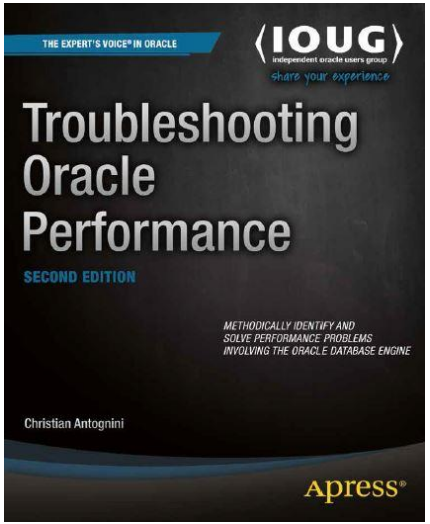
Adaptive Plans

Christian Antognini

@ChrisAntognini antognini.ch/blog

@ChrisAntognini



Senior principal consultant, trainer and partner at Trivadis

- christian.antognini@trivadis.com
- http://antognini.ch

Focus: get the most out of database engines

- Logical and physical database design
- Query optimizer
- Application performance management

Author of *Troubleshooting Oracle Performance* (Apress, 2008/14)

OakTable Network, Oracle ACE Director

trivadis
makes IT easier.

Adaptive Plans – Challenge

Object statistics don’t always provide sufficient information

To get additional insights, the query optimizer can use features like dynamic sampling and cardinality feedback

- They don’t solve all issues, though

Adaptive Plans – Concept

The query optimizer can postpone some decisions until the execution phase

The idea is to leverage information collected while executing part of an execution plan to determine how another part should be carried out

The query optimizer uses adaptive plans in three situations

- To switch the join method from a NL to a HJ and vice versa
- To switch the PX distribution method from hash to broadcast/round-robin
- To disable the access to a dimension for execution plans using the star transformation

Agenda

1. Join Method Switch
2. Star Transformation
3. Configuration
4. Dynamic Performance Views

Join Method Switch

Join Method Switch

The query optimizer adds *subplans* (one NL and one HJ) to execution plans

- One of the alternatives is the default plan

One of the subplans is chosen during the first execution

- The choice is based on the number of rows actually processed
- The query optimizer computes an inflection point

A new row source operation is used to partially buffer and count the rows

- STATISTICS COLLECTOR

The execution plan that is actually executed is called the final plan

Join Method Switch Example

```
SELECT * FROM t1, t2 WHERE t1.id = t2.id AND t1.n = 666

-----------------------------------------------
| Id  | Operation                     | Name  |
-----------------------------------------------
|   0 | SELECT STATEMENT              |       |
|   1 |  HASH JOIN                    |       |
|   2 |   NESTED LOOPS                |       |
|   3 |    NESTED LOOPS               |       |
|   4 |     STATISTICS COLLECTOR      |       |
|   5 |      TABLE ACCESS FULL        | T1    |
|   6 |     INDEX UNIQUE SCAN         | T2_PK |
|   7 |    TABLE ACCESS BY INDEX ROWID| T2    |
|   8 |   TABLE ACCESS FULL           | T2    |
-----------------------------------------------
```

Join Method Switch Example

```
SELECT * FROM t1, t2 WHERE t1.id = t2.id AND t1.n = 666

-----------------------------------------------
| Id  | Operation                     | Name  |
-----------------------------------------------
|   0 | SELECT STATEMENT              |       |
|   1 |  HASH JOIN                    |       |
|   2 |   NESTED LOOPS                |       |
|   3 |    NESTED LOOPS               |       |
|   4 |     STATISTICS COLLECTOR      |       |
|   5 |      TABLE ACCESS FULL        | T1    |
|   6 |     INDEX UNIQUE SCAN         | T2_PK |
|   7 |    TABLE ACCESS BY INDEX ROWID| T2    |
|   8 |   TABLE ACCESS FULL           | T2    |
-----------------------------------------------
```

Join Method Switch Example

```
SELECT * FROM t1, t2 WHERE t1.id = t2.id AND t1.n = 666

-----------------------------------------------
| Id  | Operation                     | Name  |
-----------------------------------------------
|   0 | SELECT STATEMENT              |       |
|   1 |  HASH JOIN                    |       |
|   2 |   NESTED LOOPS                |       |
|   3 |    NESTED LOOPS               |       |
|   4 |     STATISTICS COLLECTOR      |       |
|   5 |      TABLE ACCESS FULL        | T1    |
|   6 |     INDEX UNIQUE SCAN         | T2_PK |
|   7 |    TABLE ACCESS BY INDEX ROWID| T2    |
|   8 |   TABLE ACCESS FULL           | T2    |
-----------------------------------------------
```

