

# Atos

Trusted partner for your **Digital Journey**

# Lieblingskind oder Totgeburt?

---

Automatic Big Table Cache revisited

# Zur Person

---

- ▶ Dipl.Ing.(FH)
- ▶ 20(+) Jahre Erfahrung mit Oracle
- ▶ Seit Oracle7.3
- ▶ Database Architect
- ▶ Seit 17 Jahren bei science + computing AG
- ▶ Seit 4 Jahren bei Atos
- ▶ Themen
  - Administration
  - Performance
  - Virtualisierung
  - Oracle APEX
  - SQL & PL/SQL



1

Liebling, es ist ein Cache!

# Automatic Big Table Cache

---

- ▶ Eingeführt mit Oracle 12.1.0.2 (Juli 2014)
- ▶ No-Cost Option
- ▶ Parallele Queries im RAC, Serial + Parallel bei Single Instance
- ▶ Teil des Buffer Cache
- ▶ Objektbasiertes Management

# Warum ABTC

## eine wahre Geschichte

---

- ▶ Migration einer Kundenumgebung abgeschlossen
- ▶ 2 Wochen später: Performance schlecht
- ▶ Analyse: miserable IO Charakteristik
- ▶ Auf der HW Seite: Storage Vendor kann nichts finden
- ▶ Alternative: Gezielt wichtigste Tabelle cachen

Aber wie?

2

Basics

# Caches in Oracle 12c

## 12.1

NAME	TYPE	VALUE
client_result_cache_lag	big integer	3000
client_result_cache_size	big integer	0
db_big_table_cache_percent_target	string	0
db_cache_advice	string	ON
db_cache_size	big integer	0
db_flash_cache_file	string	
db_flash_cache_size	big integer	0
db_keep_cache_size	big integer	0
db_recycle_cache_size	big integer	0
db_16k_cache_size	big integer	0
db_2k_cache_size	big integer	0
db_32k_cache_size	big integer	0
db_4k_cache_size	big integer	0
db_8k_cache_size	big integer	0
object_cache_max_size_percent	integer	10
object_cache_optimal_size	integer	102400
result_cache_max_result	integer	5
result_cache_max_size	big integer	4608K
result_cache_mode	string	MANUAL
result_cache_remote_expiration	integer	0
session_cached_cursors	integer	50

- ▶ 21 Parameter, um die Caches konfigurieren
- ▶ 12 Parameter allein, um die Größe des jeweiligen Caches bestimmen
- ▶ Plus 9 Pool-Size Parameter (z.T. deprecated)
- ▶ Plus cell\_flash\_cache
- ▶ Full Database Caching

## 12.2

- ▶ Neu: data\_transfer\_cache\_size

Siehe

[http://docs.oracle.com/database/122/REFRN/DATA\\_TRANSFER\\_CACHE\\_SIZE.htm#REFRN-GUID-322093B7-1673-490D-8A1A-5F461D7897DD](http://docs.oracle.com/database/122/REFRN/DATA_TRANSFER_CACHE_SIZE.htm#REFRN-GUID-322093B7-1673-490D-8A1A-5F461D7897DD):

This initialization parameter needs to be set only for databases that do not use Automatic Memory Management and that use the RMAN RECOVER ... NONLOGGED BLOCK command.

## 18c/19c

- ▶ Keine neuen Parameter



# Buffercache

## Primer

---

- ▶ **Buffer caches**
  - default, keep, recycle
  - Verschiedene Blockgrößen
  - Flash cache (L2-Cache)
  - Big Table Cache



