Informatik Aktuell

www.informatik-aktuell.de

Alkmene Verlags- und
Mediengesellschaft mbh
Wittelsbacherallee 107
60385 Frankfurt am Main

IT-Tage 2015

Schwerpunkt: Datenbanken
14.12. - 18.12.2015
Frankfurt am Main

Udo Fohrmann:

Adaptive Dynamic Sampling in Oracle 12c

Adaptive Dynamic Sampling

IT-Tage Datenbanken in Frankfurt, 17. Dezember 2015

Christian Antognini
Udo Fohrmann

About me

With Trivadis since November 2008

- Senior Consultant
- Udo.fohrmann@trivadis.com

Main Focus on database performance tuning

- Instructor for Oracle Tuning and Data modeling
- Consulting in SQL Tuning, performance troubleshooting

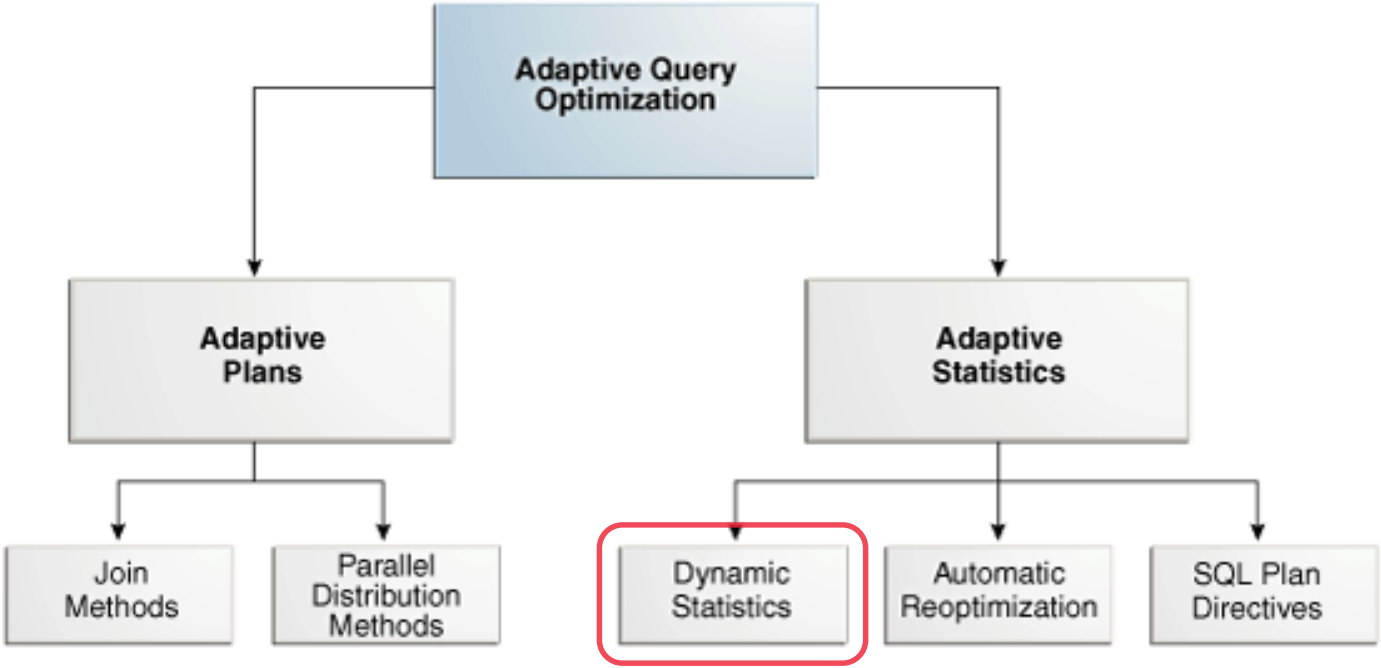
Over 25 years experience in using Oracle products