Join Method Switch Inflection Point

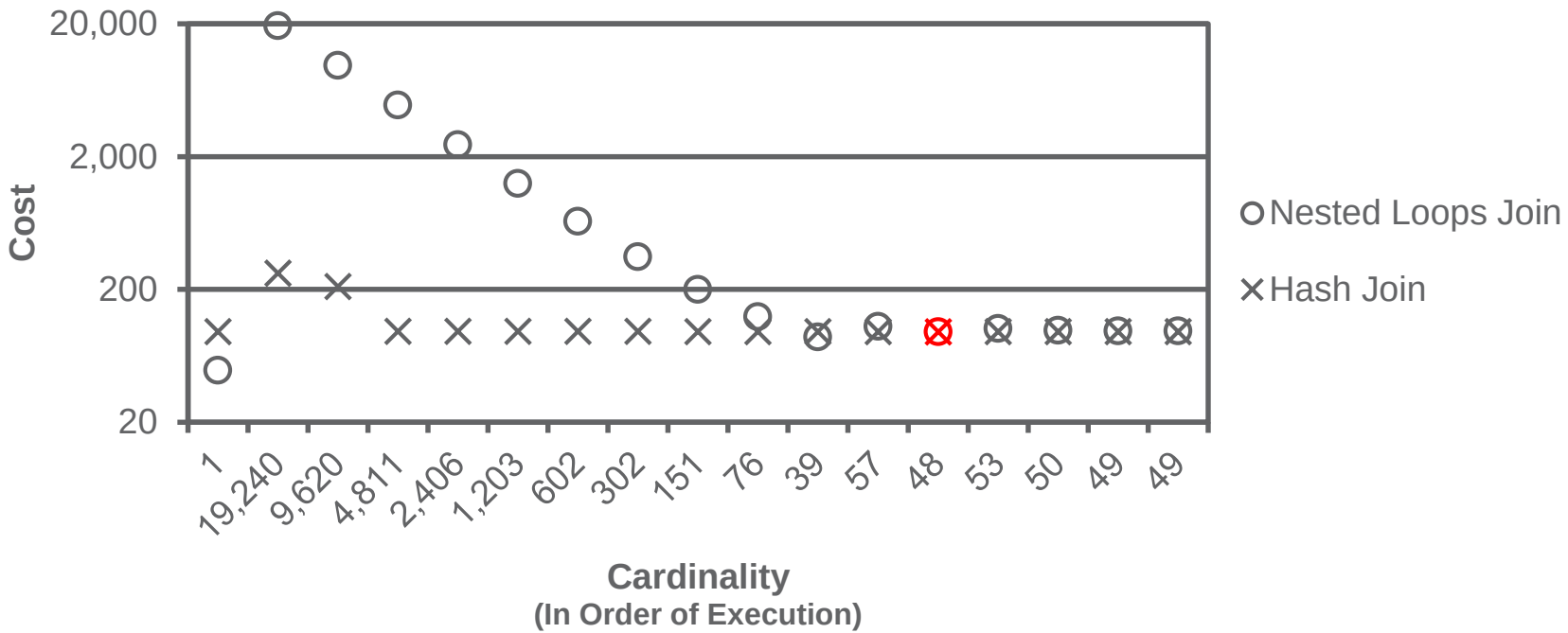
For both join methods, the cost associated to different cardinalities is estimated

- The cardinality of the outer table varies
- The cardinality of the inner table remains fixed

The query optimizer uses a binary search

The search takes place between a minimum and maximum cardinality

Join Method Switch Inflection Point Example



Join Method Switch Limitations

adaptive_plan_lob.sql
adaptive_plan_obj.sql

The amount of memory that is allocated by STATISTICS COLLECTOR is limited

- If a too large buffer is required, no adaptive plan is used

Partition-wise joins can’t be adaptive

STATISTICS COLLECTOR can’t return a LOB

An XMLTYPE or an object can’t be involved

Star Transformation

Star Transformation

With the star transformation, the data of each dimension that has a restriction applied to it might be “joined” to the corresponding bitmap index of the fact

If the number of rowids returned by such a “join” is underestimated, applying the filter can be detrimental to the performance

With an adaptive plan the access to some dimensions can be disabled

- Decision takes place during the first execution only

Star Transformation Example

```
-----------------------------------------------------
| Operation                          | Name           |
-----------------------------------------------------
| …                                  |                |
|    VIEW                            | VW_ST_5497B905 |
|     NESTED LOOPS                   |                |
|      BITMAP CONVERSION TO ROWIDS   |                |
|       BITMAP AND                   |                |
|        BITMAP MERGE                |                |
|         BITMAP KEY ITERATION       |                |
|          TABLE ACCESS FULL         | COLORS         |
|          BITMAP INDEX RANGE SCAN   | CAR_COLOR_IDX  |
|        STATISTICS COLLECTOR        |                |
|         BITMAP MERGE               |                |
|          BITMAP KEY ITERATION      |                |
|           TABLE ACCESS FULL        | MODELS         |
|           BITMAP INDEX RANGE SCAN  | CAR_MODEL_IDX  |
|         …                          |                |
|      TABLE ACCESS BY USER ROWID    | CARS           |
-----------------------------------------------------
```

