SQL Gone Bad – But Plan Not Changed!

Kerry Osborne

Enkitec

Hotsos 2014

About Me

- Supporting Oracle since V2 (1982)
- “I’ve been working with Oracle since before you were born”
- Working with Exadata since V2 (2010)
- Co-author of Expert Oracle Exadata & Pro Oracle SQL
- I Work for Enkitec
- 11th talk at Hotsos ☺

ORACLE Platinum Partner

enkitec

Obligatory Marketing Slide

- E4 - 2 day Exadata conference in Dallas
  - June 2 – 3, 2014
  - http://www.enkitec.com/e4

- Tom Kyte
- Maria Colgan
- Tanel Poder
- Doug Burns
- Sue Lee
- Alex Gorbachev

ORACLE Platinum Partner

enkitec

Good News / Bad News

That which doesn't kill us makes us stronger.

What's the Point?

- Plan Stability Problems R a Bee
- A Small(?) Subset of Them are Very Difficult
  - Where the Plan Hash Value (PHV) is Unchanged
  - What???
  - How Can the Plan Change if the PHV Doesn't?

A Bit of History - Plan Stability

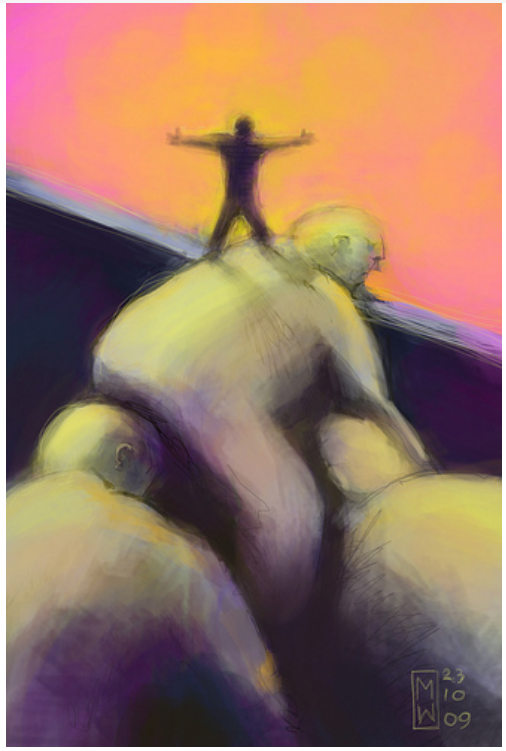
- One of the hardest issues to deal with
- CBO was released in V7
- Hints showed up in 8i
- First Plan Stability Feature (Outlines) in 8i
- SQL Profiles in 10g
- Baselines in 11g

Plan Stability – Automagic Troublemakers

- Bind Variable Peeking
- Cardinality Feedback
- Dynamic Sampling
- Adaptive Cursor Sharing
- Statistics Feedback
- Plan Directives
- Adaptive Optimization
- . . .

Credit Where Credit is Due!

- Randolf Geist
- Carlos Sierra



"If I have seen further it is by standing on the shoulders of Giants." ~ I. Newton

How Plan Hash Values are Calculated

- Randolf Geist

So in summary the following conclusions can be made:

- The same PLAN_HASH_VALUE is merely an indicator that the same operations on objects of the same name are performed in the same order.

- It tells nothing about the similarity of the expected runtime performance of the execution plan, due to various reasons as demonstrated. The most significant information that is not covered by the PLAN_HASH_VALUE are the filter and access predicates, but there are other attributes, too, that are not part of the hash value calculation.

http://oracle-randolf.blogspot.com/2009/07/planhashvalue-how-equal-and-stable-are.html

Missing From PHV Calculation

- Predicates
- Object Ids
- Partition Info (PSTART and PSTOP)
- Rows, Bytes, Cost, Time
- Number of Parallel Slaves*

* Not included in v$sql_plan at all (except in other_xml when AUTO DOP kicks in)

PHV Calculation – What's Included

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------------------------------------
SQL_ID  bxd77v75nynd8, child number 0
-------------------------------------
select /*+ parallel (a 4) */ avg(pk_col) from kso.skew a where col1 > 0

Plan hash value: 578366071

--------------------------------------------------------------------------------------------------------------------
| Id  | Operation                     | Name     | Rows  | By                                   T| PQ Distrib |
--------------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT              |          |       |       | 636   00)|              |      |            |
|   1 |  SORT AGGREGATE               |          |     1 |    11 |          |              |      |            |
|   2 |   PX COORDINATOR              |          |       |       |          |              |      |            |
|   3 |    PX SEND QC (RANDOM)        | :TQ10000 |     1 |    11 |          |        Q1,00 | P->S | QC (R   )  |
|   4 |     SORT AGGREGATE            |          |       |    11 |          |          ,00 | PCWP |            |
|   5 |      PX BLOCK ITERATOR        |          |   32M |       | 6363  (1)| 00:00:01 | Q1   |            |
|*  6 |       TABLE ACCESS STORAGE FULL| SKEW    |   32M |  335M |          |              |      |            |
--------------------------------------------------------------------------------------------------------------------

Predica       mation (identified by      on id):
---------------------------------------------------

   6 - storage(:Z>=:Z      Z<=:Z AND "COL1">0)
       filter("C      0)
```

ORACLE Platinum Partner

enkitec

Does it Matter?

