes	Parameter	Resize ops.
1	buffer cache	63.619.203.072	60.672,00	59,25	db_block_buffers = 0, db_cache_size = 42.949.672.960	356
1	shared pool	6.710.888.512	6.400,00	6,25	shared_pool_size = 5.368.709.120	27
1	numa pool	5.637.144.576	5.376,00	5,25		
1	large pool	268.435.456	256,00	0,25	large_pool_size = 0	315

Summary by pools from gv\$SGAStat for instance = 1						
I	Pool	Name	Bytes	MBytes	GBytes	Corresponding initialization parameter of instance
1		buffer cache	63.619.203.072	60.672,00	59,25	
1		numa pool	2.047.715.088	1.952,85	1,91	
1		shared pool	1.406.130.688	1.340,99	1,31	
1		SQLA	1.390.537.136	1.326,12	1,30	

Summary from gv\$DB_Object_Cache (objects that are cached in the library cache) for instance = 1										
I	Type	Namespace	DB-link	Kept	Sharable Memory (MB)	Count	Count distinct	Loads	Locks	Pin
1	CURSOR	SQL AREA		NO	1.898	132.223	41.558	2.762.349	157.583	
1	CURSOR STATS	SQL AREA STATS		YES	153	39.454	30.460	39.455	0	
1	MULTI-VERSIONED	MULTI-VERSION OBJECT		NO	43	887	57	1.005	0	
1	CURSOR STATS	SQL AREA STATS		NO	41	10.449	9.622	10.449	0	
1	PACKAGE	TABLE/PROCEDURE		NO	26	2.059	2.059	49.010	16.431	

Diese Ansicht zeigt:

- Dynamische Aufteilung des physischen Memory auf Komponenten
- Steuerung der Verteilung durch manuelle Vorgaben von Mindestgrößen

Optimierungsziel ist i.d.R. Nutzung des SGA-Memory für DB-Cache oder inMemory-Store

# Verwendung des DB-Cache nach Objekten

Menü: „SGA/PGA Details“ / „DB-Cache“ / „DB-Cache-Nutzung aktuell“

DB-Cache-Nutzung

Instance 1 Part.  Cache-Inhalt anzeigen

DB-Cache-Nutzung: Instance=1, 03.11.2018 20:07:00

Summen			
Status	Size (MB)	Blocks	%
read	0,516	66	0,00
free	0,367	47	0,00
scur	43.212,547	5.531.206	79,29
pi	108,797	13.926	0,20