Configuration

12.1 – OPTIMIZER_ADAPTIVE_FEATURES

Enables or disables adaptive query optimization features

- Adaptive plans
- SQL plan directives
- Automatic reoptimization (it isn't the case in 12.1.0.1; bug 16824474)

Dynamic statistics are controlled by OPTIMIZER_DYNAMIC_SAMPLING

The default value is TRUE

12.2 – OPTIMIZER_ADAPTIVE_PLANS

Enables or disables adaptive plans

The default value is TRUE

12.2 – OPTIMIZER_ADAPTIVE_STATISTICS

Enables or disables adaptive statistics

- SQL plan directives
  - The creation is always enabled, only their use is managed
- Performance feedback
- Statistics feedback
  - The functionality of 11.2 is always enabled

Dynamic statistics are controlled by OPTIMIZER_DYNAMIC_SAMPLING

The default value is FALSE

Backport of 12.2 Configuration in 12.1.0.2

Patch to backport the 12.2 initialization parameters to 12.1.0.2:

22652097: PROVIDE SEPARATE CONTROLS FOR ADAPTIVE PLANS AND ADAPTIVE STATISTICS FEATURES

When installed, OPTIMIZER_ADAPTIVE_FEATURES can no longer be set

- Patch 22652097 is included in PBP Oct 2017
- By default it's disabled, refer to MOS note 2312911.1 for information

Common Configurations in 12.2 or Patched 12.1

Minimal Adaptability (11.2 Default)

OPTIMIZER_ADAPTIVE_PLANS = FALSE
OPTIMIZER_ADAPTIVE_STATISTICS = FALSE
AUTO_STAT_EXTENSIONS = OFF

Medium Adaptability (12.2 Default)

OPTIMIZER_ADAPTIVE_PLANS = TRUE
OPTIMIZER_ADAPTIVE_STATISTICS = FALSE
AUTO_STAT_EXTENSIONS = OFF

Significant Adaptability

OPTIMIZER_ADAPTIVE_PLANS = TRUE
OPTIMIZER_ADAPTIVE_STATISTICS = TRUE
AUTO_STAT_EXTENSIONS = OFF

Maximum Adaptability (12.1 Default)

OPTIMIZER_ADAPTIVE_PLANS = TRUE
OPTIMIZER_ADAPTIVE_STATISTICS = TRUE
AUTO_STAT_EXTENSIONS = ON

Reporting Mode

It’s useful to assess how an execution plan would change if adaptive plans are activated

If enabled, the query optimizer generates adaptive plans but the execution engine only use the default plan and checks whether it would “switch”

OPTIMIZER_ADAPTIVE_REPORTING_ONLY controls whether it’s enabled

- FALSE (default) disables it
- TRUE enables it for the adaptive features that are enabled

Reporting Mode – DBMS_XPLAN

Use DBMS_XPLAN to get information about the “analysis”

```
SELECT *
FROM table(dbms_xplan.display_cursor(format=>'report'))
```

In 12.1.0.1 might fail with an ORA-1001 (bug 17270605)

Reporting Mode – How to List the Cursors that Would Be Impacted?

There is no trivial way to find them ☹ Here's a query that does that:

```
SELECT sql_id, child_number
FROM v$sql_plan p
WHERE other_xml IS NOT NULL
AND (sql_id, child_number) IN (SELECT sql_id, child_number
                               FROM v$sql
                               WHERE is_resolved_adaptive_plan IS NOT NULL)
AND EXISTS (SELECT 1
            FROM XMLTable('/other_xml/display_map/row' PASSING XMLType(p.other_xml)
                          COLUMNS skp INTEGER PATH '@skp', op INTEGER PATH '@op') x,
                 XMLTable('/other_xml/display_map/report_display_map/row'
                          PASSING XMLType(other_xml)
                          COLUMNS skp INTEGER PATH '@skp', op INTEGER PATH '@op') r
            WHERE x.op = r.op
            AND x.skp <> r.skp)
```

Hints

For join method switches and star transformation, as of 12.1.0.2 two hints are available:

- ADAPTIVE_PLAN
- NO_ADAPTIVE_PLAN

Dynamic Performance Views

V$SQL.IS_RESOLVED_ADAPTIVE_PLAN

New column set for join method switches and star transformation only

- NULL: the execution plan associated to the cursor isn't adaptive
- N: the final execution plan not yet determined
- Y: the final execution plan was determined
  - Also set if reporting mode is enabled

V$ACTIVE_SESSION_HISTORY. SQL_ADAPTIVE_PLAN_RESOLVED

Available as of 12.1.0.2

Don’t rely on the provided value (bug?)

Summary

- Some decisions are postponed during the execution
- The query optimizer is getting more and more dynamic

Questions and Answers

Christian Antognini
Senior Principal Consultant

christian.antognini@trivadis.com

@ChrisAntognini

trivadis
makes IT easier.