Dissecting SQL Plan Management Options
A SQL Troubleshooting, Stability
and
Performance Talk

Faisal Ijaz
Fidelity International
United Kingdom

Keywords:

Oracle Database Performance, SQL Plan Management, SQL Profiles, Hints, Execution Plans

Introduction

Application Performance and SQL Plan Stability have long been points of interest and discussion between Database Administrators and Application Developers. As Oracle continues to make Optimizer funkier and more intelligent, it poses an equally bigger challenge to maintain the performance of hybrid and ever changing workloads.

SQL Plan Stability In Perspective

Over the years there have been many different techniques through which SQL Plan Stability can be enforced either directly by emending hints into a SQL Statement or indirectly through various different tools or techniques provided by Oracle such as the use of Stored Outlines which was one of the first mechanisms to manage query plans or using more modern ways such as the use of SQL Plan Baselines.

These techniques help Databases Managers, Administrators and Developers in a number of ways:

- Prevents certain Database environment changes from affecting Application Query Performance
- Facilitates Database migrations and Version upgrades
- Help reduce risks from performance changes to critical business operations
- Has been around for a while and evolved over time
- Needed more and more as the Optimizer becomes ever more intelligent
- Query Execution Plan change for a variety of Reasons
  - Optimizer Statistics and Optimizer Parameter
  - Hybrid Workload types
  - Database Upgrades
  - Code or Structural Changes

APIs for SQL PLAN Stability

Oracle provides for a Command Line Interface (CLI) and in some cases a Graphical Unser Interface (GUI) for all the supported SQL Plan Stability options.

The APIs greatly simplify the use of these stability options and allow for their better understanding. In most cases these APIs are well documented however there can be some undocumented features as well.

- Oracle Provides API for SPM Options
  - DBMS_SPM for SQL Baselines
  - DBMS_SQLTUNE for SQL Profiles
  - DBMS_OUTLN for Stored Outlines
  - DBMS_SQLDIAG for SQL Patches
  - Hints

Enterprise Manager and SQL Developer are a couple of Oracle offerings that provide GUI to some the above mentioned APIs.

Stored Outlines

Stored outlines was one of the first SQL Plan Stability options which Oracle provided. As the image below suggests, Stored Outlines offered a way of getting a group of Hints attached to a SQL Statement. It was a relatively simple yet effective way of getting the optimizer to use the hints with the aim of generating a particular Execution Plan.

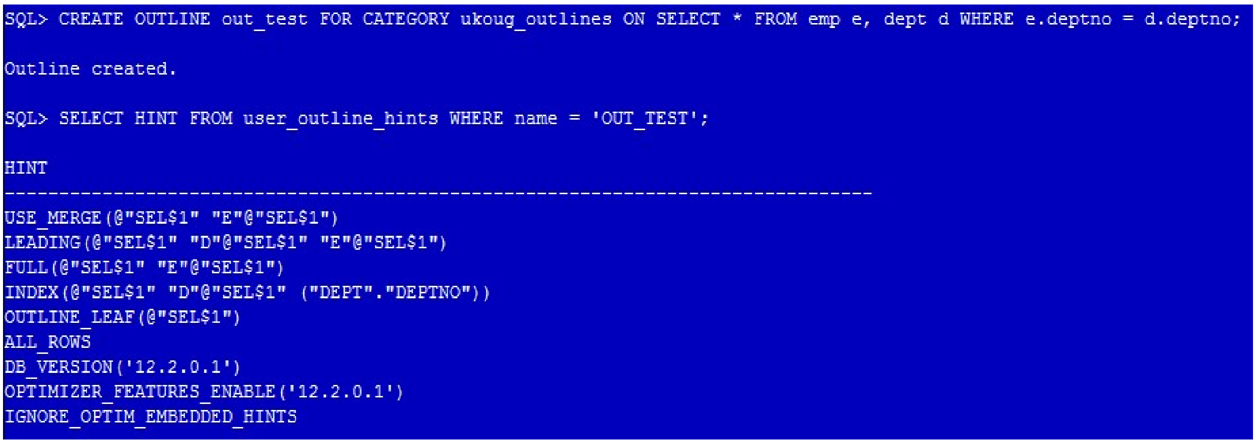
Although it is now a Deprecated feature, the API is still available in Oracle12C.