Agenda

1. Introduction
2. When It Is Used?
3. Time Limit
4. Strategies
5. Caching
6. Fallacies
7. Summary

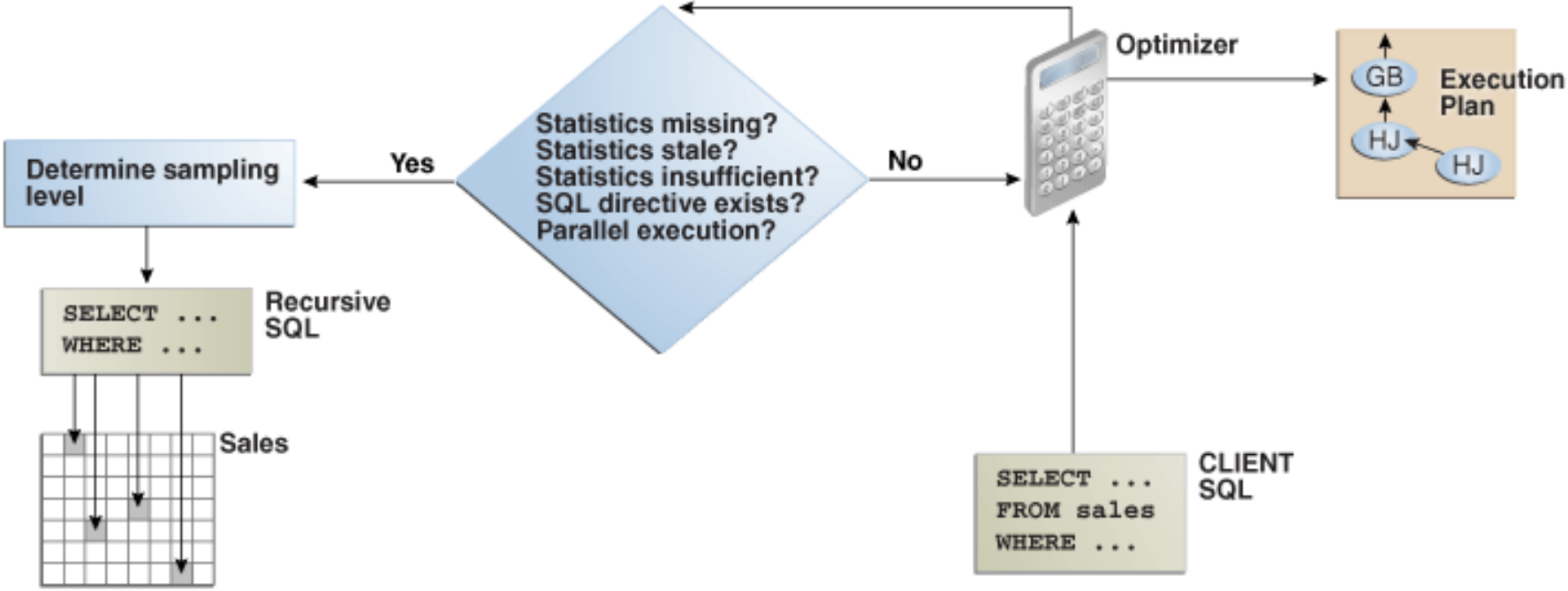
Introduction

In 12c Dynamic Statistics Are Considered an Adaptive Query Optimization Technique



Reference: Oracle Database SQL Tuning Guide 12.1

The Aim of Dynamic Statistics Is to Improve the Quality of the Optimizer Decisions



Reference: Oracle Database SQL Tuning Guide 12.1

Dynamic Statistics vs. (Adaptive) Dynamic Sampling

Dynamic statistics and *dynamic sampling* are synonyms.

In 12c a new implementation called *adaptive dynamic sampling* (ADS) is available.

The former implementation, which wasn't improved in 12c, is called *old style dynamic sampling.*

This presentation focuses on ADS.

Disclaimer

The official documentation provides little information about ADS.

Most of the content of this presentation is based on investigations carried out with self-written test cases as well as observations made in real-live projects.

Almost certainly, the authors didn't completely or correctly understand all aspects of adaptive dynamic sampling:

- The content summarizes our knowledge as of today
- Our knowledge is subject to change in the future

When It Is Used?

Controlling Adaptive Dynamic Sampling

ADS is enabled when OPTIMIZER_DYNAMIC_SAMPLING is set to 11.

ADS is a 12c feature that was back ported to 11.2.0.4.

Even though ADS is considered an adaptive query optimization technique, it's not controlled by OPTIMIZER_ADAPTIVE_FEATURES.

OPTIMIZER_DYNAMIC_SAMPLING=11 Overrules OPTIMIZER_FEATURES_ENABLE

```
Outline Data
-------------
      ...
      OPT_PARAM('optimizer_dynamic_sampling' 11)
      DB_VERSION('12.1.0.2')
      OPTIMIZER_FEATURES_ENABLE('10.2.0.5')
      ...

Note
-----
   - dynamic statistics used: dynamic sampling (level=AUTO)
```

Dynamic Sampling and Parallel Processing

When OPTIMIZER_DYNAMIC_SAMPLING is set to the default value (2) and a parallel execution plan is considered, the optimizer can automatically choose the dynamic sampling level.

