===============================================================================

**steps to compare the plans and force the optimizer to good one**

================================================================================

Below query will help you to check the plan history for that particular SQL\_ID :

set lines 155

col execs for 999,999,999

col avg\_etime for 999,999.999

col avg\_lio for 999,999,999.9

col begin\_interval\_time for a30

col node for 99999

break on plan\_hash\_value on startup\_time skip 1

select ss.snap\_id, ss.instance\_number node, begin\_interval\_time, sql\_id, plan\_hash\_value,

nvl(executions\_delta,0) execs,

(elapsed\_time\_delta/decode(nvl(executions\_delta,0),0,1,executions\_delta))/1000000 avg\_etime,

(buffer\_gets\_delta/decode(nvl(buffer\_gets\_delta,0),0,1,executions\_delta)) avg\_lio

from DBA\_HIST\_SQLSTAT S, DBA\_HIST\_SNAPSHOT SS

where sql\_id = nvl('&sql\_id','xxxxxxxxxxxxx')

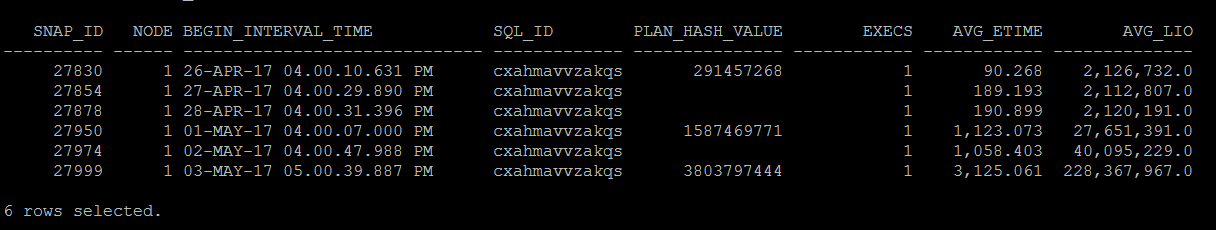
and ss.snap\_id = S.snap\_id

and ss.instance\_number = S.instance\_number

and executions\_delta > 0

order by 1, 2, 3

/



As per the above output we have 3 plans in the awr history and plan got changed on 03-MAY-17. If we compare the plans, current plan 3803797444 is taking more AVG\_ETIME and AVG\_LIO and the plan with PLAN\_HASH\_VALUE 291457268 looks good compared to the others.

**Step 2)** if you want to force the optimizer to the good plan please copy the coe\_xfr\_sql\_profile.sql script and run it as mentioned below with good plan 291457268. After you run this script it will generate one more sql and ask you to run it as mentioned in next step

**SQL> @coe\_xfr\_sql\_profile.sql**

Parameter 1:

SQL\_ID (required)

Enter value for 1: g1xp7kq3g6qvg

PLAN\_HASH\_VALUE AVG\_ET\_SECS

--------------- -----------

203366862 23.556

2602137179 159.416

2399382397 160.663

Parameter 2:

PLAN\_HASH\_VALUE (required)

