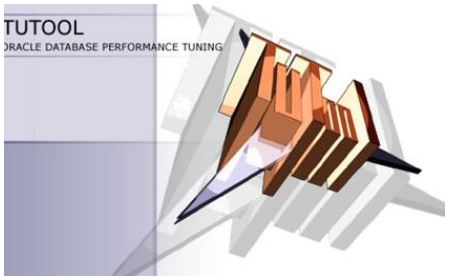
Analyse der Datenbank DW am 8. Oktober 2019.

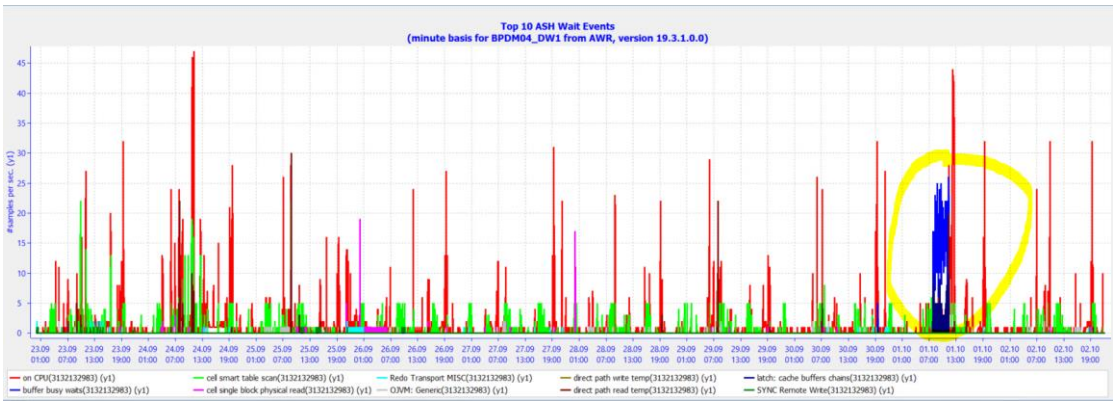
Author:	Leonid Nossov
Creation Date:	8th Oktober 2019
Last Updated:	8th Oktober 2019
Document Reference:	
Version:	Issue 1.1



Inhaltsverzeichnis:

DW

Auf dieser Datenbank habe ich folgendes entdeckt:



Bei Waits auf „buffer busy waits“ ging es um die Waits auf einen temporären File-Header (Wait-Klasse 13). Dieses Problem schien mir sehr ähnlich dem Problem mit Waits auf „enq: SS – contention“ auf BP. Auch hier ist ein Bigfile für den Temp-Tablespace konfiguriert:

Top

Tablespaces

Name	Block Size	initial Extent	next Extent	min Extents	max Extents	% increase	min Extlen	Status	Contents	Logging	Extent management	Segment space management	Allocation type	Plugged in	Default Table Compression	Retention	Bigfile	Predicate Evaluation	En
APEX_2001818539266080	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	NO	STORAGE	NO
APEX_276604290046924 7	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	NO	STORAGE	NO
APEX_46436864747585344	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	NO	STORAGE	NO
APEX_6390587782054250	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	NO	STORAGE	NO
APEX_9091752233063985	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	NO	STORAGE	NO
APEX_9141198233891014	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	NO	STORAGE	NO
AUDIT_AUX	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	YES	STORAGE	NO
DATA_APEX	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	YES	STORAGE	NO
DATA_APEX_EMDM	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	NO	STORAGE	NO
DATA_D_GLOBAL	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	YES	STORAGE	NO
DATA_D_USER	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	YES	STORAGE	NO

…

SYSAUX	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	NO	STORAGE	NO
SYSTEM	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	MANUAL	SYSTEM	NO	DISABLED	NOT APPLY	NO	STORAGE	NO
TEMP_BIG	8192	1048576	1048576	1		0	1048576	ONLINE	TEMPORARY	NOLOGGING	LOCAL	MANUAL	UNIFORM	NO	DISABLED	NOT APPLY	YES	STORAGE	NO
UNDODW1_BIG	8192	65536		1	2147483645		65536	ONLINE	UNDO	LOGGING	LOCAL	MANUAL	SYSTEM	NO	DISABLED	NOGUARANTEE	YES	STORAGE	NO
UNDODW2_BIG	8192	65536		1	2147483645		65536	ONLINE	UNDO	LOGGING	LOCAL	MANUAL	SYSTEM	NO	DISABLED	NOGUARANTEE	YES	STORAGE	NO
UNDODW3_BIG	8192	65536		1	2147483645		65536	ONLINE	UNDO	LOGGING	LOCAL	MANUAL	SYSTEM	NO	DISABLED	NOGUARANTEE	YES	STORAGE	NO
UNDODW4_BIG	8192	65536		1	2147483645		65536	ONLINE	UNDO	LOGGING	LOCAL	MANUAL	SYSTEM	NO	DISABLED	NOGUARANTEE	YES	STORAGE	NO
USERS	8192	65536		1	2147483645		65536	ONLINE	PERMANENT	LOGGING	LOCAL	AUTO	SYSTEM	NO	DISABLED	NOT APPLY	NO	STORAGE	NO

Die nächste Auswertung zeigt, dass lediglich ein SQL überwiegend auf „buffer busy waits“ gewartet hat:

```
-- Database Alias: BPDM04_DW1
-- Oracle Server Version: 19.3.1.0.0
-- Script awr_active_sess_hist_top_sql122.sql (Product TuTool NG 1.8.1.9 : www.tutool.de)
-- Start Time : 01.10.19 21:34:01
-- Input Parameters :
-- begin_time='23.09.2019'
-- end_time=''
-- con_dbid=''
-- awr_root_view=''
-- event_name='buffer busy waits'
-- top_sqls=''
```