# Buffercache

## Primer

---

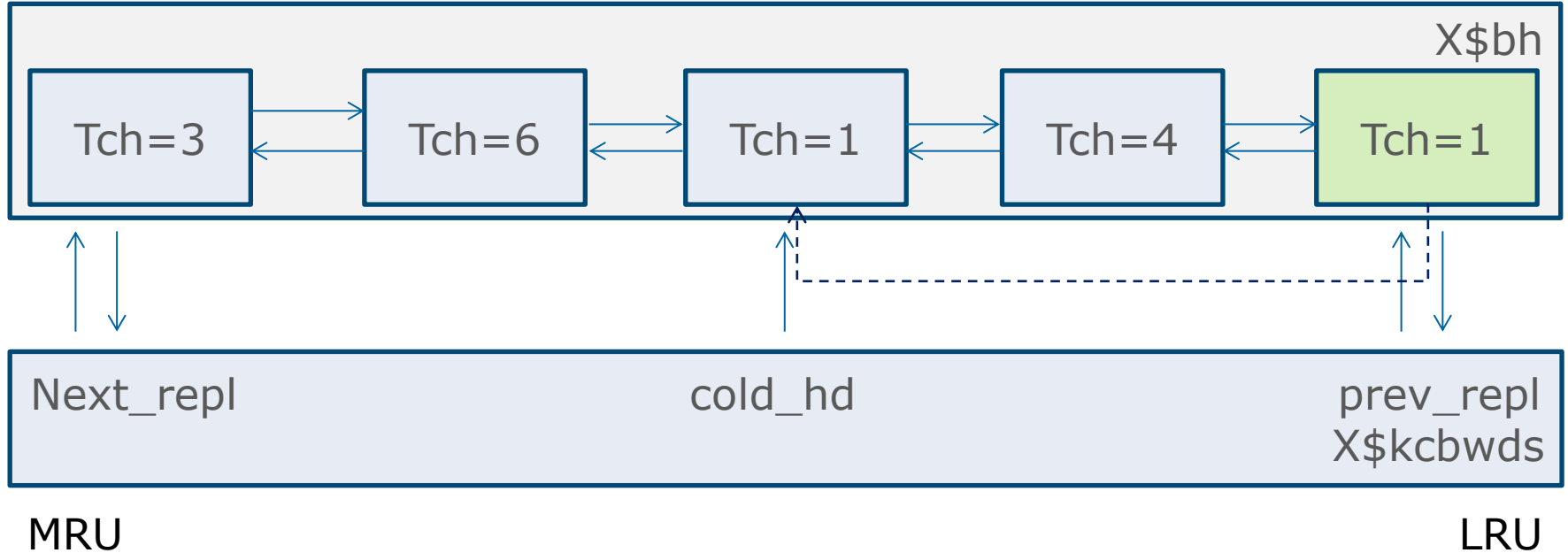
- ▶ Find free block
- ▶ Pin block in memory
- ▶ Load block into memory