```
SYS@db12c1> @fsx
Enter value for sql_text:
Enter value for sql_id: 5p4610dk709pj

SQL_ID         CHILD  PLAN_HASH   EXECS  AVG_ETIME  AVG_PX OFFLOAD IO_SAVED_%
------------- ------ ---------- ------ ---------- ------ ------- ----------
5p4610dk709pj      0  4220890033      1       4.39      0 Yes           74.96
5p4610dk709pj      1  4220890033      1     211.75      0 Yes           74.97

2 rows selected.
```

SQL_ID	CHILD	PLAN_HASH_VALUE	EXECS	ROWS	AVG_ETIME	AVG_CPU	AVG_PIO	AVG_LIO
5p4610dk709pj	0	4220890033	1	1	4.39	.71	326,073	326,177
5p4610dk709pj	1	4220890033	1	1	211.75	209.77	326,072	326,091

How would you diagnose this?

ORACLE Platinum Partner

enkitec

Does it Matter?

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------
SQL_ID  5p4610dk709pj, child number 0
-------------------------------------
select /*+ monitor */ avg(pk_col) from kso.skew2 where
translate(to_char(sin(col1)),'1','a') = 'a' and col4 = 'F'

Plan hash value: 4220890033

-----------------------------------------------------------------------------------------
| Id  | Operation                  | Name  | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT           |       |       |       | 46673 (100)|          |
|   1 |  SORT AGGREGATE            |       |     1 |    13 |            |          |
|*  2 |   TABLE ACCESS STORAGE FULL| SKEW2 |   213K|  2708K| 46673   (2)| 00:00:02 |
-----------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage(("COL4"='F' AND TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a'))
       filter(("COL4"='F' AND TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a'))
```

Does it Matter?

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------
SQL_ID  5p4610dk709pj, child number 1
-------------------------------------
select /*+ monitor */ avg(pk_col) from kso.skew2 where
translate(to_char(sin(col1)),'1','a') = 'a' and col4 = 'F'

Plan hash value: 4220890033

-------------------------------------------------------------------------------------
| Id  | Operation                   | Name  | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |       |       |       | 46673 (100)|          |
|   1 |  SORT AGGREGATE             |       |     1 |    13 |            |          |
|*  2 |   TABLE ACCESS STORAGE FULL| SKEW2 |   213K|  2708K| 46673   (2)| 00:00:02 |
-------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage((TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a' AND "COL4"='F'))
       filter((TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a' AND "COL4"='F'))
```

How Did This Example Happen?

```
SYS@db12c1> @sql_patch_hints
Enter value for patch_name: PATCH_5p4610dk709pj

HINT
--------------------------------------------------
ordered_predicates

1 rows selected.
```

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------
SQL_ID  5p4610dk709pj, child number 1
-------------------------------------
select /*+ monitor */ avg(pk_col) from kso.skew2 where
translate(to_char(sin(col1)),'1','a') = 'a' and col4 = 'F'

Plan hash value: 4220890033

-------------------------------------------------------------------------------------
| Id  | Operation                   | Name  | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |       |       |       | 46673 (100)|          |
|   1 |  SORT AGGREGATE             |       |     1 |    13 |            |          |
|*  2 |   TABLE ACCESS STORAGE FULL| SKEW2 |   213K|  2708K| 46673   (2)| 00:00:02 |
-------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage((TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a' AND "COL4"='F'))
       filter((TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a' AND "COL4"='F'))

Note
-----
   - SQL patch "PATCH_5p4610dk709pj" used for this statement
```

ORACLE Platinum Partner

enkitec

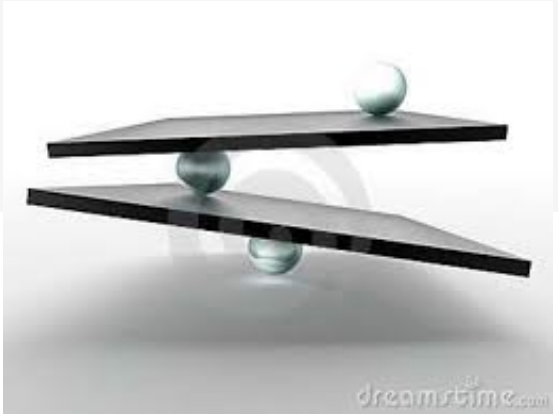
How Did This Example Happen?

```
SYS@db12c1> select count(*) from kso.skew2;

  COUNT(*)
----------
  64000008

1 row selected.

Elapsed: 00:00:04.53
SYS@db12c1> select count(*) from kso.skew2 where col4='F';

  COUNT(*)
----------
         2

1 row selected.

Elapsed: 00:00:03.87
```

Does It Happen in the Wild?