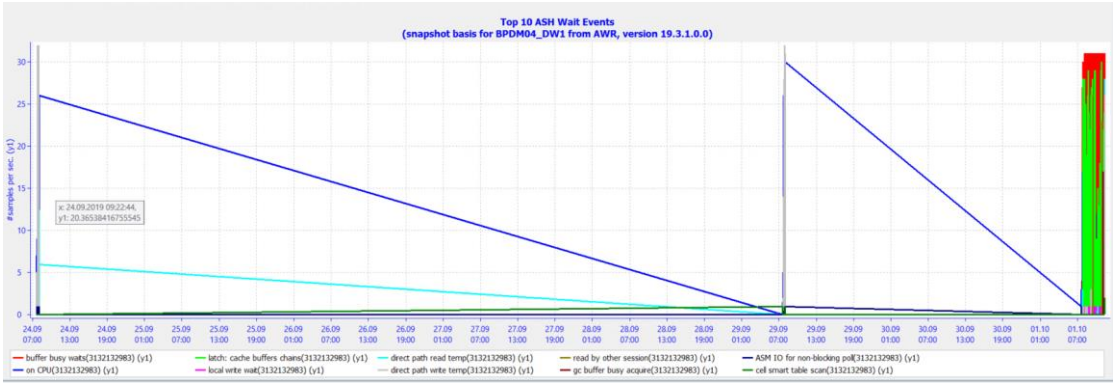
con_dbid	sql_id	cnt
3132132983	gbh7dkx7sd1cu	19454
3132132983	00sxu97uchmta	58
3132132983	9yv5dwv8k0awg	15
3132132983	fhf8upax5cxsz	6
3132132983	6ruhj0uxg6xm0	5
3132132983	0qbzfjt00pbsx	1
3132132983	none	1
3132132983	187p1rhk7bhpu	1

Das ist dieses SQL:

```
---- GROUP=none, SNAP_ID <= 81251, SNAP_TIME <= 01.10.2019 10:00:24, SQL_ID=gbh7dkx7sd1cu, CON_DBID=3132132983
---- EXECUTIONS=2, ROWS_PROCESSED=136899, PARSE_CALLS=193
---- OPTIMIZED_PHYSICAL_READS=153741, OPTIMIZED_PHYSICAL_READS_PER_EX=76870.5000
---- DISK_READS=169498763, BUFFER_GETS=15049051, DIRECT_WRITES=227204993
---- DISK_READS_PER_EX=84749381.5, BUFFER_GETS_PER_EX=7524525.5, DIRECT_WRITES_PER_EX=113602496.5, ROWS_PROCESSED_PER_EX=68449.5, PARSE_CALLS_PER_EX=96.5
---- DISK_READS_PER_ROW=1238.1300, BUFFER_GETS_PER_ROW=109.9281
---- CPU_TIME (sec.)=22756.4640, ELAPSED_TIME (sec.)=223244.7414, WAIT_TIME (sec.)=200488.0794
---- CPU_TIME_PER_EX (sec.)=11378.2320, ELAPSED_TIME_PRO_EX (sec.)=111622.2717, WAIT_TIME_PRO_EX (sec.)=100244.0397
---- PLSQL_EXEC_TIME (sec.)=0.0092, JAVA_EXEC_TIME (sec.)=0.0000
---- PLSQL_EXEC_TIME_PER_EX (sec.)=0.0046, JAVA_EXEC_TIME_PRO_EX (sec.)=0.0000
---- APPLICATION_WAIT_TIME (sec.)=5.0937, CONCURRENCY_WAIT_TIME (sec.)=190294.6723, CLUSTER_WAIT_TIME (sec.)=324.7163, USER_IO_WAIT_TIME (sec.)=14047.0401
---- APPLICATION_WAIT_TIME_PER_EX (sec.)=2.5469, CONCURRENCY_WAIT_TIME_PER_EX (sec.)=95147.3362, CLUSTER_WAIT_TIME_PER_EX (sec.)=162.3581, USER_IO_WAIT_TIME_PER_EX (sec.)=7023.5200
---- IO_CELL_OFFLOAD_ELIGIBLE_BYTES=74740940800, IO_CELL_OFFLOAD_ELIGIBLE_BYTES_PER_EX=37370470400, IO_CELL_OFFLOAD_RETURNED_BYTES=64284206232, IO_CELL_OFFLOAD_RETURNED_BYTES_PER_EX=32142103116
---- IO_INTERCONNECT_BYTES=136382358680, IO_INTERCONNECT_BYTES_PER_EX=68191179340
---- IO_CELL_UNCOMPRESSED_BYTES=10681099508, IO_CELL_UNCOMPRESSED_BYTES_PER_EX=5340549754
---- MODULE : DBMS_SCHEDULER
---- FORCE_MATCHING_SIGNATURE = 4531161893457717080
---- OPTIMIZER_MODE = ALL_ROWS, OPTIMIZER_ENV_HASH_VALUE = 3622871604
INSERT /*+ append */ INTO TMP_FACT_RETURN_DAILY ( DIM_PRODUCT_KEY, DIM_CALENDAR_KEY_ABS, DIM_CALENDAR_KEY_RET, DIM_KDF_KEY, DIM_PROMOTION_KEY, ABS_MG, ABS_WERT, RET_MG, RET_WERT, DIM_RETURN_REASON_KEY,
DUCT_KEY, TO_NUMBER(TO_CHAR(POS.REPORTINGDATE,'yyyymmdd')), TO_NUMBER(TO_CHAR(POR.REPORTINGDATE,'yyyymmdd')), DK.DIM_KDF_KEY, DR.DIM_PROMOTION_KEY, NULL, NULL, SUM(POR.RET_MENGE), SUM(POR.RET_WERT), NVI
, DP.ITEM_NO, DP.SIZE_ASTA_BIN_GROESSE, 'POSITIONS' AS VORSYSTEM FROM ( SELECT P.ORDER_ID, P.ORDERPOS_ID, MIN(P.REPORTINGDATE) AS REPORTINGDATE, DECODE(P.ITEM_PROMOTION,'','?',P.ITEM_PROMOTION) AS PROM
 EKDM_STAGE.TD_EC_POSITIONS P LEFT JOIN EKDM_STAGE.TD_EC_POSITIONS_ABZUG PA ON PA.ORDER_ID = P.ORDER_ID WHERE (P.FIRM_ID = 11 OR P.PROMOTIONSEASON > 137 OR PA.ORDER_ID IS NULL) AND P.RECTYPE = 5 AND P.
CODE(P.ITEM_PROMOTION,'','?',P.ITEM_PROMOTION), P.SAP_MAT_NO, P.PROMOTIONSEASON, P.FIRM_ID) POS JOIN ( SELECT P.ORDER_ID, P.ORDERPOS_ID, P.REPORTINGDATE, SUM(P.KPIVALUEQTY) AS RET_MENGE, SUM(P.KPIVALUE)
GRUND, P.FIRM_ID FROM EKDM_STAGE.TD_EC_POSITIONS P LEFT JOIN EKDM_STAGE.TD_EC_POSITIONS_ABZUG PA ON PA.ORDER_ID = P.ORDER_ID WHERE (P.FIRM_ID = 11 OR P.PROMOTIONSEASON > 137 OR PA.ORDER_ID IS NULL) AND
ER_ID, P.ORDERPOS_ID, P.REPORTINGDATE, P.FIRM_ID) POR ON POS.ORDER_ID = POR.ORDER_ID AND POS.ORDERPOS_ID = POR.ORDERPOS_ID AND POS.FIRM_ID = POR.FIRM_ID JOIN EKDM.V_DIM_PRODUCT DP ON DP.SIZE_SAP_MAT_NR
ASGN AND POS.REPORTINGDATE BETWEEN DP.DIM_ENTRY_DATE AND NVL(DP.DIM_EXIT_DATE, TO_DATE('31.12.2099','dd.mm.yyyy')) JOIN EKDM.V_DIM_KDF DK ON DK.ACTIVITY_NO = DECODE(POS.FIRM_ID, 11, 13, 1, 4) AND DK.LOG
_DATE AND NVL(DK.DIM_EXIT_DATE, TO_DATE('31.12.2099','dd.mm.yyyy')) LEFT JOIN EKDM.V_DIM_PROMOTION DR ON DR.PROMOTION = POS.PROM_ENDS AND POS.REPORTINGDATE BETWEEN DR.DIM_ENTRY_DATE AND NVL(DR.DIM_EXIT_
KDM.DIM_RETURN_REASON RR ON RR.RETURN_REASON = DECODE(SUBSTR(POR.RET_GRUND,2),'00','0',SUBSTR(POR.RET_GRUND,2)) WHERE TO_CHAR(POR.REPORTINGDATE,'yyyymmdd') > DECODE(POS.FIRM_ID, 1, :B2 , :B1 ) GROUP BY
ATE, DK.DIM_KDF_KEY, DR.DIM_PROMOTION_KEY, NVL(RR.DIM_RETURN_REASON_KEY, 1001), POS.PROMOTIONSEASON, DP.ITEM_NO, DP.SIZE_ASTA_BIN_GROESSE
---- Execution Plan (Plan Hash Value : 3721184675) :
INSERT STATEMENT Optimizer=HINT: ALL_ROWS (Cost=2367068)
  LOAD AS SELECT OF TMP_FACT_RETURN_DAILY
    PX COORDINATOR
      PX SEND (QC (RANDOM)) OF :TQ10006 [PARALLEL_TO_SERIAL -> :Q1006] QC (RANDOM) (Cost=2367068 Card=20 Bytes=5320 CPU_Cost=692470283282 IO_Cost=2338882 Time=93)
        OPTIMIZER STATISTICS GATHERING [PARALLEL_COMBINED_WITH_CHILD -> :Q1006] (Cost=2367068 Card=20 Bytes=5320 CPU_Cost=692470283282 IO_Cost=2338882 Time=93)
          HASH (GROUP BY) [PARALLEL_COMBINED_WITH_PARENT -> :Q1006] (Cost=2367068 Card=20 Bytes=5320 CPU_Cost=692470283282 IO_Cost=2338882 Time=93)
            PX RECEIVE [PARALLEL_COMBINED_WITH_PARENT -> :Q1006] (Cost=2367068 Card=20 Bytes=5320 CPU_Cost=692470283282 IO_Cost=2338882 Time=93)
              PX SEND (HASH) OF :TQ10005 [PARALLEL_TO_PARALLEL -> :Q1005] HASH (Cost=2367068 Card=20 Bytes=5320 CPU_Cost=692470283282 IO_Cost=2338882 Time=93)
                HASH (GROUP BY) [PARALLEL_COMBINED_WITH_PARENT -> :Q1005] (Cost=2367068 Card=20 Bytes=5320 CPU_Cost=692470283282 IO_Cost=2338882 Time=93)
                  HASH JOIN (OUTER) [PARALLEL_COMBINED_WITH_PARENT -> :Q1005] (Cost=2367068 Card=20 Bytes=5320 CPU_Cost=692470280018 IO_Cost=2338882 Time=93)
                    HASH JOIN [PARALLEL_COMBINED_WITH_PARENT -> :Q1005] (Cost=2367065 Card=20 Bytes=5180 CPU_Cost=692451059466 IO_Cost=2338880 Time=93)
                      PX RECEIVE [PARALLEL_COMBINED_WITH_PARENT -> :Q1005] (Cost=1683376 Card=115 Bytes=21850 CPU_Cost=428657910502 IO_Cost=1665928 Time=66)
                        PX SEND (HYBRID HASH) OF :TQ10004 [PARALLEL_TO_PARALLEL -> :Q1004] HYBRID HASH (Cost=1683376 Card=115 Bytes=21850 CPU_Cost=428657910502 IO_Cost=1665928 Time=66)
                          STATISTICS COLLECTOR [PARALLEL_COMBINED_WITH_CHILD -> :Q1004]
                            HASH JOIN [PARALLEL_COMBINED_WITH_PARENT -> :Q1004] (Cost=1683376 Card=115 Bytes=21850 CPU_Cost=428657910502 IO_Cost=1665928 Time=66)
                              JOIN FILTER (CREATE) OF :BF0000 [PARALLEL_COMBINED_WITH_PARENT -> :Q1004] (Cost=1662012 Card=121518 Bytes=16647966 CPU_Cost=383647433332 IO_Cost=1646396 Time=65)
                                PX RECEIVE [PARALLEL_COMBINED_WITH_PARENT -> :Q1004] (Cost=1662012 Card=121518 Bytes=16647966 CPU_Cost=383647433332 IO_Cost=1646396 Time=65)
```