Details

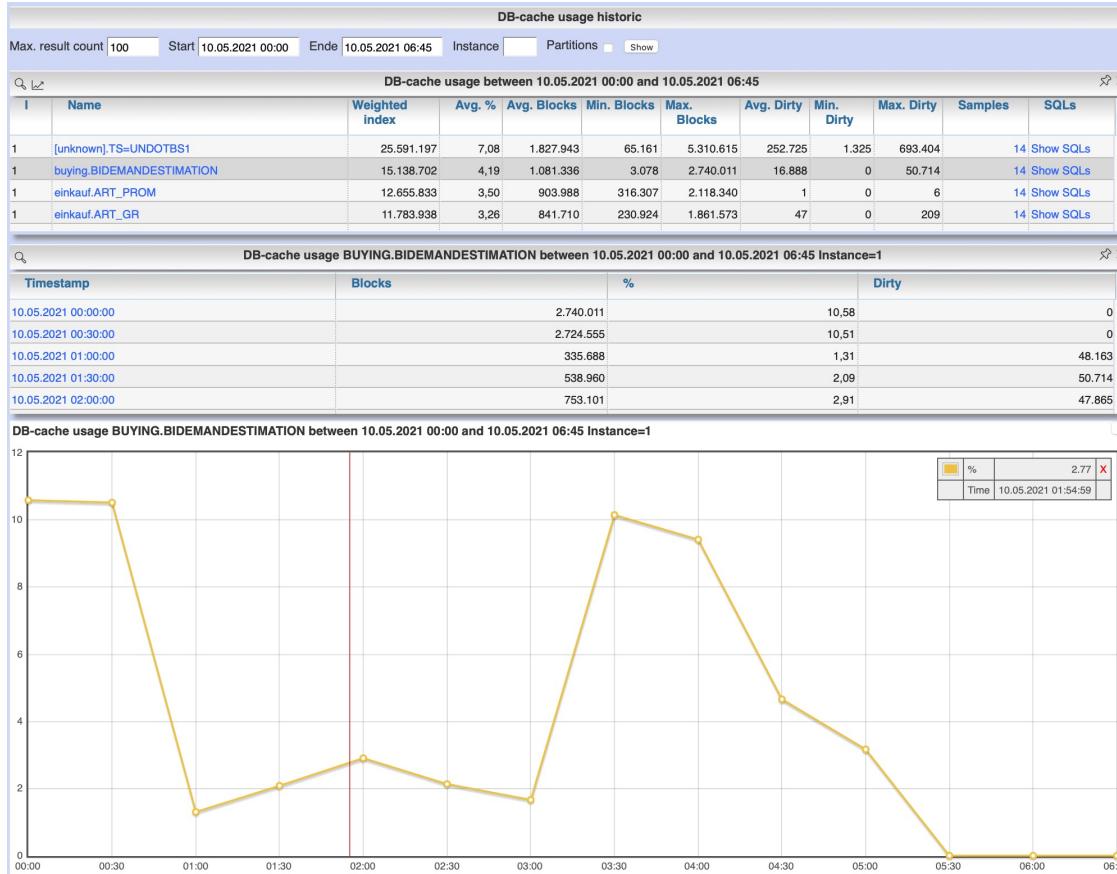
Name	Type	Tablename	SQLs	Size (MB)	Blocks	%	Dirty	cr	pi
CUST . PKEY_CUSTSCRES	INDEX PARTITION	CUST.CUSTSCORINGRESULT	240	3.665.031	469.124	6,66	1	151	
ITM . PKEY_AVAILABIL	INDEX	ITM.AVAILABILITY	15	3.221.188	412.312	5,86	0	0	
BONUS . PK_CUSTMASTBO	INDEX PARTITION	BONUS.CUSTMASTERBONUS	976	2.038.711	260.955	3,71	0	8	
CAPS . EMLOGENTRY	TABLE PARTITION		1.575	1.350.273	172.835	2,46	501	0	
[UNKNOWN] . TS-UNDOTBS1			0	1.344.664	172.117	2,44	33.585	206	
CAPS . EMREPLICATIONLOG	TABLE PARTITION		72	1.333.273	170.659	2,42	228	764	
AUFTRAG . AU_FORDERUNG_OP	TABLE		43	1.248.180	159.767	2,27	351	6.667	
CAPS . PK_PTLDLVRTR	INDEX	CAPS.PARTIALDELIVPARTREL	19	1.098.258	140.577	2,00	2	0	
ITM . EMREPLICATIONLOG	TABLE PARTITION		28	1.074.656	137.556	1,95	64	309	
LOLS . A_DELIVERYSTOPHISTORY	TABLE		2	987.859	126.446	1,80	0	0	
AUFTRAG . IX_LZLOOKUP_GROUP	INDEX	AUFTRAG.AU_LZ_LOOKUP	1	949.398	121.523	1,73	0	0	
JOURNAL .	INDEX	JOURNAL.OFMESSAGE	860	874.234	111.902	1,59	3.356	3.876	
CUST . CUSTTOPIC	TABLE		18	762.148	97.555	1,39	23	254	

## Nutzung des DB-Cache durch Tabellen und Indizes.

- Sicherstellen dass DB-Cache auch für fachlich relevante Objekte genutzt wird
- Erkennen von suboptimalen SQLs, die Objekte mit geringer fachlicher Relevanz an die Spitze der LRU-Liste bringen