- Absolutely - why else would I be talking about it ☺
- Why Don't We See It More?
- Probably It's Partly Because We Weren't Looking For It
- The more critical / the more noticeable?

One Real Life Story



- Symptoms
  - Unstable Performance
  - No PHV Change
  - No Apparent Difference in Work Load
  - Predicates Were Changing
  - Turned Out That it was a Stats Issue
  - Partitioned Tables
  - Generally We'd Expect PHV Changes
  - But Instead, Predicates Changed

Identification

- So How Can We Verify This Is Our Problem?
- Easy - Find a Previous Version and Compare
  - Well That Sounds Pretty Easy!
  - But There can be 100's or 1000's of Predicates
  - So Visual Comparison can be Difficult at Best
  - But we Can Write SQL to Solve that Problem

How to Identify the Problem

- Carlos Sierra

Conclusion

When it comes to Execution Plans and their Plan Hash Value, it is possible that two Plans with same PHV are actually different if you consider the Predicates, and also possible you get a different PHV even if the Plan is the same if your Plan has system-generated names. So, during your analysis just looking at the PHV to determine if two Plans are the same or not is not enough. If you are already using SQLT, pay attention to the "Execution Plans" section, and to the SQLT PHV1 and PHV2 columns.

http://carlos-sierra.net/2013/06/09/has-my-plan-changed-or-not/

Demo Time

Possible Approaches - Identification

- Try Cursor Cache
  - Yes – But Only if Both Plans Exist Simultaneously
  - See my Blog Post on the References Slide
- Try AWR
  - Unfortunately AWR Doesn't Capture Predicates
  - See Bugs on References Slide
- Try Statspack
  - Also Doesn't Capture Predicates
  - But the code is available, so you could change it
  - See my Blog Post on the References Slide

Possible Approaches - Identification

- Try SQL Monitor
  - Sadly, SQL Monitor Doesn't Capture Predicates Either
- Try SPM
  - In 11g this won't work either
  - But in 12c, the Plan is Actually Recorded
  - Including the Predicates – Yay!
  - Use dbms_xplan.display_sql_plan_baseline
  - Unfortunately only 1 set per PHV

Digression - Baselines

- Based on Hints
- Attempts to Recreate a PHV
- If PHV Not Matched, Throws Out Hints
- If PHV Matched, It's Good to Go
- 11g Doesn't Store Plan At All
- 12c Stores Plan so DBMS_XPLAN Can Show Plan

12c Baselines Example

```
SQL_ID        CHILD PLAN_HASH_VALUE      EXECS       ROWS  AVG_ETIME    AVG_CPU    AVG_PIO        AVG_LIO
------------- ------ --------------- ---------- ---------- ---------- ---------- ---------- --------------
5p4610dk709pj      0      4220890033          1          1       4.39        .71    326,073        326,177
5p4610dk709pj      1      4220890033          1          1     211.75     209.77    326,072        326,091
```

```
SYS@db12c1> @create_baseline
Enter value for sql_id: 5p4610dk709pj
Enter value for plan_hash_value: 4220890033
Enter value for fixed (NO):
Enter value for enabled (YES):
Enter value for plan_name (ID_sqlid_planhashvalue): 5p4610dk709pj_fast

Baseline 5P4610DK709PJ_FAST created.

Elapsed: 00:00:02.16
```

* Note: create_baseline.sql uses dbms_spm.load_plans_from_cursor_cache
* Note 2: child 1 was flushed before creating the baseline

12c Baselines Example

```
Elapsed: 00:00:04.03

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------
SQL_ID  5p4610dk709pj, child number 1
-------------------------------------
select /*+ monitor */ avg(pk_col) from kso.skew2 where
translate(to_char(sin(col1)),'1','a') = 'a' and col4 = 'F'

Plan hash value: 4220890033

-----------------------------------------------------------------------------------------
| Id  | Operation                   | Name  | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT            |       |       |       | 46344 (100)|          |
|   1 |  SORT AGGREGATE             |       |     1 |    13 |            |          |
|*  2 |   TABLE ACCESS STORAGE FULL| SKEW2 |     1 |    13 | 46344   (2)| 00:00:02 |
-----------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage(("COL4"='F' AND TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a'))
       filter(("COL4"='F' AND TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a'))

Note
-----
   - SQL plan baseline 5p4610dk709pj_fast used for this statement
```