# Buffercache

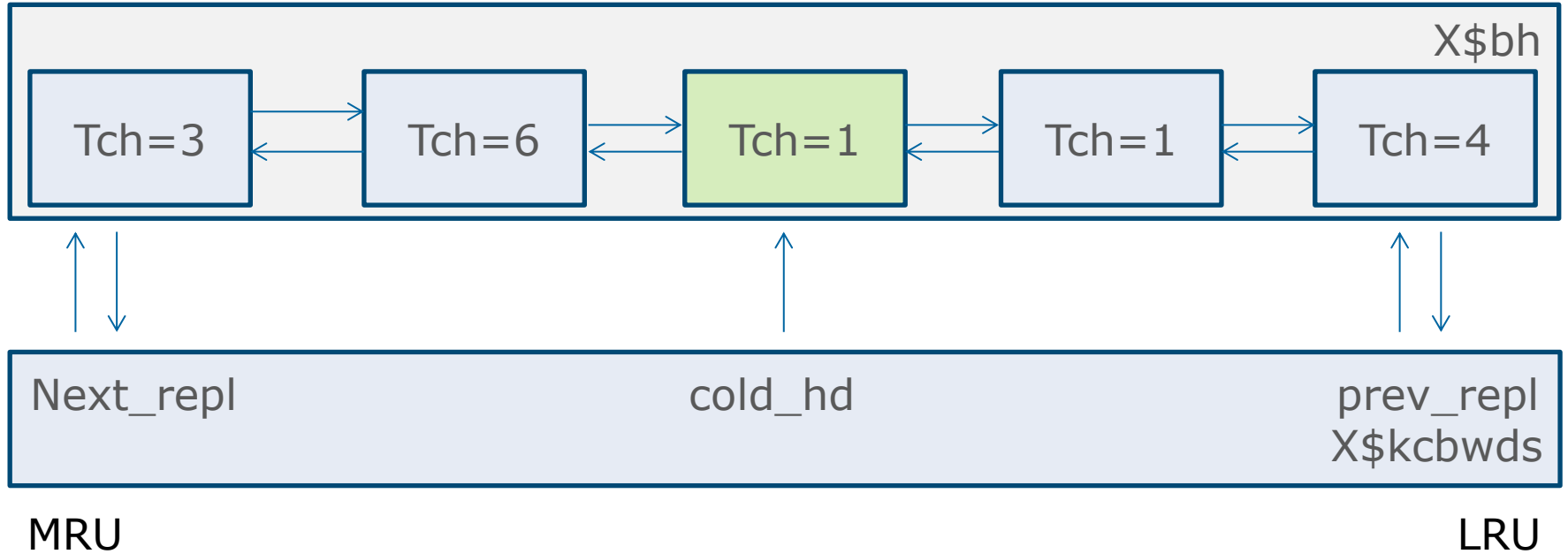
## LRU/TCH

- ▶ Least recently used / Touch Mechanismus



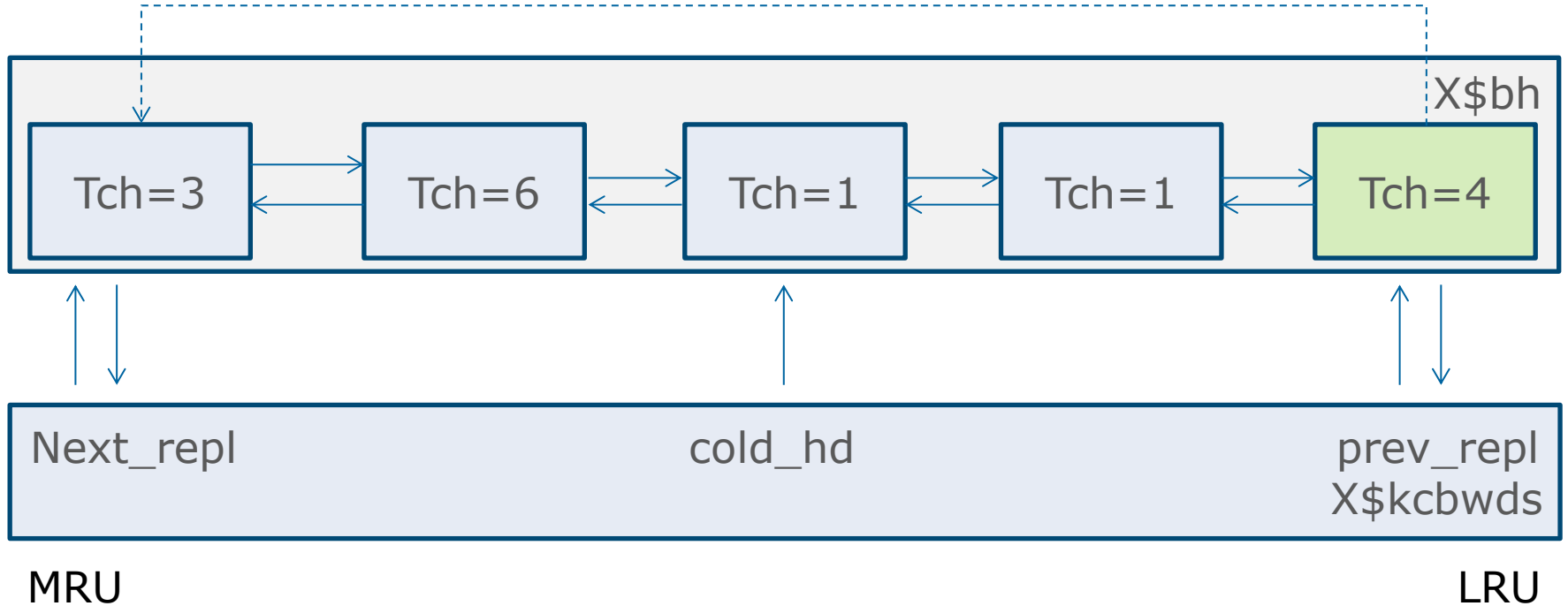
# Buffercache

## LRU/TCH



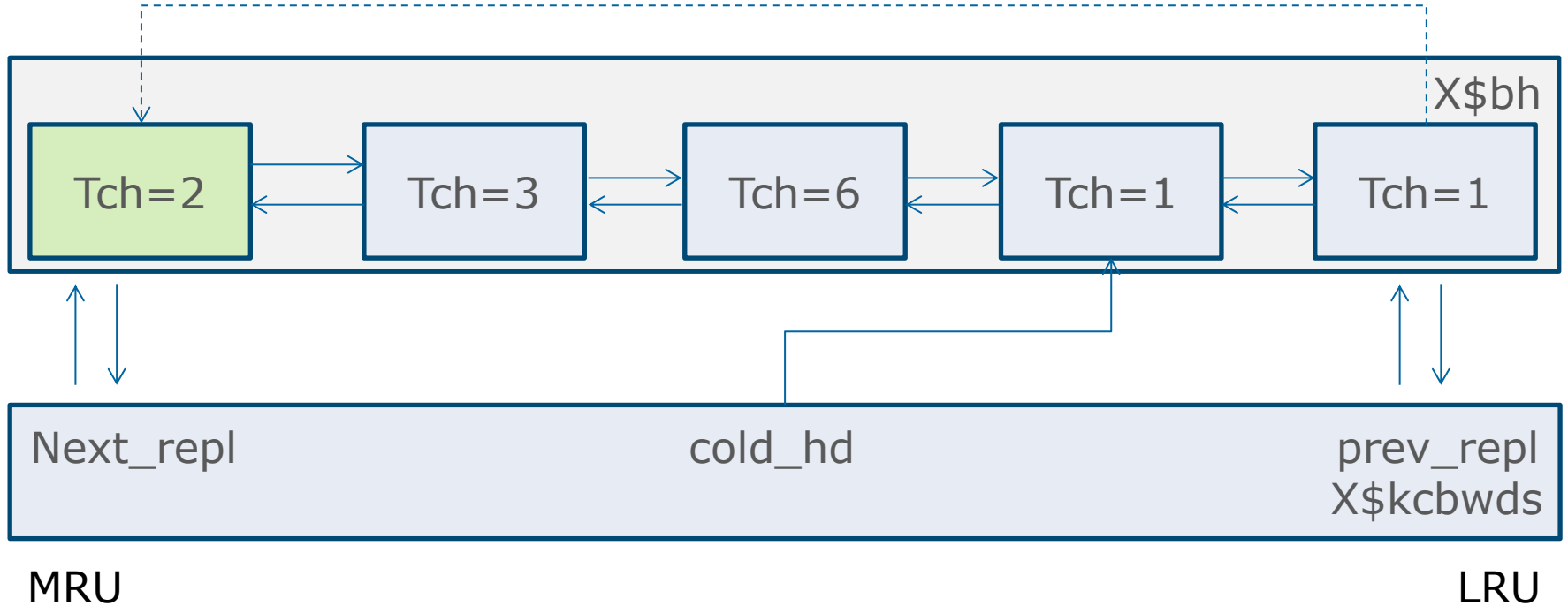
