

Atos

Trusted partner for your **Digital Journey**

Ist das Kunst oder kann das weg?

Oracle Caches im Einsatz

Zur Person

- ▶ Dipl.Ing.(FH)
- ▶ 19 Jahre Erfahrung mit Oracle
- ▶ Seit Oracle7.3
- ▶ Database Architect
- ▶ Seit 15 Jahren bei science + computing AG
- ▶ Seit 2 Jahren bei Atos
- ▶ Themen
 - Administration
 - Performance
 - Virtualisierung
 - Oracle APEX
 - SQL & PL/SQL



1

... a dba's tale...

Warum Caching

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

Caching

- ▶ Reduzierung von IO
- ▶ Reduzierung von CPU

Cache frequently accessed, rarely changing data that is expensive to retrieve on a repeated basis. However, make this cache mechanism easy to use, and ensure that it is indeed cheaper than accessing the data in the original method. This is applicable to all modules where frequently used data values should be cached or stored locally, rather than be repeatedly retrieved from a remote or expensive data store.

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:

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.

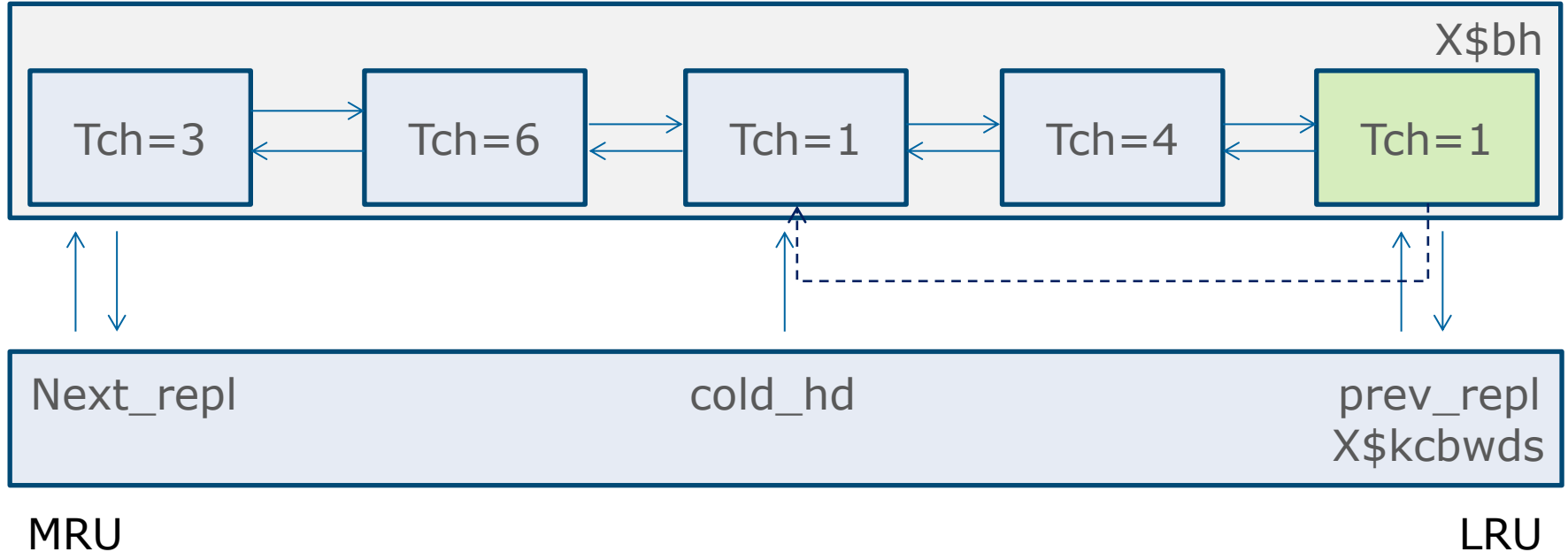
Ist das Kunst oder kann das weg?