Enter value for 2: 203366862

Values passed:

~~~~~~~~~~~~~

SQL\_ID : "g1xp7kq3g6qvg"

PLAN\_HASH\_VALUE: "203366862"

Execute coe\_xfr\_sql\_profile\_g1xp7kq3g6qvg \_203366862.sql

on TARGET system in order to create a custom SQL Profile

with plan 203366862 linked to adjusted sql\_text.

COE\_XFR\_SQL\_PROFILE completed.

**Step 3)** Remove current SQL Plan from Shared Pool

SQL> select inst\_id, 'exec DBMS\_SHARED\_POOL.PURGE('||''''||ADDRESS||','||HASH\_VALUE||''''||','||''''||'C'||''''||');' from GV$SQLAREA where sql\_id like 'g1xp7kq3g6qvg';

The output would be something like this;

INST\_ID SCRIPT

---------- -----------------------------------------------------------------------------

4 exec DBMS\_SHARED\_POOL.PURGE('00000008450E8BF0,804557861','C');

3 exec DBMS\_SHARED\_POOL.PURGE('0000000844D60E58,804557861','C');

1 exec DBMS\_SHARED\_POOL.PURGE('000000084EF1F588,804557861','C');

Execute above query on **respective nodes** to remove the SQL Plan from library cache.

**Step 4)** From 2nd step it’s generated the coe\_xfr\_sql\_profile\_g1xp7kq3g6qvg \_203366862.sql and asking us run to pin the plan. Once you run this script, the plan will be pin to the optimizer and it will pick the forced plan from the next run.

**SQL>@** **coe\_xfr\_sql\_profile\_g1xp7kq3g6qvg \_203366862.sql**

SQL>REM

SQL>REM $Header: 215187.1 coe\_xfr\_sql\_profile\_g1xp7kq3g6qvg \_203366862.sql 11.4.1.4 2017/05/03 csierra $

SQL>REM

SQL>REM Copyright (c) 2000-2010, Oracle Corporation. All rights reserved.

SQL>REM

SQL>REM AUTHOR

SQL>REM carlos.sierra@oracle.com

SQL>REM

SQL>REM SCRIPT

SQL>REM coe\_xfr\_sql\_profile\_cxahmavvzakqs\_291457268.sql

SQL>REM

SQL>REM DESCRIPTION

SQL>REM This script is generated by coe\_xfr\_sql\_profile.sql

SQL>REM It contains the SQL\*Plus commands to create a custom

SQL>REM SQL Profile for SQL\_ID cxahmavvzakqs based on plan hash

SQL>REM value 291457268.

SQL>REM The custom SQL Profile to be created by this script

SQL>REM will affect plans for SQL commands with signature

SQL>REM matching the one for SQL Text below.

SQL>REM Review SQL Text and adjust accordingly.

SQL>REM

SQL>REM PARAMETERS

SQL>REM None.

SQL>REM

SQL>REM EXAMPLE

SQL>REM SQL> START coe\_xfr\_sql\_profile\_cxahmavvzakqs\_291457268.sql;

SQL>REM

SQL>REM NOTES

SQL>REM 1. Should be run as SYSTEM or SYSDBA.

SQL>REM 2. User must have CREATE ANY SQL PROFILE privilege.

SQL>REM 3. SOURCE and TARGET systems can be the same or similar.

SQL>REM 4. To drop this custom SQL Profile after it has been created:

SQL>REM EXEC DBMS\_SQLTUNE.DROP\_SQL\_PROFILE('coe\_cxahmavvzakqs\_291457268');

SQL>REM 5. Be aware that using DBMS\_SQLTUNE requires a license

SQL>REM for the Oracle Tuning Pack.

SQL>REM

SQL>WHENEVER SQLERROR EXIT SQL.SQLCODE;

SQL>REM

SQL>VAR signature NUMBER;

SQL>REM

SQL>DECLARE

2 sql\_txt CLOB;

3 h SYS.SQLPROF\_ATTR;

4 BEGIN

5 sql\_txt := q'[

6 SELECT /\*+ NO\_CPU\_COSTING \*/ NVL(B.CUSIP\_C, B.ISIN\_C), DECODE(B.CUSIP\_C, NULL, 'I-', DECODE(SUBSTR(B.CUSIP\_C,1,1),'0','US','1','US','2','US','3','US','4','US','5','US','6','US','7','US','8','US','9','US','U\*')), TRUNC(A.BID\_PRICE,3), ROUND(A.BID\_YIELD,3), TRUNC(A.MEAN\_PRICE,3), ROUND(A.MEAN\_YIELD,3), TRUNC(A.ASK\_PRICE,3), ROUND(A.ASK\_YIELD,3), DECODE(A.PRICE\_FLAG,'D',TRUNC(A.BID\_PRICE,3),NULL), A.LAST\_PRICED\_DATE, DECODE(A.BID\_PRICING\_FEATURE,'P','U','A',' ',A.BID\_PRICING\_FEATURE), DECODE(A.MEAN\_PRICING\_FEATURE,'P','U','A',' ',A.MEAN\_PRICING\_FEATURE), DECODE(A.ASK\_PRICING\_FEATURE,'P','U','A',' ',A.ASK\_PRICING\_FEATURE), (SELECT E.ACC\_ACCRUED\_INT\_100 FROM

7 ACC\_INT\_STAGING E WHERE E.CUSIP = A.IDENTIFIER\_VALUE AND E.ADDED\_DATE = (SELECT MAX(H.ADDED\_DATE) FROM ACC\_INT\_STAGING H WHERE H.CUSIP = E.CUSIP)), A.BOND\_CURRENCY, A.CONVEXITY, A.DURATION, B.ISSUE\_DESCRIPTION\_T, B.CUSIP\_C, NULL, B.ISIN\_C, B.COUNTRY\_C, B.DEFAULT\_CODE\_C, B.MOODYS\_RATING2\_C, B.S\_P\_RATINGS2\_C, B.FITCH\_RATING\_C, B.CALL\_DATE\_D, B.CALL\_PRICE\_L, B.PUT\_DATE\_D, B.PUT\_PRICE\_L, B.INTEREST\_TYPE\_C, TO\_CHAR(B.COUPON\_L), B.DATED\_DATE\_D, B.ISSUER\_DESCRIPTION\_T, B.MATURITY\_D, DECODE(B.SECURITY\_TYPE,'CHILD','CHLD',SUBSTR(B.SECURITY\_TYPE,1,4)), NVL((SELECT S.ASSET\_CLASS FROM BOB\_ASSET\_TYPE\_ASIA\_CORP S WHERE S.CORPEVALS\_SUBS = A.SUBSET\_NUMBER AND FILE\_NAME LIKE

8 '%asia%'),NVL((SELECT T.ASSET\_CLASS FROM BOB\_ASSET\_TYPE\_EMEA\_CORP T WHERE T.CORPEVALS\_SUBS = A.SUBSET\_NUMBER),'CORP')) FROM BOB\_CORP\_INT\_PRC A, CORP\_INSTR\_STAGING B WHERE (B.CUSIP\_C IS NOT NULL OR B.ISIN\_C IS NOT NULL) AND (A.IDENTIFIER\_VALUE = B.SEDOL\_C OR ( B.SEDOL\_C IS NULL AND A.IDENTIFIER\_VALUE = B.FTEX\_C )) AND A.LAST\_PRICED\_DATE = TO\_DATE(:B3 ,'YYYYMMDD') AND (A.FILE\_NAME NOT LIKE '%latam%' OR A.SUBSET\_NUMBER <> 11) AND A.ROWID = (SELECT MAX(C.ROWID) FROM BOB\_CORP\_INT\_PRC C WHERE C.IDENTIFIER\_VALUE = A.IDENTIFIER\_VALUE AND C.LAST\_PRICED\_DATE = TO\_DATE(:B3 ,'YYYYMMDD') AND (C.FILE\_NAME NOT LIKE '%latam%' OR C.SUBSET\_NUMBER <> 11)) AND ( B.STATUS\_CODE