# Buffercache

## LRU/TCH



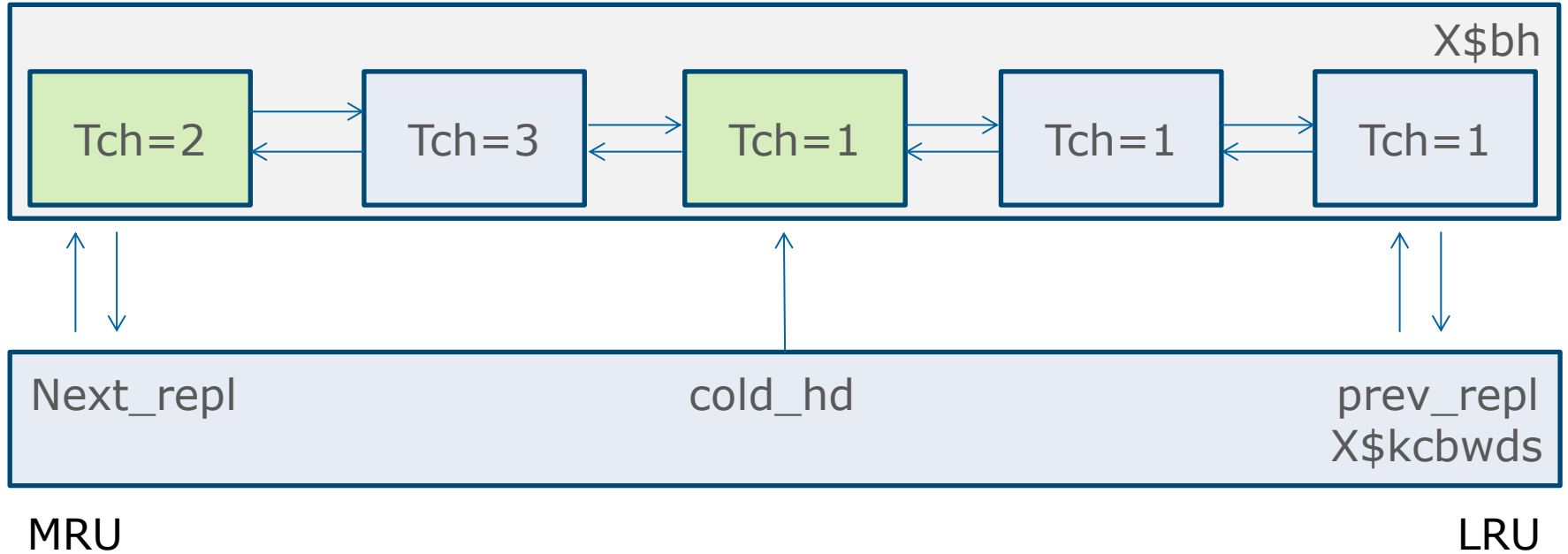
# Buffercache

## LRU/TCH



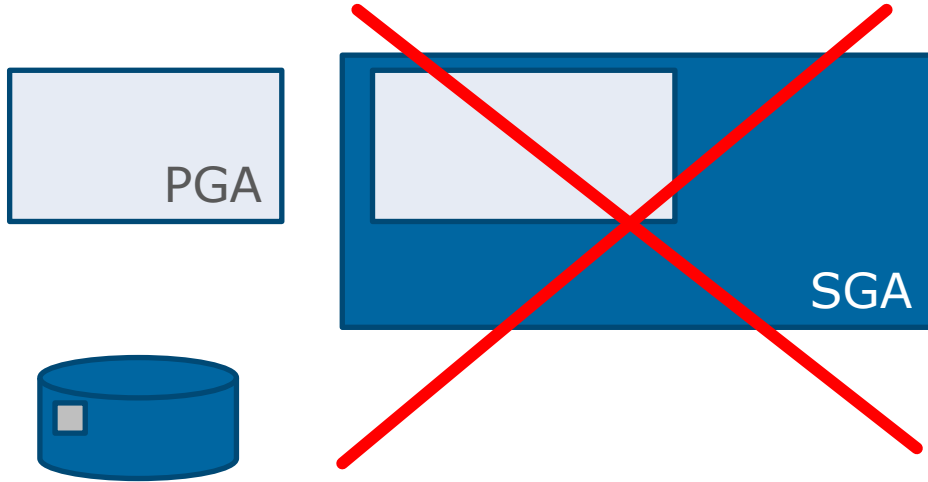
# Buffercache

## LRU/TCH



# Direct Read

- ▶ Umgehen des Buffercache bei sequential reads (TFS, FFS)
- ▶ Segment > „\_small\_table\_threshold“



<https://blogs.oracle.com/smartscale-deep-dive/when-bloggers-get-it-wrong-part-1>



# Direct Read

---

- ▶ IO
- ▶ Auch populäre Tabellen betroffen die in den Buffer Cache passen würden
- ▶ (Default) Buffer Cache vergrössern?
- ▶ Keep Cache / Recycling Cache?