Oracle **BUFFER** Caches im Einsatz

Buffercache

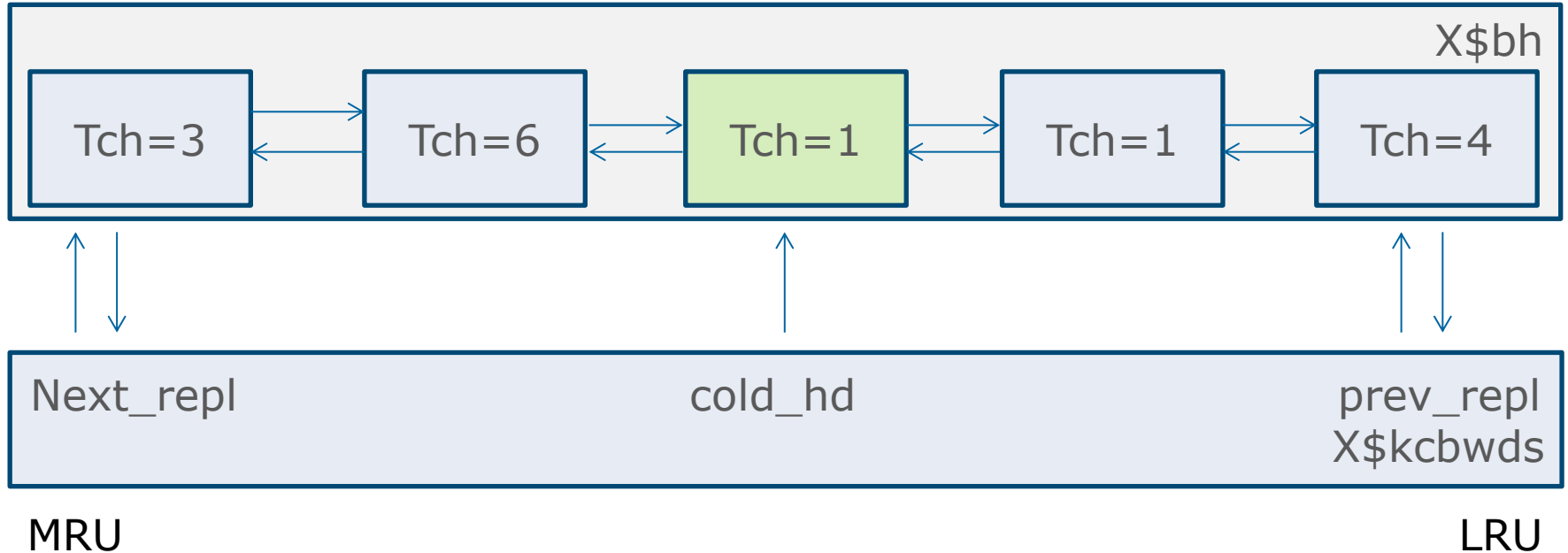
LRU/TCH

- ▶ Least recently used / Touch Mechanismus



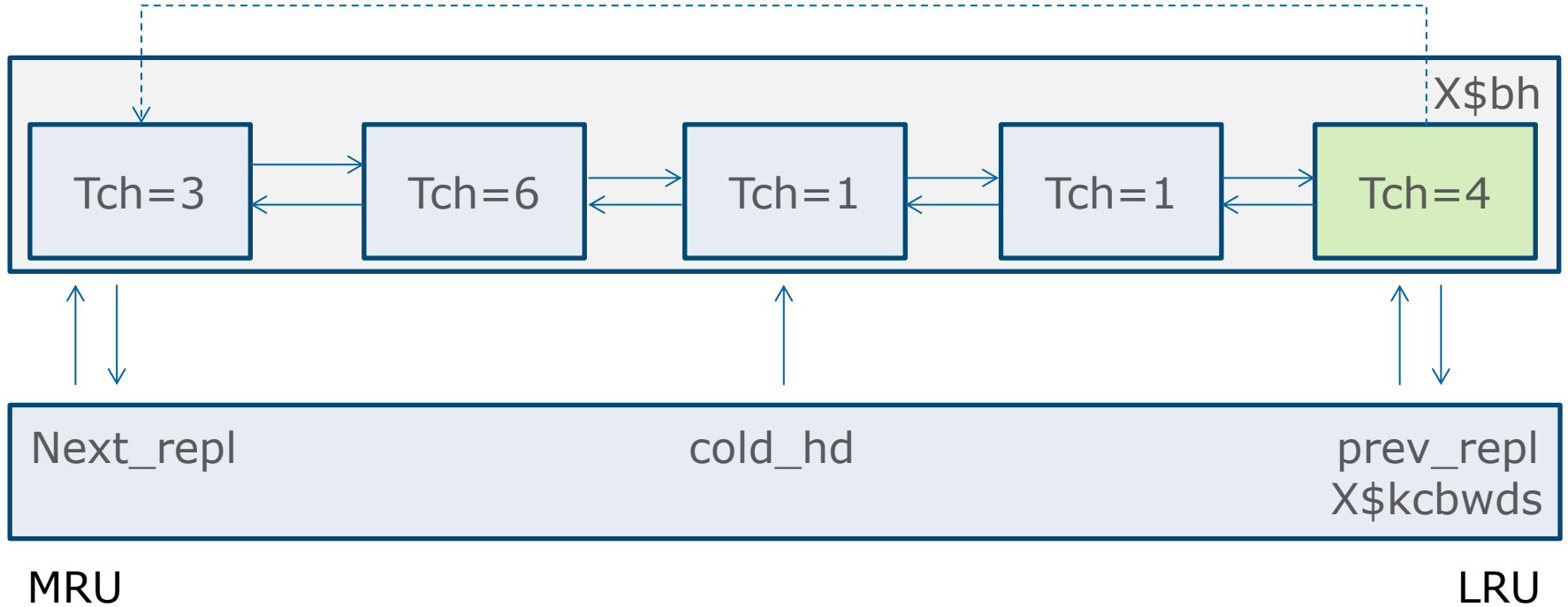
Buffercache

LRU/TCH