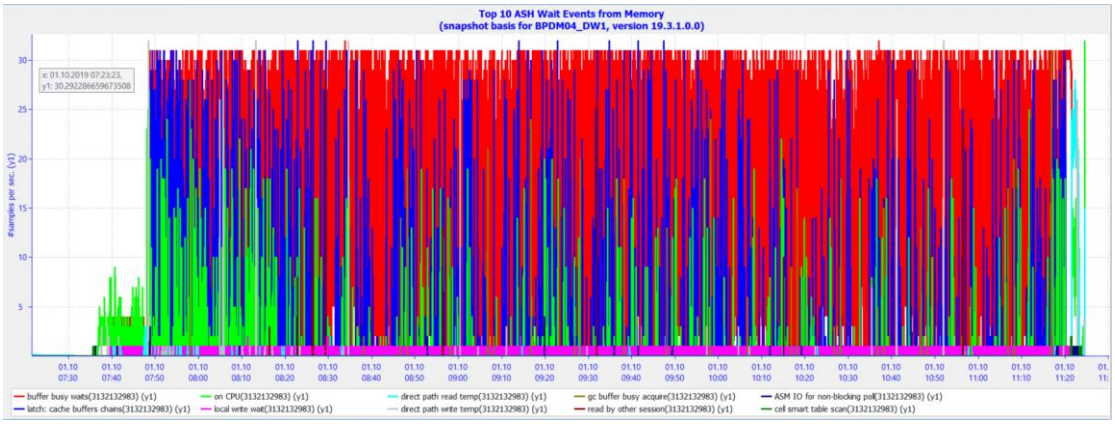
Es ist auffällig, dass dieses SQL nur einen einzigen Ausführungsplan hatte (zumindest laut Informationen aus AWR, s. oben).