9 != 'M' OR ( B.STATUS\_CODE = 'M' AND B.STATUS\_REASON != 'ISS\_ERR' AND B.STATUS\_REASON != 'DUP\_IIE')) AND B.INSTRUMENT\_ID = NVL( (SELECT MAX(D.INSTRUMENT\_ID) FROM CORP\_INSTR\_STAGING D WHERE (A.IDENTIFIER\_VALUE = D.SEDOL\_C OR (D.SEDOL\_C IS NULL AND A.IDENTIFIER\_VALUE = D.FTEX\_C)) AND D.STATUS\_CODE = 'Lv'), NVL((SELECT MAX(D.INSTRUMENT\_ID) FROM CORP\_INSTR\_STAGING D WHERE (A.IDENTIFIER\_VALUE = D.SEDOL\_C OR (D.SEDOL\_C IS NULL AND A.IDENTIFIER\_VALUE = D.FTEX\_C)) AND D.STATUS\_CODE = 'PR'), NVL((SELECT MAX(D.INSTRUMENT\_ID) FROM CORP\_INSTR\_STAGING D WHERE (A.IDENTIFIER\_VALUE = D.SEDOL\_C OR (D.SEDOL\_C IS NULL AND A.IDENTIFIER\_VALUE = D.FTEX\_C)) AND D.STATUS\_CODE = 'M'

10 AND D.STATUS\_REASON != 'ISS\_ERR' AND D.STATUS\_REASON != 'DUP\_IEE'), (SELECT MAX(D.INSTRUMENT\_ID) FROM CORP\_INSTR\_STAGING D WHERE (A.IDENTIFIER\_VALUE = D.SEDOL\_C OR (D.SEDOL\_C IS NULL AND A.IDENTIFIER\_VALUE = D.FTEX\_C)) AND D.STATUS\_CODE NOT IN ('Lv','PR','M'))))) AND A.IDENTIFIER\_VALUE BETWEEN :B2 AND :B1 AND 0 = (SELECT COUNT(\*) FROM FIXED\_INCOME.DEBT\_MASTER\_M G WHERE G.PRICE\_DATE\_D = TO\_DATE(:B3 ,'YYYYMMDD') AND G.IDENTIFIER\_C = NVL(B.CUSIP\_C, B.ISIN\_C))

11 ]';

12 h := SYS.SQLPROF\_ATTR(

13 q'[BEGIN\_OUTLINE\_DATA]',

14 q'[IGNORE\_OPTIM\_EMBEDDED\_HINTS]',

15 q'[OPTIMIZER\_FEATURES\_ENABLE('11.2.0.3')]',

16 q'[DB\_VERSION('11.2.0.3')]',

17 q'[ALL\_ROWS]',

18 q'[OUTLINE\_LEAF(@"SEL$291F8F59")]',

19 q'[OUTLINE\_LEAF(@"SEL$B29E968D")]',

20 q'[UNNEST(@"SEL$3")]',

21 q'[OUTLINE\_LEAF(@"SEL$4")]',

22 q'[OUTLINE\_LEAF(@"SEL$5")]',

23 q'[OUTLINE\_LEAF(@"SEL$7")]',

24 q'[OUTLINE\_LEAF(@"SEL$8")]',

25 q'[OUTLINE\_LEAF(@"SEL$9")]',

26 q'[OUTLINE\_LEAF(@"SEL$10")]',

27 q'[OUTLINE\_LEAF(@"SEL$13AC9F9F")]',

28 q'[PUSH\_PRED(@"SEL$DC7577F0" "VW\_SQ\_2"@"SEL$74927A26" 6)]',

29 q'[OUTLINE\_LEAF(@"SEL$DC7577F0")]',

30 q'[UNNEST(@"SEL$11")]',

31 q'[UNNEST(@"SEL$6")]',

32 q'[OUTLINE(@"SEL$3")]',

33 q'[OUTLINE(@"SEL$E9784550")]',

34 q'[OUTLINE(@"SEL$CE1D94FA")]',

35 q'[OUTLINE(@"SEL$DC7577F0")]',

36 q'[UNNEST(@"SEL$11")]',

37 q'[UNNEST(@"SEL$6")]',

38 q'[OUTLINE(@"SEL$74927A26")]',

39 q'[OUTLINE(@"SEL$11")]',

40 q'[OUTLINE(@"SEL$6")]',

41 q'[OUTLINE(@"SEL$2")]',

42 q'[OUTLINE(@"SEL$1")]',

43 q'[FULL(@"SEL$DC7577F0" "A"@"SEL$1")]',

44 q'[INDEX\_RS\_ASC(@"SEL$DC7577F0" "B"@"SEL$1" ("CORP\_INSTR\_STAGING"."INSTRUMENT\_ID"))]',

45 q'[NO\_ACCESS(@"SEL$DC7577F0" "VW\_SQ\_2"@"SEL$74927A26")]',

46 q'[INDEX\_SS(@"SEL$DC7577F0" "G"@"SEL$11" ("DEBT\_MASTER\_M"."PRICE\_DATE\_D" "DEBT\_MASTER\_M"."SOURCE\_C" "DEBT\_MASTER\_M"."IDENTIFIER\_TYPE\_C" "DEBT\_MASTER\_M"."IDENTIFIER\_C"))]',