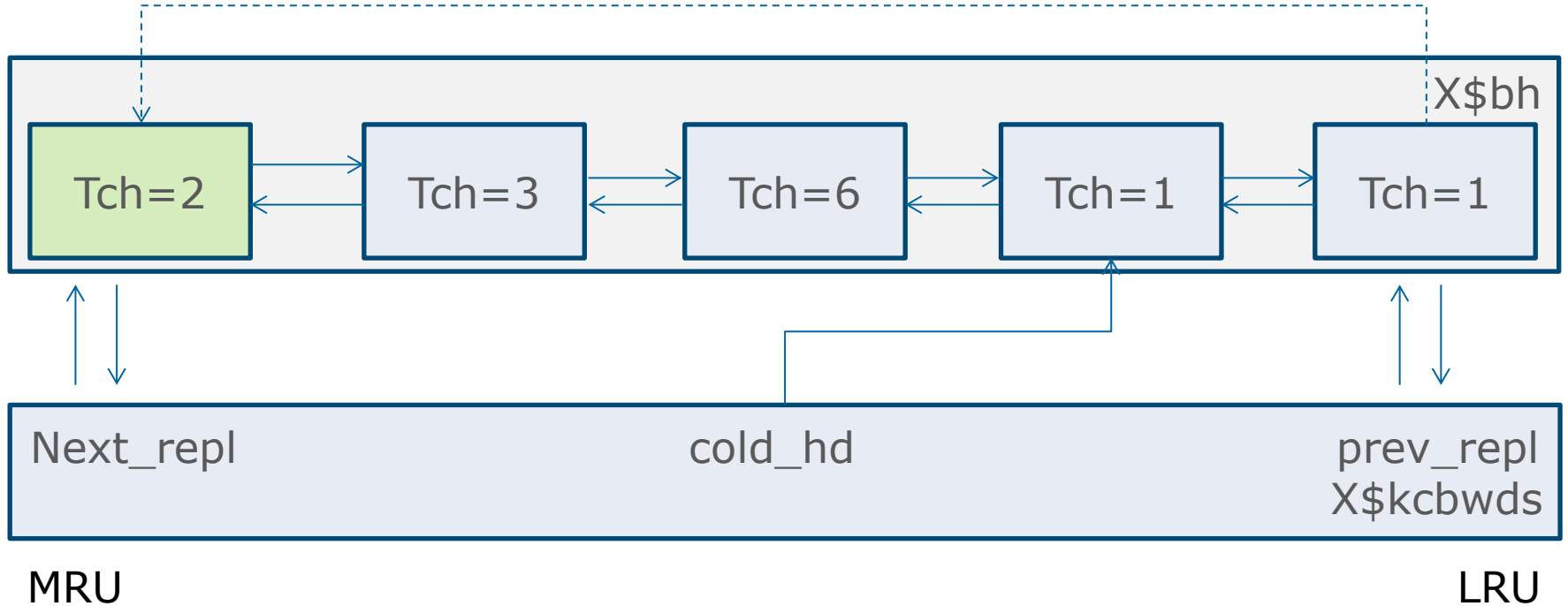
Buffercache

LRU/TCH



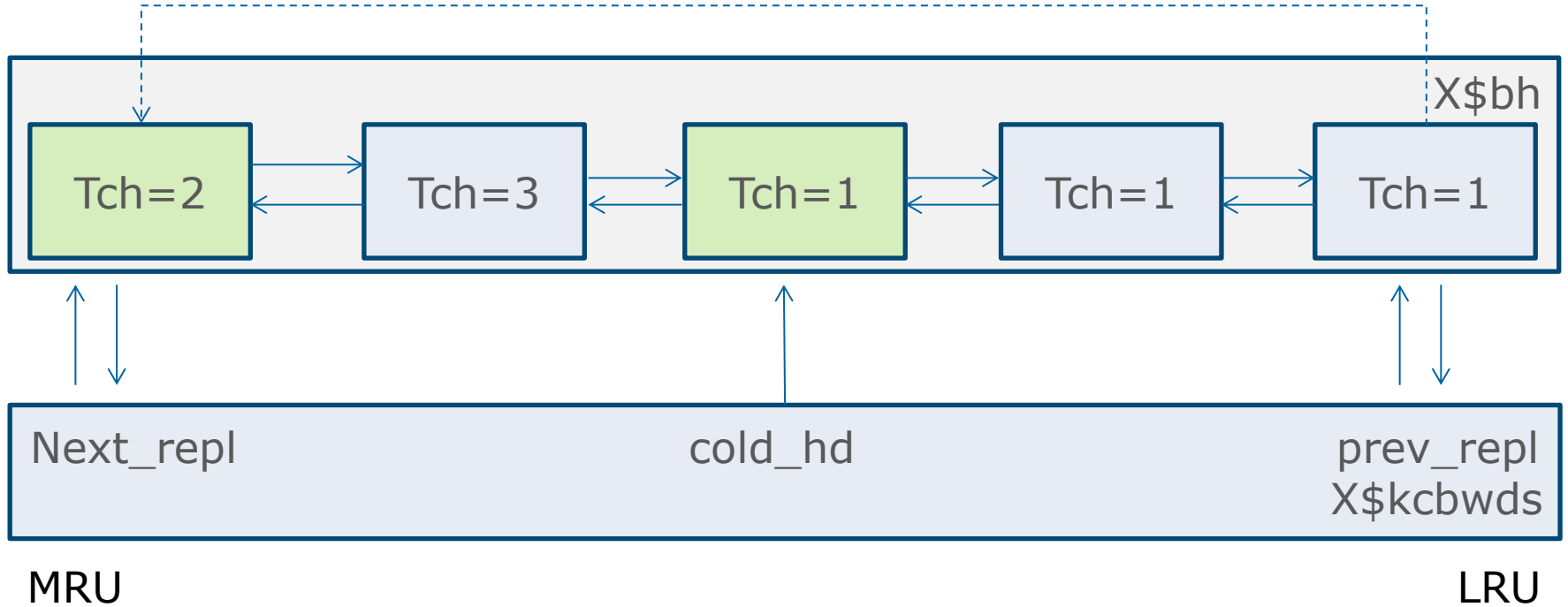
Buffercache

LRU/TCH



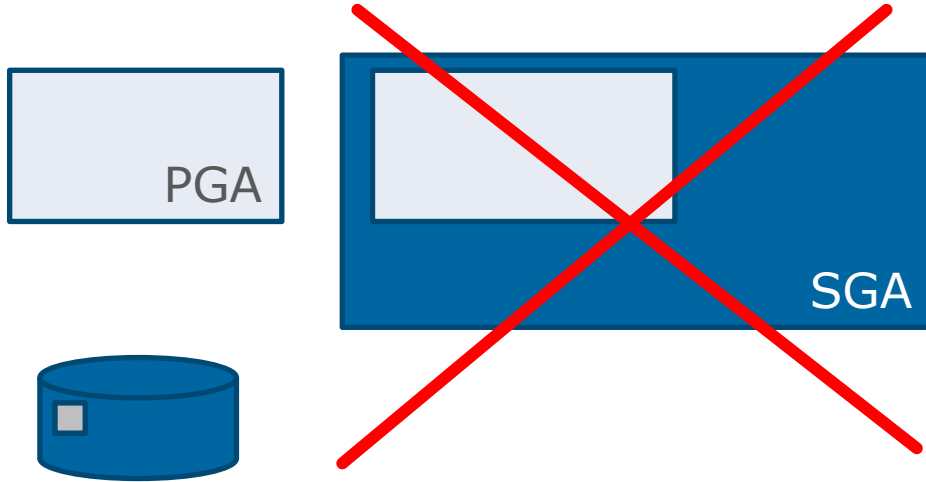
Buffercache

LRU/TCH



Direct Read

- ▶ Umgehen des Buffercache bei sequential reads (TFS, FFS)
- ▶ Segment > „_small_table_threshold“



Direct Read

- ▶ Mehr 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
- ▶ Pro Segment über die Häufigkeit des Zugriffs bestimmt
- ▶ „Temperatur“basiert
- ▶ Parallel Query in RAC und Single Instance
- ▶ Serial Scan nur Single Instance
- ▶ Nur Tabellen, kein Index FFS
- ▶ Kein flush Operator vorhanden