# Historie des DB-Cache nach Objekten

Menü: „SGA/PGA Details“ / „DB-Cache“ / „DB-Cache-Nutzung historisch“

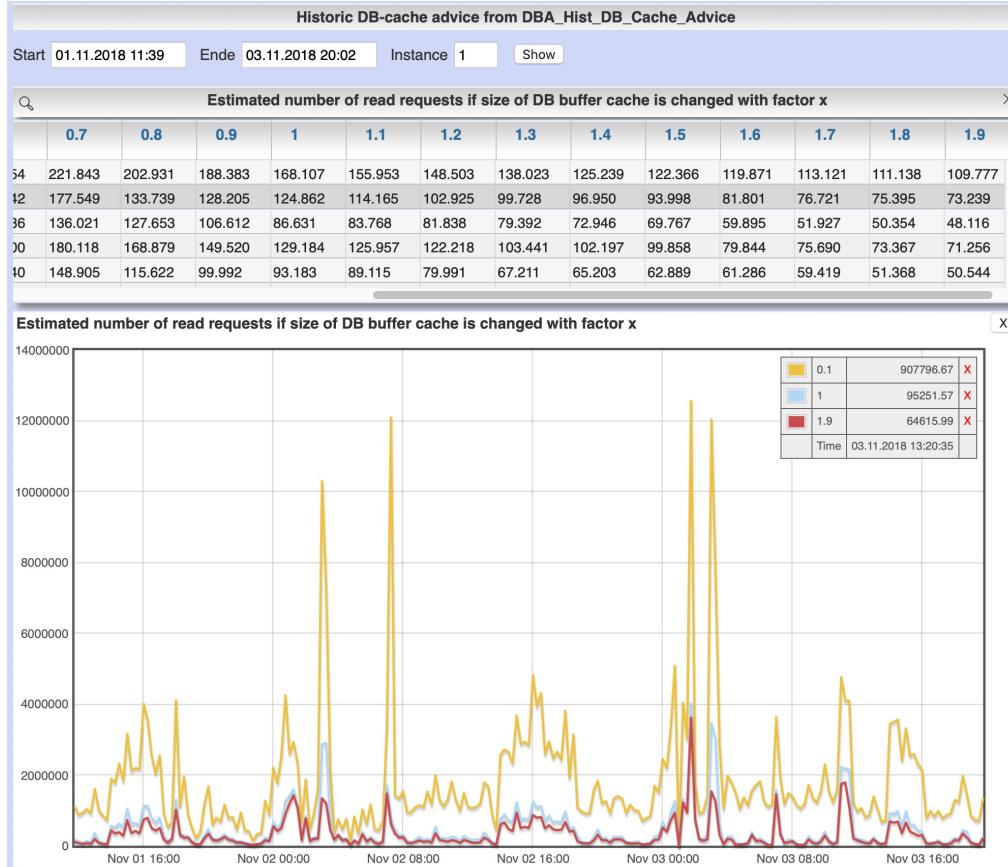


Nutzung des DB-Cache durch Tabellen und Indizes in einem Zeitraum der Vergangenheit.

- Datenquelle ist der Panorama-Sampler, nicht AWR
- Zeitlicher Verlauf der DB-Cache-Nutzung je Objekt

# Prognose bei Veränderung der DB-Cache-Size

Menü: „SGA/PGA Details“ / „DB-Cache“ / „DB-Cache-Advice historisch“



Prognose der Veränderung der Anzahl I/O-Read-Requests bei Vergrößerung/Verkleinerung des DB-Caches zu bestimmten Zeiten.

1 = aktuelle Cache-Size

0.1 = Reduktion Cache auf 10%

1.8 = Erhöhung Cache um 80%

# Konfiguration der Redo-Logs: Ist-Zustand

Menü: „DBA Allgemein“ / „Redologs“ / „Aktuell“

Redo-Logfiles from GV\$LOG at 19.10.2018 08:13:05							
Instance	Group	Size (MB)	Status	Start timestamp	Members	Archived	Log switch interval
1	1	200,00	CURRENT	18.10.2018 10:36:38		1 NO	
1	3	200,00	INACTIVE	15.10.2018 07:47:04		1 NO	269.374
1	2	200,00	INACTIVE	14.10.2018 23:15:10		1 NO	30.714

Dieses Bild zeigt die Standard-Konfiguration nach Installation einer Oracle-DB.