47 q'[LEADING(@"SEL$DC7577F0" "A"@"SEL$1" "B"@"SEL$1" "VW\_SQ\_2"@"SEL$74927A26" "G"@"SEL$11")]',

48 q'[USE\_NL(@"SEL$DC7577F0" "B"@"SEL$1")]',

49 q'[USE\_NL(@"SEL$DC7577F0" "VW\_SQ\_2"@"SEL$74927A26")]',

50 q'[USE\_NL(@"SEL$DC7577F0" "G"@"SEL$11")]',

51 q'[PUSH\_SUBQ(@"SEL$10")]',

52 q'[PUSH\_SUBQ(@"SEL$9")]',

53 q'[PUSH\_SUBQ(@"SEL$8")]',

54 q'[PUSH\_SUBQ(@"SEL$7")]',

55 q'[INDEX\_RS\_ASC(@"SEL$13AC9F9F" "C"@"SEL$6" ("BOB\_CORP\_INT\_PRC"."IDENTIFIER\_VALUE" "BOB\_CORP\_INT\_PRC"."IDENTIFIER\_TYPE" "BOB\_CORP\_INT\_PRC"."FILE\_NAME"))]',

56 q'[BITMAP\_TREE(@"SEL$10" "D"@"SEL$10" OR(1 1 ("CORP\_INSTR\_STAGING"."SEDOL\_C") 2 ("CORP\_INSTR\_STAGING"."FTEX\_C")))]',

57 q'[BITMAP\_TREE(@"SEL$9" "D"@"SEL$9" OR(1 1 ("CORP\_INSTR\_STAGING"."SEDOL\_C") 2 ("CORP\_INSTR\_STAGING"."FTEX\_C")))]',

58 q'[BITMAP\_TREE(@"SEL$8" "D"@"SEL$8" OR(1 1 ("CORP\_INSTR\_STAGING"."SEDOL\_C") 2 ("CORP\_INSTR\_STAGING"."FTEX\_C")))]',

59 q'[BITMAP\_TREE(@"SEL$7" "D"@"SEL$7" OR(1 1 ("CORP\_INSTR\_STAGING"."SEDOL\_C") 2 ("CORP\_INSTR\_STAGING"."FTEX\_C")))]',

60 q'[FULL(@"SEL$5" "T"@"SEL$5")]',

61 q'[FULL(@"SEL$4" "S"@"SEL$4")]',

62 q'[NO\_ACCESS(@"SEL$B29E968D" "VW\_SQ\_1"@"SEL$E9784550")]',

63 q'[INDEX(@"SEL$B29E968D" "E"@"SEL$2" ("ACC\_INT\_STAGING"."CUSIP"))]',

64 q'[LEADING(@"SEL$B29E968D" "VW\_SQ\_1"@"SEL$E9784550" "E"@"SEL$2")]',

65 q'[USE\_NL(@"SEL$B29E968D" "E"@"SEL$2")]',

66 q'[NLJ\_BATCHING(@"SEL$B29E968D" "E"@"SEL$2")]',

67 q'[INDEX\_RS\_ASC(@"SEL$291F8F59" "H"@"SEL$3" ("ACC\_INT\_STAGING"."CUSIP"))]',

68 q'[USE\_HASH\_AGGREGATION(@"SEL$291F8F59")]',

69 q'[END\_OUTLINE\_DATA]');

70 :signature := DBMS\_SQLTUNE.SQLTEXT\_TO\_SIGNATURE(sql\_txt);

71 DBMS\_SQLTUNE.IMPORT\_SQL\_PROFILE (

72 sql\_text => sql\_txt,

73 profile => h,

74 name => 'coe\_cxahmavvzakqs\_291457268',

75 description => 'coe cxahmavvzakqs 291457268 '||:signature||'',

76 category => 'DEFAULT',

77 validate => TRUE,

78 replace => TRUE,

79 force\_match => FALSE /\* TRUE:FORCE (match even when different literals in SQL). FALSE:EXACT (similar to CURSOR\_SHARING) \*/ );

80 END;

81 /

PL/SQL procedure successfully completed.

SQL>WHENEVER SQLERROR CONTINUE

SQL>SET ECHO OFF;

SIGNATURE

---------------------

2373650410408505516

... manual custom SQL Profile has been created

COE\_XFR\_SQL\_PROFILE\_cxahmavvzakqs\_291457268 completed

**Step 5)** to verify that the Profile got created:-

SQL> select name, category, created, status, description, sql\_text from dba\_sql\_profiles where name='coe\_cxahmavvzakqs\_291457268';

Predicate “name” was taken from previous script line number 74 in step 4.