Dieses SQL wird nicht sehr oft aber regelmäßig ausgeführt:



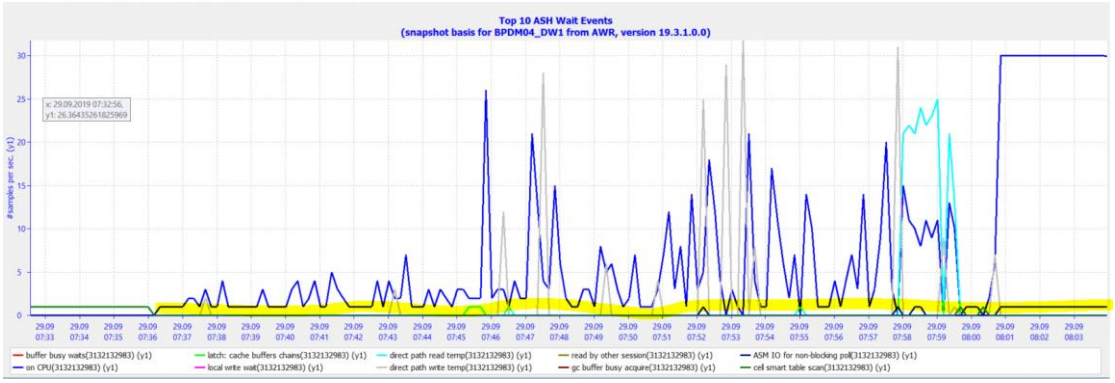
Das Problem mit „buffer busy waits“ trat lediglich einmal am 1. Oktober 2019 auf (s. oben). Dafür aber sehr massiv.

Die untere Grafik zeigt mehr Details dazu. Man sieht, dass das Problem stundenlang dauerte:



Dieses Problem war so gravierend, dass ich keine Daten im historischen SQL-Monitoring zu dieser Ausführung fand.

Deswegen habe ich entschieden eine „normale“ Ausführung von diesem SQL zu untersuchen. Ich nahm diese hier:



Man sieht, dass dieses SQL auch im Normalfall ca. 25 Minuten für seine Ausführung braucht.

Herunter folgt der jeweilige SQL-Monitoring-Report:

```
SQL Monitoring Report

SQL Text
------------------------------
INSERT /*+ append */ INTO TMP_FACT_RETURN_DAILY ( DIM_PRODUCT_KEY, DIM_CALENDAR_KEY_ABS, DIM_CALENDAR_KEY_RET, DIM_KDF_KEY, DIM_PROMOTION_KEY, ABS_MG, ABS_WERT, RET_MG, RET_WERT, DIM_RETURN_REASON_KEY,
POS.PROMOTIONSEASON, DP.ITEM_NO, DP.SIZE_ASTA_BIN_GROESSE, 'POSITIONS' AS VORSYSTEM FROM ( SELECT P.ORDER_ID, P.ORDERPOS_ID, MIN(P.REPORTINGDATE) AS REPORTINGDATE, DECODE(P.ITEM_PROMOTION,'','?',P.ITEM
P.ORDER_ID, P.ORDERPOS_ID, DECODE(P.ITEM_PROMOTION,'','?',P.ITEM_PROMOTION), P.SAP_MAT_NO, P.PROMOTIONSEASON, P.FIRM_ID) POS JOIN ( SELECT P.ORDER_ID, P.ORDERPOS_ID, P.REPORTINGDATE, SUM(P.KPIVALUEQTY)
= 6 AND P.ORDER_ID != '0' GROUP BY P.ORDER_ID, P.ORDERPOS_ID, P.REPORTINGDATE, P.FIRM_ID) POR ON POS.ORDER_ID = POR.ORDER_ID AND POS.ORDERPOS_ID = POR.ORDERPOS_ID AND POS.FIRM_ID = POR.FIRM_ID JOIN EKDM
DK.LOCALE_ID > 0 AND POS.REPORTINGDATE BETWEEN DK.DIM_ENTRY_DATE AND NVL(DK.DIM_EXIT_DATE, TO_DATE('31.12.2099','dd.mm.yyyy')) LEFT JOIN EKDM.V_DIM_PROMOTION DR ON DR.PROMOTION = POS.PROM_ENDZ AND POS.I
, :B1 ) GROUP BY DP.DIM_PRODUCT_KEY, POS.REPORTINGDATE, POR.REPORTINGDATE, DK.DIM_KDF_KEY, DR.DIM_PROMOTION_KEY, NVL(RR.DIM_RETURN_REASON_KEY, 1001), POS.PROMOTIONSEASON, DP.ITEM_NO, DP.SIZE_ASTA_BIN_GI

Global Information
------------------------------
 Status                                 :  DONE
 Instance ID                            :  1
 Session                                :  EKDM_CHECK (736:52697)
 SQL ID                                 :  gbh7dkx7sd1cu
 SQL Execution ID                       :  16777216
 Execution Started                      :  09/29/2019 07:35:37
 First Refresh Time                     :  09/29/2019 07:35:37
 Last Refresh Time                      :  09/29/2019 08:00:52
 Duration                               :  1515s
 Module/Action                          :  DBMS_SCHEDULER/FACT_RETURN_DAILY
 Service                                :  DM_S
 Program                                :
 PLSQL Entry Ids (Object/Subprogram)    :
 PLSQL Current Ids (Object/Subprogram)  :  3892204,4

Binds
========================================================================================
| Name | Position |  Type  |                        Value                              |
========================================================================================
| :B2  |        1 | NUMBER | 20190927                                                  |
| :B1  |        2 | NUMBER | 20190927                                                  |
========================================================================================
```