# Neu in Oracle 12c

---

## FULL Database Caching Automatic Big Table Cache!

3

## Automatic Big Table Caching

---

# Automatic Big Table Cache

---

- ▶ Optionaler Teil des Default(?) Pool
- ▶ ein einziger Parameter
- ▶ Prozentualer Anteil des Buffer Cache
- ▶ Caching Pro Segment
- ▶ über die Häufigkeit des Zugriffs bestimmt
- ▶ „Temperatur“basiert
- ▶ Parallel Query in RAC und Single Instance
- ▶ Serial Scan nur Single Instance

# ABTC im Einsatz

- ▶ V\$BT\_SCAN\_CACHE (Übersicht)
- ▶ V\$BT\_SCAN\_OBJ\_TEMPS (Details)

```
SYS@ORCL SQL> alter system set db_big_table_cache_percent_target=80;
```

System wurde geändert.

```
SYS@ORCL SQL> select * from v$bt_scan_cache;
```

BT_CACHE_ALLOC	BT_CACHE_TARGET	OBJECT_COUNT	MEMORY_BUF_ALLOC	MIN_CACHED_TEMP	CON_ID
,154348845	80	22	1040510	1000	0

# ABTC im Einsatz

## Demo

---

- ▶ Workload ausführen
- ▶ V\$BT\_SCAN\* Views analysieren
- ▶ Zusätzlich autotrace in SQL\*Plus