ORACLE Platinum Partner

enkitec

12c Baselines Example

```
SYS@db12c1> alter session set "_optimizer_cost_model" = IO;

Session altered.

Elapsed: 00:00:00.00
SYS@db12c1> @xyz6

AVG(PK_COL)
-----------


1 row selected.

Elapsed: 00:03:52.51
```

12c Baselines Example

```
PLAN_TABLE_OUTPUT
------------------------------------------------------------------------
SQL_ID  5p4610dk709pj, child number 0
-------------------------------------
select /*+ monitor */ avg(pk_col) from kso.skew2 where
translate(to_char(sin(col1)),'1','a') = 'a' and col4 = 'F'

Plan hash value: 4220890033

-----------------------------------------------------------------------
| Id  | Operation                  | Name  | Rows  | Bytes | Cost  |
-----------------------------------------------------------------------
|   0 | SELECT STATEMENT           |       |       |       | 49684 |
|   1 |  SORT AGGREGATE            |       |     1 |    13 |       |
|*  2 |   TABLE ACCESS STORAGE FULL| SKEW2 |     1 |    13 | 49684 |
-----------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage((TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a' AND
               "COL4"='F'))
       filter((TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a' AND
               "COL4"='F'))

Note
-----
   - cpu costing is off (consider enabling it)
   - SQL plan baseline 5p4610dk709pj_fast used for this statement
```

ORACLE Platinum Partner

enkitec

Good News
12c Baselines Save Predicates!

```
SYS@db12c1> select * from table(dbms_xplan.display_sql_plan_baseline('&sql_handle','&plan_name','all'));
Enter value for sql_handle:
Enter value for plan_name: 5p4610dk709pj_fast

PLAN_TABLE_OUTPUT
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
SQL handle: SQL_0305192ef56bf7d0
SQL text: select /*+ monitor */ avg(pk_col) from kso.skew2 where
          translate(to_char(sin(col1)),'1','a') = 'a' and col4 = 'F'
--------------------------------------------------------------------------------

--------------------------------------------------------------------------------
Plan name: 5p4610dk709pj_fast         Plan id: 1438813450
Enabled: YES     Fixed: NO      Accepted: YES     Origin: MANUAL-LOAD
Plan rows: From dictionary
--------------------------------------------------------------------------------

Plan hash value: 4220890033

-------------------------------------------------------------------------------------
| Id  | Operation                  | Name  | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT           |       |       |       | 46344 (100)|          |
|   1 |  SORT AGGREGATE            |       |     1 |    13 |     0   (0)|          |
|*  2 |   TABLE ACCESS STORAGE FULL| SKEW2 |     1 |    13 | 46344   (2)| 00:00:02 |
-------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - storage(("COL4"='F' AND TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a'))
       filter(("COL4"='F' AND TRANSLATE(TO_CHAR(SIN("COL1")),'1','a')='a'))
```

enkitec

That's a Wrap

- Plans Can Change While PHV Remains Constant
- Predicates Are Not Captured
  - Except by Baselines in 12c
  - AFAIK
- So Even Identification Can Be Difficult
- Saving Predicate Info Yourself May Be Necessary
- Profiles and Baselines Will Probably Not Help
- Although 12c SQL Translation Framework Might
- Rewriting SQL May Be Required

References

- http://oracle-randolf.blogspot.com/2009/07/planhashvalue-how-equal-and-stable-are.html
- http://oracle-randolf.blogspot.com/2009/07/planhashvalue-how-equal-and-stable-are_26.html
- http://carlos-sierra.net/2013/06/09/has-my-plan-changed-or-not/
- http://kerryosborne.oracle-guy.com/2013/06/sql-gone-bad-but-plan-not-changed/
- http://kerryosborne.oracle-guy.com/2013/06/sql-gone-bad-but-plan-not-changed-part-2/
- http://kerryosborne.oracle-guy.com/2013/07/sql-translation-framework/
- Bug 5217053 : IN DBA_HIST_SQL_PLAN SOME FIELDS ARE NOT GETTING POPULATED
- Bug 7493519 : ACCESS AND FILTER PREDICATES MISSING IN DBA_SQL_PLAN_HIST
- Bug 5683955 : ACCESS AND FILTER PREDICATE NOT POPULATED STATS$SQL_PLAN

Questions

Email: kerry.osborne@enkitec.com
Blog: kerryosborne.oracle-guy.com
Twitter: @kerryoracleguy

ORACLE Platinum Partner

enkitec