Which level is used?

- 11.2/12.1.0.1: 2 < level < 11
  - Provided OPTIMIZER_FEATURES_ENABLE ≥ 11.2.0.1
- 12.1.0.2: level = 11
  - Provided OPTIMIZER_FEATURES_ENABLE ≥ 12.1.0.1

Dynamic Sampling and SQL Plan Directives

SQL plan directives instruct the database engine to take one of two measures:

- Use dynamic sampling
  - Provided both dynamic sampling and the SQL plan directives are enabled
- Create extended statistics

Which dynamic sampling level is used?

- 12.1.0.1: the current level is used (not sensible if level < 4)
  - Bug 16571451
- 12.1.0.2: level = 11

When Does Adaptive Dynamic Sampling Take Place?

It isn't limited (as the old style dynamic sampling) to single-table cardinality estimations involving

- missing statistics,
- expressions, or
- predicates referencing several columns of the same table

The query optimizer can basically use ADS for all SQL statements!

It's used for the estimation of single-table cardinalities as well as join cardinalities (12.1?) and query block cardinalities (12.1.0.2).

Time Limit

Time Limit

The dynamic sampling phase can't exceed a given amount of time called *time limit* (the minimum is 1 second).

It depends on whether the SQL statement is in the cursor cache or AWR.

- AWR is checked only if the lookup in the cursor cache is unsuccessful
- CONTROL_MANAGEMENT_PACK_ACCESS is evaluated by 12.1.0.2 only

If it's *not* present, the time limit defaults to 10 seconds.

If it's present, it depends on the CPU utilization and the number of executions.

- E.g. for less than 10 executions: time limit = CPU utilization / 10
- 11.2.0.4/12.1.0.1 always use the default

Time Limit – Cursor Cache Query Used in 12.1.0.2 (frjd8zfy2jfdq)

```
SELECT executions, end_of_fetch_count, elapsed_time/px_servers elapsed_time,
       cpu_time/px_servers cpu_time, buffer_gets/executions buffer_gets
FROM (SELECT sum(executions) AS executions,
             sum(case when px_servers_executions > 0
                 then px_servers_executions
                 else executions end) AS px_servers,
             sum(end_of_fetch_count) AS end_of_fetch_count,
             sum(elapsed_time) AS elapsed_time,
             sum(cpu_time  ) AS cpu_time,
             sum(buffer_gets) AS buffer_gets
      FROM gv$sql
      WHERE executions > 0
      AND sql_id = :1
      AND parsing_schema_name = :2)
```

Time Limit – AWR Query Used in 12.1.0.2 (4b4wp0a8dvkf0)

```
SELECT executions, end_of_fetch_count, elapsed_time/px_servers elapsed_time,
       cpu_time/px_servers cpu_time, buffer_gets/executions buffer_gets
FROM (SELECT sum(executions_delta) AS executions,
             sum(case when px_servers_execs_delta > 0
                 then px_servers_execs_delta else executions_delta end) AS px_servers,
             sum(end_of_fetch_count_delta) AS end_of_fetch_count,
             sum(elapsed_time_delta) AS elapsed_time,
             sum(cpu_time_delta) AS cpu_time,
             sum(buffer_gets_delta) AS buffer_gets
      FROM dba_hist_sqlstat s, v$database d, dba_hist_snapshot sn
      WHERE s.dbid = d.dbid
      AND bitand(nvl(s.flag, 0), 1) = 0
      AND sn.end_interval_time > (SELECT systimestamp AT TIME ZONE dbtimezone FROM dual)-7
      AND s.sql_id = :1
      AND s.snap_id = sn.snap_id
      AND s.instance_number = sn.instance_number
      AND s.dbid = sn.dbid
      AND parsing_schema_name = :2)
```

Time Slice

During the dynamic sampling phase several queries can be executed.

Each query can’t exceed a given amount of time called *time slice*.

$$timeSlice = \frac{timeLimit \cdot 1000}{\#candTabs \cdot 4}\ [ms]$$

Each query that exceeds the time slice should be stopped.

- When SQL trace is enabled, an ORA-10173 (Dynamic Sampling time-out error) may be observed

Strategies

Sampling Queries

With old style dynamic sampling the optimizer runs a single sampling query.

- The sampling percentage depends on the dynamic sampling level

With adaptive dynamic sampling the optimizer runs several sampling queries.