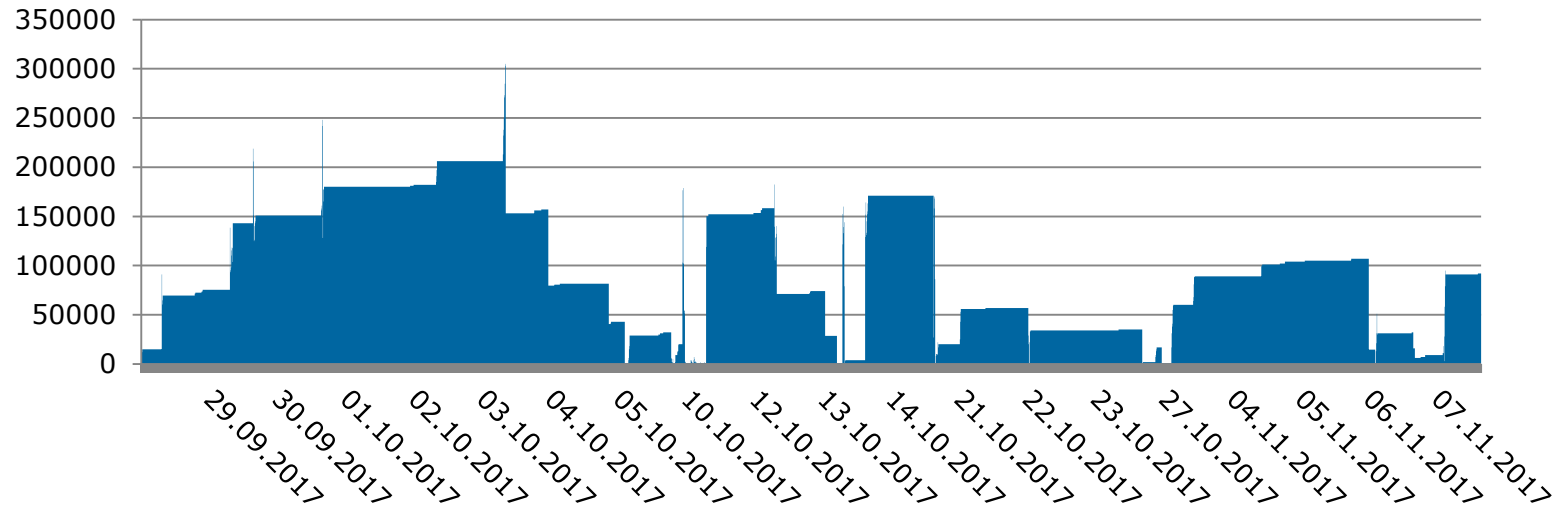
DEMO

# ABTC im Einsatz

## Langzeit

- ▶ V\$BT\_SCAN\_OBJ\_TEMPS archivieren
- ▶ Verlauf für eine Tabelle über mehrere Tage

### TEMPERATURE



# ABTC im Einsatz

## Probleme + Einschränkungen

---

- ▶ Keine Möglichkeit bestimmte Segmente zu pinnen
- ▶ Mechanismus der Temperatur reduziert nicht dokumentiert
  - Abhängig vom Workload
  - Zeitabhängig
- ▶ Index FFS nur bei parallel Query
- ▶ Kein flush Operator !



# ABTC im Einsatz

## Anomalien

---

- ▶ Truncate Table
  - Data\_object\_id ändert sich
  - Neue data\_object\_id nicht im Cache
  - Alte data\_object\_id weiter im Cache
- ▶ Move Table hat den gleichen Effekt
- ▶ Komplexe Poolkonfiguration
- ▶ Partitionierte Tabellen werden falsch abgebildet (v\$bt\_scan\_obj\_temps)



DEMO

# ABTC im Einsatz

## Bugs

---

- ▶ Bug 21514877:
  - BT\_CACHE\_TARGET Is Re-adjusted To 0 When A DataPump Job is Started (Doc ID 2060627.1)
  - Fixed mit Patch 2154877 / Upgrade 12.2 / PSU JUL2017 (1707)
- ▶ Bug 19200677:
  - Ora-00600[kcblrs\_2] wenn dynamic\_sampling aktiv ist
- ▶ Doc ID 2471008.1
  - **Wrong Result Sometimes Occur When Set DB\_BIG\_TABLE\_CACHE\_PERCENT\_TARGET Parameter**

# ABTC im Einsatz

## Anomalien

1500000 Zeilen aktualisiert.

PCOL	COUNT(*)
2	798
4	520242
5	479758
3	499202

OWNER	TBS	DATAOBJ#	OBJECT_TYPE	OBJECT_NAME	SIZE_IN_BKLS	TEMPERATURE	POLICY	CACHED_IN_MEM
HB	TBS16K	75474	TABLE	MEDIUM	14604	48000	MEM_ONLY	14604
	USERS	104445			11430	10000	MEM_ONLY	11430
	USERS	104441			12090	10000	MEM_ONLY	12090
	USERS	104451			12090	10000	DISK	0
	USERS	104446			11430	10000	MEM_PART	11217
	USERS	104442			12090	9000	DISK	12090
HB	USERS	104456	TABLE PARTITION	MTAB.P1	12090	9000	DISK	0
	USERS	104452			12090	9000	DISK	0
HB	USERS	104457	TABLE PARTITION	MTAB.P2	12090	8000	DISK	0
	USERS	104454			11273	8000	DISK	0
HB	USERS	104459	TABLE	SQLTEST	11430	8000	DISK	0
	USERS	104453			12090	6000	DISK	0
	USERS	104443			12090	6000	DISK	12090
HB	USERS	74608	TABLE	MEDIUM5M	42737	6000	DISK	9149
HB	USERS	104458	TABLE PARTITION	MTAB.P3	12090	5000	DISK	0
HB	USERS	74336	TABLE	SMALL30K	3125	2000	DISK	0

# ABTC im Einsatz

## Anomalien

- ▶ Non-Standard Blocksize
  - V\$BT\_SCAN\* Views nicht mehr korrekt
  - Änderung von db\_big\_table\_cache\_percent\_target keine Wirkung bis Neustart