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



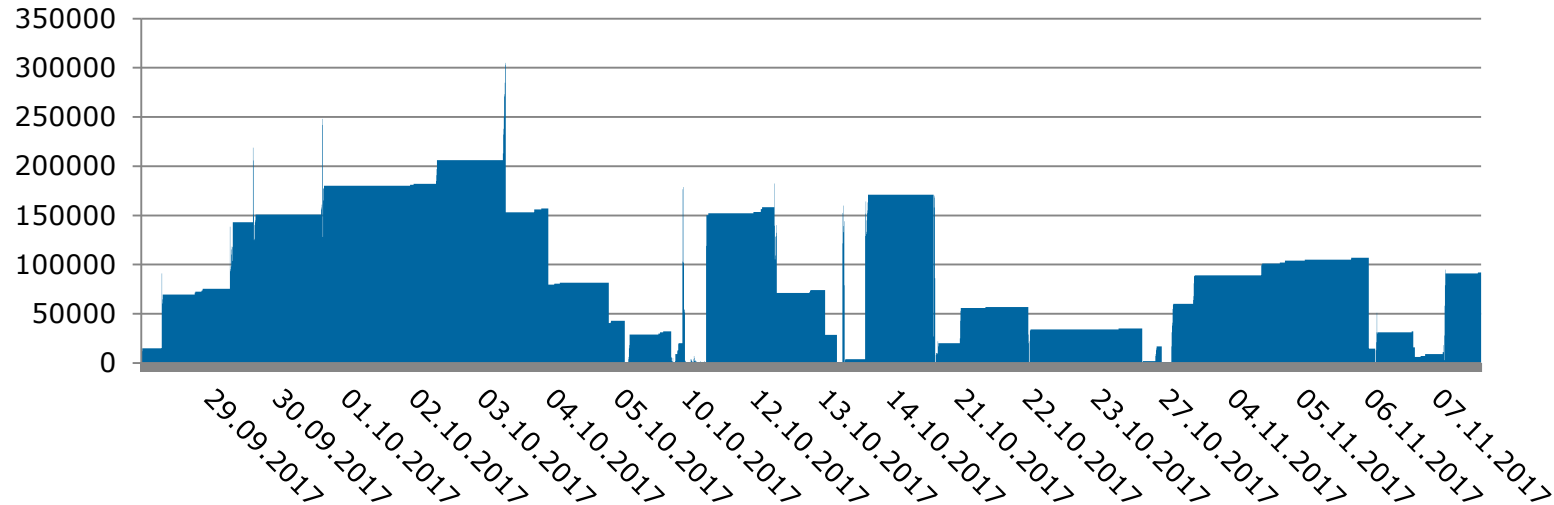
DEMO

ABTC im Einsatz

Langzeit

- ▶ V\$BT_SCAN_OBJ_TEMPS archivieren
- ▶ Verlauf für eine Tabelle über mehrere Tage

TEMPERATURE



ABTC im Einsatz

- ▶ Keine Möglichkeit bestimmte Segmente zu pinnen
- ▶ Mechanismus der Temperatur reduziert nicht dokumentiert
 - Abhängig vom Workload
 - Zeitabhängig

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
- ▶ Ebenso bei Move Table

ABTC im Einsatz

Anomalien

- ▶ 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)

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;
```

Was ist wirklich im Cache?

16K buffer_pool

| OBJ | STATE | CUR | CT |
|------------|-------|-----|-----|
| 73635 | 0 | 0 | 3 |
| | 0 | 0 | 1 |
| | 0 | 0 | 2 |
| 75145 | 0 | 0 | 129 |
| | 0 | 0 | 125 |
| | 0 | 0 | 129 |
| | 0 | 0 | 129 |
| 75374 | 0 | 0 | 18 |
| | 0 | 0 | 18 |
| | 0 | 0 | 17 |
| | 0 | 0 | 18 |
| | 1 | 0 | 101 |
| | 1 | 0 | 102 |
| | 1 | 0 | 101 |
| 4294967295 | 0 | 0 | 1 |
| | 0 | 0 | 1 |

```
select
    obj,
    state,
    bitand(bh.flag, power(2,13)) cur,
    count(*) ct
from
    x$bh bh,
    x$kcwds wds
where wds.addr = bh.set_ds
      and wds.set_id between 25 and 28
group by obj, state, set_id, bitand(bh.flag,
power(2,13))
order by obj, state, set_id, bitand(bh.flag,
power(2,13))
;
```

```
select count(*) from v$bh where objd=75374
SYS@ora12c2 SQL>/

COUNT(*)
-----
         476
```

Zusammenfassung

- ▶ Wenig zu verlieren
- ▶ bestimmte Workloads können profitieren
- ▶ Zuwenig Kontrolle in vielen Fällen
- ▶ Bugs!
- ▶ Nichts für Multiple-Blocksize DB

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

Vielen Dank für die Aufmerksamkeit

Holger Bär

holger.baer@atos.net

Atos