…

```
SQL Plan Monitoring Details (Plan Hash Value=1721184875)
=====================================================================================================================================================================================================
| Id |                  Operation                  |         Name          |  Rows   | Cost  |   Time    | Start  | Execs |   Rows   | Read  | Read  | Write | Write |  Mem  | Temp  | Activit
|    |                                             |                       | (Estim) |       | Active(s) | Active |       | (Actual) | Reqs  | Bytes | Reqs  | Bytes | (Max) | (Max) |   (%)
=====================================================================================================================================================================================================
|  0 | INSERT STATEMENT                            |                       |         |       |      1346 |   +170 |     1 |        2 |       |       |       |       |     . |     . |
|  1 |   LOAD AS SELECT                            | TMP_FACT_RETURN_DAILY |         |       |         1 |  +1515 |     1 |        2 |     3 |   2MB |     3 |   2MB |     . |     . |
|  2 |    PX COORDINATOR                           |                       |         |       |         1 |  +1515 |    65 |    23507 |       |       |       |       |     . |     . |
|  3 |     PX SEND QC (RANDOM)                     | :TQ10006              |     115 |    2M |         1 |  +1515 |    32 |    23507 |       |       |       |       |     . |     . |
|  4 |      OPTIMIZER STATISTICS GATHERING         |                       |     115 |    2M |         1 |  +1515 |    32 |        0 |       |       |       |       |   8MB |     . |
|  5 |       HASH GROUP BY                         |                       |     115 |    2M |         1 |  +1515 |    32 |    23507 |       |       |       |       | 152MB |     . |
|  6 |        PX RECEIVE                           |                       |     115 |    2M |         1 |  +1515 |    32 |    23691 |       |       |       |       |     . |     . |
|  7 |         PX SEND HASH                        | :TQ10005              |     115 |    2M |         1 |  +1515 |    32 |    23691 |       |       |       |       |     . |     . |
|  8 |          HASH GROUP BY                      |                       |     115 |    2M |         1 |  +1515 |    32 |    23691 |       |       |       |       | 152MB |     . |
|  9 |           HASH JOIN OUTER                   |                       |     115 |    2M |         3 |  +1513 |    32 |    23695 |       |       |       |       | 134MB |     . |
| 10 |            HASH JOIN                        |                       |     115 |    2M |       176 |  +1340 |    32 |    23695 | 12658 |   6GB | 12658 |   6GB |  25GB |   6GB |
| 11 |             PX RECEIVE                      |                       |     115 |    2M |       147 |  +1340 |    32 |     113M |       |       |       |       |     . |     . |
| 12 |              PX SEND HYBRID HASH            | :TQ10004              |     115 |    2M |       176 |  +1340 |    32 |     113M |       |       |       |       |     . |     . |
| 13 |               STATISTICS COLLECTOR          |                       |         |       |       175 |  +1341 |    32 |     113M |       |       |       |       |     . |     . |
| 14 |                HASH JOIN                    |                       |     115 |    2M |       908 |   +608 |    32 |     113M |    1M | 547GB |    1M | 547GB |  18GB | 556GB |
| 15 |                 JOIN FILTER CREATE          | :BF0000               |    122K |    2M |       730 |   +608 |    32 |       4G |       |       |       |       |     . |     . |
| 16 |                  PX RECEIVE                 |                       |    122K |    2M |       730 |   +608 |    32 |       4G |       |       |       |       |     . |     . |
| 17 |                   PX SEND BROADCAST         | :TQ10003              |    122K |    2M |       733 |   +607 |    32 |       4G |       |       |       |       |     . |     . |
| 18 |                    HASH JOIN RIGHT OUTER    |                       |    122K |    2M |      1251 |    +89 |    32 |     113M |       |       |       |       | 147MB |     . |
| 19 |                     PX RECEIVE              |                       |    2722 |     2 |         1 |    +89 |    32 |    87104 |       |       |       |       |     . |     . |
| 20 |                      PX SEND BROADCAST      | :TQ10002              |    2722 |     2 |         1 |   +607 |    32 |    87104 |       |       |       |       |     . |     . |
| 21 |                       PX SELECTOR           |                       |         |       |         1 |   +607 |    32 |     2722 |       |       |       |       |     . |     . |
| 22 |                        TABLE ACCESS STORAGE FULL | DIM_PROMOTION    |    2722 |     2 |         1 |   +607 |     1 |     2722 |       |       |       |       |     . |     . |
| 23 |                     HASH JOIN               |                       |    122K |    2M |      1251 |    +89 |    32 |     113M |       |       |       |       |  28MB |     . |
| 24 |                      PX BLOCK ITERATOR      |                       |      30 |     2 |         1 |    +89 |    32 |       30 |       |       |       |       |     . |     . |
| 25 |                       TABLE ACCESS STORAGE FULL | DIM_KDF           |      30 |     2 |         1 |    +89 |     5 |       30 |       |       |       |       |     . |     . |
| 26 |                      BUFFER SORT            |                       |         |       |      1250 |    +90 |     5 |     567M | 95175 |  46GB | 95175 |  46GB | 488MB |  46GB |
| 27 |                       PX RECEIVE            |                       |    109M |    2M |       518 |    +90 |    32 |     567M |       |       |       |       |     . |     . |
| 28 |                        PX SEND BROADCAST    | :TQ10000              |    109M |    2M |       518 |    +89 |     1 |       4G |       |       |       |       |     . |     . |
| 29 |                         VIEW                |                       |    109M |    2M |       517 |    +90 |     1 |     113M |       |       |       |       |     . |     . |
| 30 |                          HASH GROUP BY      |                       |    109M |    2M |       605 |     +2 |     1 |     113M | 18754 |   9GB | 18754 |   9GB |   1GB |   9GB |
| 31 |                           FILTER            |                       |         |       |        89 |     +2 |     1 |     116M |       |       |       |       |     . |     . |
| 32 |                            HASH JOIN RIGHT OUTER |                  |    109M | 59308 |        89 |     +2 |     1 |     116M |       |       |       |       |  66MB |     . |
| 33 |                             PARTITION RANGE ALL |                   |      1M |    23 |         1 |     +2 |     1 |       1M |       |       |       |       |     . |     . |
| 34 |                              TABLE ACCESS STORAGE FULL | TD_EC_POSITIONS_ABZUG | 1M | 23 |     1 |     +2 |   516 |       1M |       |       |       |       |     . |     . |
| 35 |                             PARTITION RANGE ALL |                   |    109M | 30853 |        89 |     +2 |     1 |     116M |       |       |       |       |     . |     . |
| 36 |                              TABLE ACCESS STORAGE FULL | TD_EC_POSITIONS | 109M | 30853 |    90 |     +1 |   996 |     116M | 19692 |  14GB |       |       |   7MB |     . |
| 37 |                 JOIN FILTER USE             | :BF0000               |     24M | 31240 |         5 |  +1337 |    32 |      18M |       |       |       |       |     . |     . |
| 38 |                  PX BLOCK ITERATOR          |                       |     24M | 31240 |         5 |  +1337 |    32 |      18M |       |       |       |       |     . |     . |
| 39 |                   TABLE ACCESS STORAGE FULL | DIM_PRODUCT           |     24M | 31240 |         5 |  +1337 |   669 |      18M | 22001 |  19GB |       |       | 276MB |     . |
```

…

In diesem Report fällt auf, dass die Kardinalität in einigen Ausführungsplanschritten auf 4G ansteigt. Das passiert wegen Broadcast-Distribution bei Hash-Joins. Bei dieser Distribution sendet Oracle die Daten an alle Receiver. Bei DOP=32 sind es gerade 32. Das bedeutet, dass Oracle die Daten 32mal multipliziert, wofür lezten Endes viel Platz im Temp-Tablespace verbraucht wird (in der oberen Ausführung 556 G).