SET_ID	POOL_ID	BLK_SIZE	CNUM_SET	CNUM_DWB	ALLOC
1	1	8192	1959	773	,39
2	1	8192	1959	758	,39
3	1	8192	1959	794	,41
4	1	8192	1959	776	,4
9	3	8192	23017	16112	,7
10	3	8192	23021	16115	,7
11	3	8192	23019	16114	,7
12	3	8192	23016	16112	,7
13	4	2048	8707	3460	,4
14	4	2048	8707	3465	,4
15	4	2048	8707	3481	,4
16	4	2048	8704	3542	,41
25	7	16384	3250	1451	,45
26	7	16384	3250	1339	,41
27	7	16384	3250	1455	,45
28	7	16384	3250	1439	,44

```
select
    set_id,
    Pool_id,
    blk_size,
    cnum_set,
    cnum_dwb,
    round(cnum_dwb/cnum_set,2) alloc
from x$kcwds
where cnum_dwb >0;
```

# ... und 19c?

---

- ▶ Leider keine Verbesserung erkennbar!
- ▶ Genau das gleiche Verhalten wie 12.1
  - Evt. kleine Unterschiede bei nicht-Defaultblocksize

# Fazit

---

- ▶ Wenig zu verlieren?
- ▶ bestimmte Workloads könnten profitieren
- ▶ Zuwenig Kontrolle in vielen Fällen
- ▶ Irritationen und Bugs!
- ▶ Nichts für Multiple-Blocksize DB
- ▶ Bei falschen Ergebnissen hört der Spass auf

4

... the dba's tale ends

# Der wahre Schluss

---

## ► 2 Wochen später...

Hallo Herr ██████, Hallo Herr ██████

Ich schreibe Ihnen um sie zu informieren, dass unsere IT gestern vermutlich das Problem gefunden hat. Erste Ergebnisse nach dessen Beseitigung sehen absolut vielversprechend aus.

Problem:

Es sieht so aus, als wäre „einfach nur“ ein LWL-Kabel beim Server defekt gewesen. Daher rührte eine schlechte Kommunikation von ESX, NetApp etc.

(Stand jetzt: Latenzzeiten der Festplatte von prod-01 bei ca. 1-10ms dauerhaft (vorher bis zu 15.000). Unser Performance Tool zeigt konstant zwischen 13 und 16 Sekunde (22 Sekunden waren Bestwerte vorher, die letzten Tage waren wir zwischen 100 und 600)

Letztlich ein schnelles und absolut unspektakuläres Ende eines anhaltenden Problems ☺

Ich denke dennoch, dass keine Optimierung die vorgenommen vergebens war. Ich danke Ihnen beide für die kontinuierliche Hilfe!



# Quellen

---

- ▶ <http://docs.oracle.com/database/122/TGDBA/designing-and-developing-for-performance.htm#TGDBA94123>
- ▶ <https://blogs.oracle.com/smartscan-deep-dive/when-bloggers-get-it-wrong-part-1>
- ▶ <https://blogs.oracle.com/smartscan-deep-dive/when-bloggers-get-it-wrong-part-2>
- ▶ <https://jonathanlewis.wordpress.com/2011/03/14/buffer-sqlplustates/>
- ▶ Jonathan Lewis: "Oracle Core: Essential Internals for DBAs and Developers"

# Quellen

---

- ▶ <https://www.red-gate.com/simple-talk/sql/oracle/oracle-database-12-1-0-2c-automatic-big-table-cache/>
- ▶ <https://blogs.oracle.com/oraclemagazine/big-table-caching>
- ▶ <https://apex.oracle.com/pls/apex/germancommunities/dbacommunity/tipp/3841/index.html>
- ▶ <https://www.youtube.com/watch?v=-dU69ni8pMQ>
- ▶ <https://mahmoudhatem.wordpress.com/2014/10/26/oracle-12c-automatic-big-table-caching/>

# Vielen Dank für die Aufmerksamkeit

---

Holger Bär      [holger.baer@atos.net](mailto:holger.baer@atos.net)

**Atos**