- The number depends on the complexity of the query and the number of available indexes
- Even for queries accessing few tables, it's not unusual at all to see 10-30 sampling query for a single parse
- The number also depends on the database version
  - Newer releases usually executes more sampling queries

Block Sampling

Sampling queries can use block sampling to reduce the amount of processing.

The optimizer decides how many blocks to scan based on the table's size.

- Sampling is used for tables larger than 800 blocks

The sampling queries uses the "SAMPLE BLOCK (*x*,8) SEED(*y*)" syntax.

- For the 1st execution, the observed values for *x* are percentages that lead to either 800, 8000 or 80000 blocks being read
- If the 1st execution doesn't return reliable data, *x* is increased
  - The typical increase factor is 2

Single Table Cardinality Adjustments

For each table having columns referenced in the WHERE clause, at least one sampling query is executed.

- The sample clause is optional
- When the sampling percentage has to be increased, several queries are executed

The sampling queries access data through either a full table scan or an index range scan.

- NO_INDEX_FFS specified

The WHERE clause of the sampling queries contains all filter conditions that apply to the sampled table.

Index Cardinality Adjustments

For each index having columns referenced in the WHERE clause, two or more sampling queries are usually executed.

- The actual number depends on the accuracy of the estimations

Data is typically accessed through an index-only (range) scan.

- Only indexed columns are accessed
- INDEX is specified

The sample clause is not frequently used.

- When it's used, the choice between a index range scan and a full table scan is left to the optimizer (NO_INDEX_FFS specified)

Join Cardinality Adjustments (12.1?)

For some (not all!) of the joins, sampling queries accessing only two tables are executed.

- The sample clause can be added to a single table only
- When the sampling percentage has to be increased, several queries for the same join are executed

Sampling queries are executed also in 11.2.0.4, but no adjustments were observed.

Query Block Cardinality Adjustments (12.1.0.2)

Query block cardinality adjustments can take place in different situations:

- For each GROUP BY clause, two sampling queries performing the aggregation are usually executed
  - No sample clause!
- For some joins, sampling queries referencing more than two tables can be executed
- For (unmergeable) subqueries, several sampling queries are usually executed

The WHERE clause of the sampling queries contains all available filter and join conditions.

Caching

Caching

Two caches are used:

- Internal cache
- Result cache

The aim of the internal cache is to avoid the execution of sampling queries that would be executed several times during a single parse operation.

- E.g. because of query transformations

The aim of the result cache is to avoid the execution of sampling queries that would be executed by different parse operations.

Caching – Result Cache

Requirement: RESULT_CACHE_MAX_SIZE > 0 (Enterprise Edition only)

To activate the result cache a hint is added to the sampling queries:

- 11.2.0.4/12.1.0.1: RESULT_CACHE
- 12.1.0.2: RESULT_CACHE(SNAPSHOT=3600)

Cache invalidation:

- 11.2.0.4/12.1.0.1: based on the usual result cache features
- 12.1.0.2: the SNAPSHOT parameter specifies for how long (in seconds) an entry remains cached

Fallacies

Exceeding Time Limit and/or Time Slice

The SQL engine isn't always able to (exactly) limit the allotted time.

Even though 12.1.0.2 is slightly better than 12.1.0.1/11.2.0.4 in this regard, there are situations where way to much time is spent for dynamic sampling.

Available Object Statistics Are Ignored

Adaptive dynamic sampling might be used despite the fact that the available object statistics are sufficient to compute sound estimations.

Two are the issues with this strategy:

- The parse takes longer
- The adjustments can make the estimations worse

Bug# 19728268: Partition-Extended Names Not Applied

This problem typically leads to overestimates.

The RDF Semantic Graph option is impacted by this bug.

The bug was fixed and a patch for 12.1.0.2 can be requested.

Bug# 21260111: V$SQL_PLAN.OTHER_XML Can Contain Wrong Dynamic Sampling Level

When all the following conditions are fulfilled, V$SQL_PLAN.OTHER_XML contains the current level instead of 11:

- 12.1.0.2
- Level < 11
- Adaptive dynamic sampling activated through SQL plan directive

DBMS_XPLAN also shows the wrong level.

Summary

Summary

Adaptive dynamic sampling can provide very useful information to the optimizer.

More (in some situations, much more!) parse time can be required.

The feature needs to be carefully tested!

- Every version has a slightly different implementation

Questions and Answers

Udo Fohrmann
Udo.fohrmann@trivadis.com

trivadis
makes IT easier.