Weswegen Oracle eine Broadcast-Distribution für 113M Rows anwendet, ist mir nicht klar. Ich kann nicht ausschließen, dass das wieder ein Bug ist.

Ich habe entschieden, die Broadcast-Distribution für dieses SQL zu verbieten und zugleich die DOP auf 8 zu reduzieren. Für mein Experiment nahm ich den Select aus dem ursprünglichen Insert.

Meine Idee war richtig: Ich habe die Laufzeit auf ca. 2,5 Minuten reduziert. Das Allozieren im Temp-Tablespace ging auch zurück auf 11G:

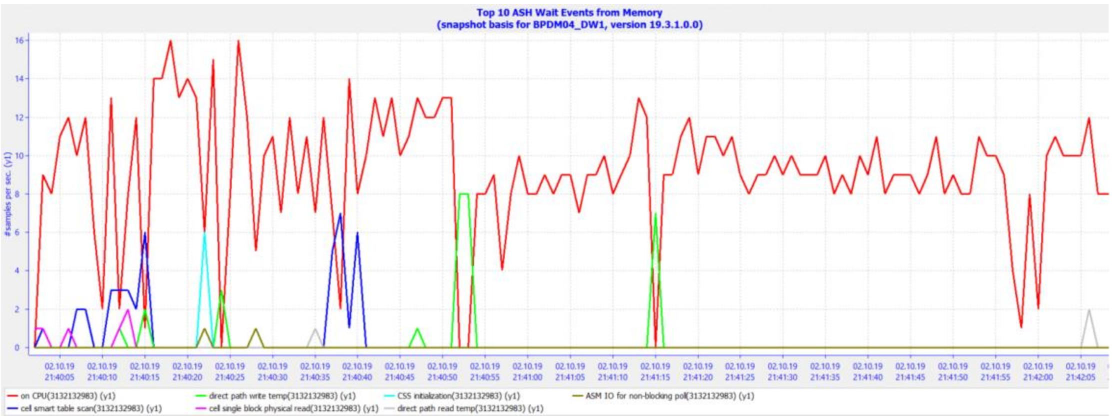
```
SQL Monitoring Report

SQL Text
------------------------------
SELECT /*+ opt_param('_parallel_broadcast_enabled' 'false') parallel(8) */ DP.DIM_PRODUCT_KEY, TO_NUMBER(TO_CHAR(POS.REPORTINGDATE,'yyyymmdd')), TO_NUMBER(TO_CHAR(POR.REPORTINGDATE,'yyyymmdd')), DK.DIM
DECODE(P.ITEM_PROMOTION,'','?',P.ITEM_PROMOTION) AS PROM_ENDZ, P.SAP_MAT_NO, P.PROMOTIONSEASON, P.FIRM_ID FROM EKDM_STAGE.TD_EC_POSITIONS P LEFT JOIN EKDM_STAGE.TD_EC_POSITIONS_ABZUG PA ON PA.ORDER_ID
P.ORDERPOS_ID, P.REPORTINGDATE, SUM(P.KPIVALUEQTY) AS RET_MENGE, SUM(P.KPIVALUE) AS RET_WERT, MAX(NVL(P.RETURNREASONCD, '0')) AS RET_GRUND, P.FIRM_ID FROM EKDM_STAGE.TD_EC_POSITIONS P LEFT JOIN EKDM_ST
POR.ORDERPOS_ID AND POS.FIRM_ID = POR.FIRM_ID JOIN EKDM.V_DIM_PRODUCT DP ON DP.SIZE_SAP_MAT_NR = POS.SAP_MAT_NO AND DP.EKSEASON_ID = POS.PROMOTIONSEASON AND POS.REPORTINGDATE BETWEEN DP.DIM_ENTRY_DATE
EKDM.V_DIM_PROMOTION DR ON DR.PROMOTION = POS.PROM_ENDZ AND POS.REPORTINGDATE BETWEEN DR.DIM_ENTRY_DATE AND NVL(DR.DIM_EXIT_DATE, TO_DATE('31.12.2099','dd.mm.yyyy')) LEFT JOIN EKDM.DIM_RETURN_REASON RR
NVL(RR.DIM_RETURN_REASON_KEY, 1001), POS.PROMOTIONSEASON, DP.ITEM_NO, DP.SIZE_ASTA_BIN_GROESSE

Global Information
------------------------------
 Status              :  DONE (FIRST N ROWS)
 Instance ID         :  1
 Session             :  SYS (1037:2250)
 SQL ID              :  3635wpwj52vkw
 SQL Execution ID    :  16777216
 Execution Started   :  10/02/2019 21:40:01
 First Refresh Time  :  10/02/2019 21:40:02
 Last Refresh Time   :  10/02/2019 21:42:28
 Duration            :  147s
 Module/Action       :  sqlplus.exe/-
 Service             :  SYS$USERS
 Program             :  sqlplus.exe
 Fetch Calls         :  727

Binds
========================================================================================
| Name | Position |  Type  |                           Value                           |
========================================================================================
| :B2  |        1 | NUMBER | 20190927                                                   |
| :B1  |        2 | NUMBER | 20190927                                                   |
========================================================================================
```