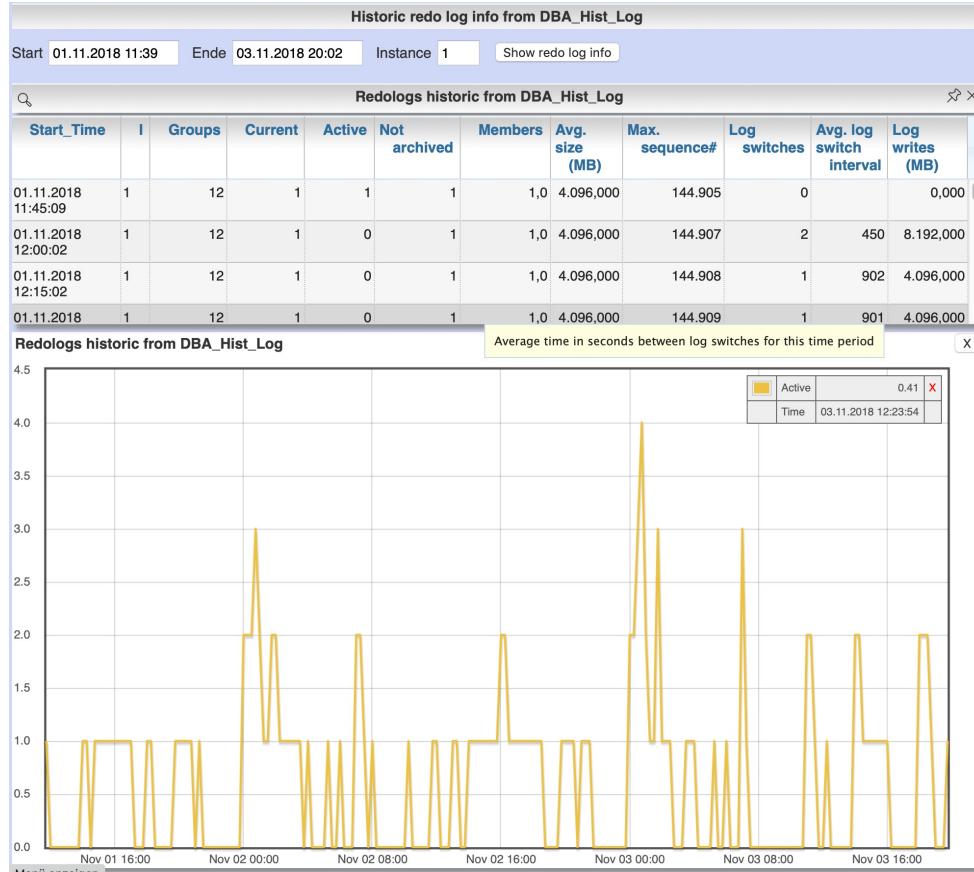
Findet sich leider vielfach auch exakt so in Produktionssystemen.

## Problempunkte:

- Mit nur 3 Redo-Log-Gruppen besteht bei entspr. DML-Aufkommen latente Gefahr von „cannot switch redo log file“ nachdem aktuelles Logfile durch den Logwriter-Prozess vollgeschrieben wurde.  
Bis zur Wiederverfügbarkeit eines freien Redo-Logfiles werden alle Commit-Operationen angehalten. (GAU für OLTP-Systeme)
- 200 MB Default-Size kann viel zu klein sein  
(Optimierungsziel > 10 Sekunden zwischen Log-Switches)

# Auslastung der Redo-Logs in Historie

Menü: „DBA Allgemein“ / „Redologs“ / „Historisch“



## Bewertung der Auslastung der Redo-Logs im AWR-Takt.

- Current + Active sollte niemals die Anzahl der verfügbaren Redo-Log-Gruppen erreichen
- Gegencheck auch über alert.log mit Suche nach „cannot allocate new log“ um evtl. Probleme innerhalb der AWR-Zyklen zu erkennen
- Dieser [Blog-Post](#) beschreibt das Problem im Detail

# Agenda

- Vorstellung des verwendeten Tools „Panorama“
- Top/Down-Analyse der Session-Aktivitäten
- Ausgeführte SQL-Statements mit Detail-Analyse
- Segment-Statistiken: Aufzeichnung von Kennwerten für Objekte (Tabellen, Indizes ...)
- Kapazität und Auslastung von CPU, Memory und I/O-System, Konfiguration der Instance
- Rasterfahndung: Systematischer Scan des Systems auf Performance-Antipattern

# Rasterfahndung nach Performance-Antipattern

Menü „Spez.Erweiterungen“ / „Rasterfahndung“

**OSP**  
Otto Group Solution Provider

The screenshot shows a web-based application for database performance analysis. On the left, a sidebar titled "Auswahl des Rasterfahndungs-SQL" lists several categories with icons:

- 1. Potential in DB-Strukturen
- 2. Ermittlung von SQL-Statements mit suboptimalen Ausführungsplänen
- 3. Tuning bzw. Entlastung von SGA/PGA-Strukturen
- 4. Redundante Cursoren / Nutzung Bindevariablen
  - 1. Fehlende Nutzung von Bindevariablen: Ermittlung über identische Plan-Hash-Values aus Active Session History (selected)
  - 2. Fehlende Nutzung von Bindevariablen: Ermittlung über identische Plan-Hash-Values aus SGA
  - 3. Fehlende Nutzung von Bindevariablen: Ermittlung über identische Teile des SQL-Textes
  - 4. Anzahl unterschiedlicher SQLs je Zeit in Zeitleiste
  - 5. Mehrfach offene Cursoren: Überblick über SQL

At the top right of the sidebar are search filters: "Filter:", "Include description", and "Search".