The Data dictionary views below provide more information about Stored Outlines that may exist in an Oracle Database.

- Outline information OL$ View (OL$, OL$HINTS etc
- Source DBA|ALL|USER_OUTLINE_HINTS

The image below shows how a Stored Outline can be created and how the hints which are embedded in the Stored Outline can be viewed.

Stored Outlines can be automatically created by setting an initialization parameter CREATE_STORED_OUTLINES.



```
SQL> CREATE OUTLINE out_test FOR CATEGORY ukoug_outlines ON SELECT * FROM emp e, dept d WHERE e.deptno = d.deptno;

Outline created.

SQL> SELECT HINT FROM user_outline_hints WHERE name = 'OUT_TEST';

HINT
--------------------------------------------------------------------------------
USE_MERGE(@"SEL$1" "E"@"SEL$1")
LEADING(@"SEL$1" "D"@"SEL$1" "E"@"SEL$1")
FULL(@"SEL$1" "E"@"SEL$1")
INDEX(@"SEL$1" "D"@"SEL$1" ("DEPT"."DEPTNO"))
OUTLINE_LEAF(@"SEL$1")
ALL_ROWS
DB_VERSION('12.2.0.1')
OPTIMIZER_FEATURES_ENABLE('12.2.0.1')
IGNORE_OPTIM_EMBEDDED_HINTS
```

SQL Profiles

Another way of injecting hints in to a SQL Statement is through the use of SQL Profiles. SQL Profiles are actually a tool of injecting Cardinality Hints into a SQL to help the optimizer get more accurate Cardinality estimates and thus assisting the Optimizer in generating an optimized execution plan.

However since their inception, SQL Profiles have evolved as a mechanism of enforcing an intended execution plan.

More information about SQL Profiles can be found in views below.

- Source SQLOBJ$DATA & DBA_SQL_PROFILES

SQL Plan Baselines

SQL Plan Baselines or Baselines as they are more commonly known as, are the corner stone of Oracle's SQL Plan Management. SQL Plan Management was first introduced in Oracle11g and has evolved in Oracle 12c and 18.

There are various ways and means of getting Baselines in place and it is important to understand how more meaningful and relevant SQL Plans can be captured.

SQL Plan Baselines - Particulars

Some ways of Capturing SQL Plan Baselines is below.

- Capture Automatically if OPTIMZIER_SQL_PLAN_BASELINES is set to TRUE
- Capture Manually from
- Cursor Cache → DBMS_SPM.LOAD_PLANS_FROM_CURSOR_CACHE
- SQL Tuning Set → DBMS_SPM.LOAD_PLANS_FROM_SQLSET
- Baseline Staging → DBMS_SPM.UNPACK_STGTAB_BASELINE

- AWR → DBMS_SPM.LOAD_PLANS_FROM_AWR

There are lots of practical implications need to be considered as to how the SQL Baselines are captured. These include but are not limited to space, performance etc

Practical Consideration on Enabling SQL Plan Baseline Catpure

When enabling real time capture, can being some challenges around space and have an impact on parse times. This will be discussed in the presetation.

SQL Plan Baselines Enhancements in Oracle 12.2 or 18C

In Oracle 12.2 and in Oracle 18c there are further enhancement to SQL Plan Management which makes collecting Baselines more specific.

The session will cover the internals to all the above techniques, enhancements to these and how an Oracle Database professional can use them to exert more control over the SQL Execution Plans and use them as a tool to resolve active SQL Performance related problems.

Contact address:

Faisal Ijaz
Fidelity International
130 Oakhill House,
Tonbridge Road, Hildenborough
Kent, United Kingdom
TN11 9DZ

Phone:	+44(0) 7799868617
Fax:	+44 (N/A)
Email	faisal_ijaz@hotmail.com and ijazfais@gmail.com
Internet:	Twitter **_faisal_ijaz**