…

```
SQL Plan Monitoring Details (Plan Hash Value=3613907456)
===================================================================================================================================================================================================
| Id |                Operation                  |   Name   |  Rows   | Cost |   Time    | Start  | Execs |   Rows   | Read  | Read  | Write | Write |  Mem  | Temp  | A
|    |                                           |          | (Estim) |      | Active(s) | Active |       | (Actual) | Reqs  | Bytes | Reqs  | Bytes | (Max) | (Max) |
===================================================================================================================================================================================================
|  0 | SELECT STATEMENT                          |          |         |      |       102 |    +46 |    17 |    10891 |       |       |       |       |     . |     . |
|  1 |  PX COORDINATOR                           |          |         |      |        19 |   +129 |    17 |    10891 |  2291 |  18MB |       |       |     . |     . |
|  2 |   PX SEND QC (RANDOM)                     | :TQ10015 |       1 | 293K |        18 |   +130 |     8 |    19921 |       |       |       |       |     . |     . |
|  3 |    HASH GROUP BY                          |          |       1 | 293K |        18 |   +130 |     8 |    18432 |       |       |       |       |  40MB |     . |
|  4 |     PX RECEIVE                            |          |       1 | 293K |        18 |   +130 |     8 |     227K |       |       |       |       |     . |     . |
|  5 |      PX SEND HASH                         | :TQ10014 |       1 | 293K |         1 |   +129 |     8 |     227K |       |       |       |       |     . |     . |
|  6 |       HASH GROUP BY                       |          |       1 | 293K |         1 |   +129 |     8 |     227K |       |       |       |       |  61MB |     . |
|  7 |        HASH JOIN OUTER                    |          |       1 | 293K |         1 |   +129 |     8 |     234K |       |       |       |       |  43MB |     . |
|  8 |         PX RECEIVE                        |          |       1 | 293K |         1 |   +129 |     8 |     234K |       |       |       |       |     . |     . |
|  9 |          PX SEND HASH (NULL RANDOM)       | :TQ10012 |       1 | 293K |        18 |   +130 |     8 |     234K |       |       |       |       |     . |     . |
| 10 |           BUFFER SORT                     |          |       1 | 293K |        28 |   +120 |     8 |     234K |       |       |       |       |  22MB |     . |
| 11 |            NESTED LOOPS OUTER             |          |       1 | 293K |        28 |   +120 |     8 |     234K |       |       |       |       |     . |     . |
| 12 |             HASH JOIN                     |          |       1 | 293K |        72 |    +76 |     8 |     234K | 13549 |   7GB | 13549 |   7GB |   8GB |   7GB |
| 13 |              JOIN FILTER CREATE           | :BF0000  |     115 | 209K |        43 |    +76 |     8 |     114M |       |       |       |       |     . |     . |
| 14 |               PX RECEIVE                  |          |     115 | 209K |        44 |    +75 |     8 |     114M |       |       |       |       |     . |     . |
| 15 |                PX SEND HYBRID HASH        | :TQ10010 |     115 | 209K |        47 |    +72 |     8 |     114M |       |       |       |       |     . |     . |
| 16 |                 STATISTICS COLLECTOR      |          |         |      |        47 |    +72 |     8 |     114M |       |       |       |       |     . |     . |
| 17 |                  HASH JOIN BUFFERED       |          |     115 | 209K |        79 |    +40 |     8 |     114M | 11117 |  11GB | 11117 |  11GB |   8GB |  11GB |
|    |                                           |          |         |      |           |        |       |          |       |       |       |       |       |       |
| 18 |                   JOIN FILTER CREATE      | :BF0001  |    122K | 205K |        13 |    +40 |     8 |     114M |       |       |       |       |     . |     . |
| 19 |                    PX RECEIVE             |          |    122K | 205K |        12 |    +41 |     8 |     114M |       |       |       |       |     . |     . |
| 20 |                     PX SEND HYBRID HASH   | :TQ10007 |    122K | 205K |        16 |    +37 |     8 |     114M |       |       |       |       |     . |     . |
| 21 |                      STATISTICS COLLECTOR |          |         |      |        16 |    +37 |     8 |     114M |       |       |       |       |     . |     . |
| 22 |                       HASH JOIN BUFFERED  |          |    122K | 205K |        27 |    +26 |     8 |     114M |  9768 |   9GB |  9768 |   9GB | 209MB |  10GB |
|    |                                           |          |         |      |           |        |       |          |       |       |       |       |       |       |
| 23 |                        JOIN FILTER CREATE | :BF0002  |      30 |    2 |         1 |    +27 |     8 |      240 |       |       |       |       |     . |     . |
| 24 |                         PX RECEIVE        |          |      30 |    2 |         1 |    +27 |     8 |      240 |       |       |       |       |     . |     . |
| 25 |                          PX SEND HYBRID HASH | :TQ10003 |   30 |    2 |         1 |    +27 |     8 |      240 |       |       |       |       |     . |     . |
| 26 |                           STATISTICS COLLECTOR |     |         |      |         1 |    +27 |     8 |       30 |       |       |       |       |     . |     . |
| 27 |                            PX BLOCK ITERATOR |      |      30 |    2 |         1 |    +27 |     8 |       30 |       |       |       |       |     . |     . |
| 28 |                             TABLE ACCESS STORAGE FULL | DIM_KDF | 30 | 2 |    1 |    +27 |     5 |       30 |       |       |       |       |     . |     . |
| 29 |                        PX RECEIVE         |          |    109M | 205K |        12 |    +26 |     8 |     114M |       |       |       |       |     . |     . |
| 30 |                         PX SEND HYBRID HASH | :TQ10004 |  109M | 205K |        12 |    +26 |     8 |     114M |       |       |       |       |     . |     . |
| 31 |                          JOIN FILTER USE  | :BF0002  |    109M | 205K |        11 |    +27 |     8 |     114M |       |       |       |       |     . |     . |
| 32 |                           VIEW            |          |    109M | 205K |        11 |    +27 |     8 |     114M |       |       |       |       |     . |     . |
| 33 |                            HASH GROUP BY  |          |    109M | 205K |        23 |    +15 |     8 |     114M |  3674 |   4GB |  3674 |   4GB |   8GB |   4GB |
```

…

Das ist die jeweilige Grafik zur obigen Ausführung:



Ich habe nach bekannten Problemen für Broadcast-Distribution im MOS gesucht. Leider nichts außer diesem alten Bug gefunden:

Bug 4911999 : LOW COSTS LEAD TO BAD PLAN WITH PX SEND BROADCAST, TEMP SPACE CONSUMPTION HIGH

Bug Attributes

Type	B - Defect	**Fixed in Product Version**	
Severity	2 - Severe Loss of Service	**Product Version**	10.1.0.4
Status	91 - Closed, Could Not Reproduce	**Platform**	46 - Linux x86
Created	27-Dec-2005	**Platform Version**	
Updated	02-Feb-2007	**Base Bug**	N/A
Database Version	10.1.0.4	**Affects Platforms**	Generic
Product Source	Oracle	Knowledge, Patches and Bugs related to this bug	