The main panel on the right is titled "Fehlende Nutzung von Bindevariablen: Ermittlung über identische Plan-Hash-Values aus Active Session History". It contains the following information:

Nutzung von Literalen statt Bindevariablen mit hoher Anzahl unterschiedlicher Literale führt zu hohen Parse-Zahlen und Flutung der SQL-Area in der SGA.  
Auch wenn die Folgen durch Setzen von cursor\_sharing != EXACT reduziert werden können, bleibt eine hohe Speicheranforderung in SGA für das Matchen der SQLs mit den korrespondierenden SQLs mit ersetzen Bindevariablen.  
Die eindeutige Empfehlung ist: Bindevariablen benutzen!  
Die Abfrage sucht über identische Ausführungspläne per Plan-Hash-Value aus Active Session History.

Betrachtung der Historie rückwärts in Tagen:

Minimale Anzahl unterschiedlicher SQL-IDs:

## Performance-Optimierung: reaktiv Ereignis-getrieben oder vorbeugend?

### Motivation

- Erkennung aller weiteren Vorkommen einer einmal analysierten Problemstellung im System
- Möglichst einfache Umsetzbarkeit der Lösungsvorschläge ohne Eingriff in die Architektur und Design
- Fixen erkannter leicht lösbarer Problemstellungen systemweit statt step by step nach Eskalation
- Komfortable Einbettung in weiteren Analyse-Workflow

Erweiterbar mit eigenen Abfragen, Details dazu in [diesem Blog-Post](#)

# Beispiel: Ungenutzte und unnötige Indizes

Menü „Spez.Erweiterungen“ / „Rasterfahndung“: Punkt 1.2.4



## Erkennung der Nutzung von Indizes durch SQL-Statements

- ALTER INDEX xxx MONITORING USAGE aktiviert die Protokollierung der Nutzung des Index durch SQL
- Damit wird eine eindeutige Aussage möglich: Index wurde seit x Tagen niemals verwendet in 1st-level SQL
- Einschränkung: Rekursive Zugriffe auf Index beim Check eines Foreign Keys werden hier nicht protokolliert

Für stetige Überwachung des Nutzungszustandes von Indizes empfiehlt sich zyklische Ausführung eines Scriptes:

- Reaktivieren des Index-Monitoring nach x Tagen wenn eine Nutzung des Index festgestellt wurde
- Damit Erkennung ab wann ein bislang aktiver Index nicht mehr genutzt wird
- Ausführung von ALTER INDEX xxx MONITORING USAGE führt zu Invalidisierung aller Cursors, die diesen Index nutzen und damit zu Lastspitzen beim erneuten Parsen. Dies kann in hoch frequentierten OLTP-Systemen kritisch sein.
- Deshalb Reaktivierung per ALTER INDEX xxx MONITORING USAGE nur für Indizes, die aktuell nicht in Ausführungsplänen der SGA enthalten sind
- [Blog-Post](#) enthält u.a. das PL/SQL-Script für zyklische Reaktivierung des Monitoring-Zustandes sowie SQL-Abfragen

**Fazit: Automatisierte Erkennung von entbehrlichen Indizes ist machbar.**

**Non-Unique Indizes ohne Foreign Key und ohne Nutzung seit x Tagen können gesichert entfernt werden.**

# Letztes Wort

Panorama greift nur lesend auf die Datenbank zu und benötigt keine eigenen PL/SQL-Objekte.

Sie können die Funktionen also ohne Risiko testen und verstehen. Probieren Sie es gern aus.

Tip:

Gestartet mit Environment-Variable „PANORAMA\_LOG\_LEVEL= debug“ werden alle durch Panorama ausgeführten SQL-Statements im Original im Log-Output des Panorama-Servers protokolliert.

Beschreibung inkl. Download-Link:

<https://rammpeter.github.io/panorama.html>

Docker-Image

<https://hub.docker.com/r/rammpeter/panorama>

Blog zum Thema:

<https://rammpeter.blogspot.com>

# Vielen Dank für Ihr Interesse

Otto Group Solution Provider (OSP) GmbH  
Freiberger Str. 35 | 01067 Dresden  
Telefon +49 (0)351 49723 0  
[osp.de](http://osp.de)