Ich kann nicht ganz ausschließen, dass die System-Statistiken diese Distribution verursachen:

```
-- Database Alias: BPDM04_DW1
-- Oracle Server Version: 19.3.1.0.0
-- Script optimizer_system_stats10.sql (Product TuTool NG 1.8.1.9 : www.tutool.de)
-- Start Time : 03.10.19 09:43:27

SNAME                          PNAME                                                                                  PVAL1 PVAL2
------------------------------ -------------------------------------------------------------------------------------- ----------- ------------------------
SYSSTATS_INFO                  DSTART                                                                                            02-28-2019 11:59
SYSSTATS_INFO                  DSTOP                                                                                             02-28-2019 11:59
SYSSTATS_INFO                  FLAGS                                                                                           1
SYSSTATS_INFO                  STATUS                                                                                            COMPLETED
SYSSTATS_MAIN                  CPUSPEEDNW(the speed of the processor)                                                       2447
SYSSTATS_MAIN                  CPUSPEED(the average speed of the processor, in million/sec)
SYSSTATS_MAIN                  IOSEEKTIM(the speed of the disk head seek time, in ms)                                         10
SYSSTATS_MAIN                  IOTFRSPEED(the speed of disk reads per read operation, in bytes/ms)                        204800
SYSSTATS_MAIN                  MAXTHR(the maximum amount of I/O bandwidth, in bytes per second)
SYSSTATS_MAIN                  MBRC(the average number of blocks in one I/O operation at a full table or index scans)        128
SYSSTATS_MAIN                  MREADTIM(the average time in ms required for reading multiple blocks at a time)
SYSSTATS_MAIN                  SLAVETHR(the average throughput of parallel slave processes, in bytes per second)
SYSSTATS_MAIN                  SREADTIM(the average time in ms to read the single block)
```

Diese Statistiken zeigen eine gute I/O-Geschwindigkeit. Möglicherweise kommt Oracle deswegen auf die Idee mit Broadcast-Distribution und verlagert die Ausführung auf den Temp-Tablespace.

Verbesserungsvorschlag:

Ich glaube nicht, dass es sich lohnt auf Bigfile für den Temp-Tablespace wie auf BP zu verzichten. Dieses Problem ist einmalig und läßt sich mit SQL-Tuning beheben.

Man kann 2 Hints: PARALLEL(8) und OPT_PARAM('_parallel_broadcast_enabled' 'false') in den Programmcode des jeweiligen Insert einfügen.

Das problematische SQL wird von 2 anderen SQLs aufgerufen:

```
-- Database Alias: BPDM04_DW1
-- Oracle Server Version: 19.3.1.0.0
-- Script top_and_recursive_sqlid121.sql (Product TuTool NG 1.8.1.9 : www.tutool.de)
-- Start Time : 08.10.19 08:09:18
-- Input Parameters :
-- sql_id='gbh7dkx7sdlcu'


top sqlid     recursive sqlid
------------- ---------------
ak42950jdat2t gbh7dkx7sdlcu
a3ygkkhbuasny gbh7dkx7sdlcu
gbh7dkx7sdlcu gbh7dkx7sdlcu
```

Ein SQL ist das folgende Package:

```
---- GROUP=none, SNAP_ID <= 81345, SNAP_TIME <= 05.10.2019 08:00:24, SQL_ID=ak42950jdat2t, CON_DBID=3132132983
---- EXECUTIONS=6, ROWS_PROCESSED=0, PARSE_CALLS=25
---- OPTIMISED_PHYSICAL_READS=46167748, OPTIMISED_PHYSICAL_READS_PER_EX=7694624.6667
---- DISK_READS=506180414, BUFFER_GETS=20645000639, DIRECT_WRITES=38341059
---- DISK_READS_PER_EX=84363402.33, BUFFER_GETS_PER_EX=3440833439.83, DIRECT_WRITES_PER_EX=6390176.5, ROWS_PROCESSED_PER_EX=0, PARSE_CALLS_PER_EX=4.17
---- DISK_READS_PER_ROW=506180414.0000, BUFFER_GETS_PER_ROW=##############
---- CPU_TIME (sec.)=108518.6413, ELAPSED_TIME (sec.)=122456.4916, WAIT_TIME (sec.)=13937.8503
---- CPU_TIME_PER_EX (sec.)=18086.4402, ELAPSED_TIME_PRO_EX (sec.)=20409.4153, WAIT_TIME_PRO_EX (sec.)=2322.9751
---- PLSQL_EXEC_TIME (sec.)=0.0000, JAVA_EXEC_TIME (sec.)=0.0000
---- PLSQL_EXEC_TIME_PER_EX (sec.)=0.0000, JAVA_EXEC_TIME_PRO_EX (sec.)=0.0000
---- APPLICATION_WAIT_TIME (sec.)=67.0277, CONCURRENCY_WAIT_TIME (sec.)=13.5139, CLUSTER_WAIT_TIME (sec.)=3477.8390, USER_IO_WAIT_TIME (sec.)=13270.8069
---- APPLICATION_WAIT_TIME_PER_EX (sec.)=11.1713, CONCURRENCY_WAIT_TIME_PER_EX (sec.)=2.2523, CLUSTER_WAIT_TIME_PER_EX (sec.)=579.6398, USER_IO_WAIT_TIME_PER_EX (sec.)=2211.8012
---- IO_CELL_OFFLOAD_ELIGIBLE_BYTES=3479272882176, IO_CELL_OFFLOAD_ELIGIBLE_BYTES_PER_EX=579878813696, IO_CELL_OFFLOAD_RETURNED_BYTES=1098125885352, IO_CELL_OFFLOAD_RETURNED_BYTES_PER_EX=183020980892
---- IO_INTERCONNECT_BYTES=2079572909992, IO_INTERCONNECT_BYTES_PER_EX=346595484999
---- IO_CELL_UNCOMPRESSED_BYTES=87151582643, IO_CELL_UNCOMPRESSED_BYTES_PER_EX=14525263774
---- MODULE : DBMS_SCHEDULER
---- OPTIMIZER_MODE = ALL_ROWS, OPTIMIZER_ENV_HASH_VALUE = 3002913072
call pkg_tkdm_fact_return_daily.prc_fact_return_daily_main ( )
```

Das 2. SQL ist auch dieses Package aber anders ausgeführt:

```
-- Database Alias: BPDM04_DW1
-- Oracle Server Version: 19.3.1.0.0
-- Script create_sqltext_from_awr12.sql (Product TuTool NG 1.8.1.9 : www.tutool.de)
-- Start Time : 08.10.19 08:21:25
-- Input Parameters :
-- sql_id='a3ygkkhbuasny'
```

```
1  begin
2    pkg_ekdm_fact_return_daily.prc_fact_return_daily_main;
3* end;
```

Man könnte auch überlegen und eventuell '_parallel_broadcast_enabled' = false systemweit einsetzen. Möglicherweise verliert man an einigen Stellen etwas an der Performance. Dafür beseitigt man solche Ausreißer.

Man könnte auf dem StandBy testen, ob das Löschen von System-Statistiken das Verhalten mit Broadcast-Distribution ändert.