Adaptive Features – Fluch oder Segen

Thomas Lehmann
Robotron Datenbank-Software GmbH
Dresden

Schlüsselworte
Adaptive query optimization, adaptive plan, adaptive statistics, dynamic statistics, auto reoptimiziation, sql plan directives, oracle 12c, optimizer

Einleitung
Mit Oracle Version 12c wurden viele neue und interessante Erweiterungen für den Optimizer eingeführt. Unter dem Schlagwort „Adaptive Query Optimization“ werden verschiedene Methoden der Optimierung zusammengefasst.

Adaptive Query Optimization
Unter diesem Oberbegriff gliedern sich die zwei Bereiche *Adaptive Pläne* und *Adaptive Statistiken*.

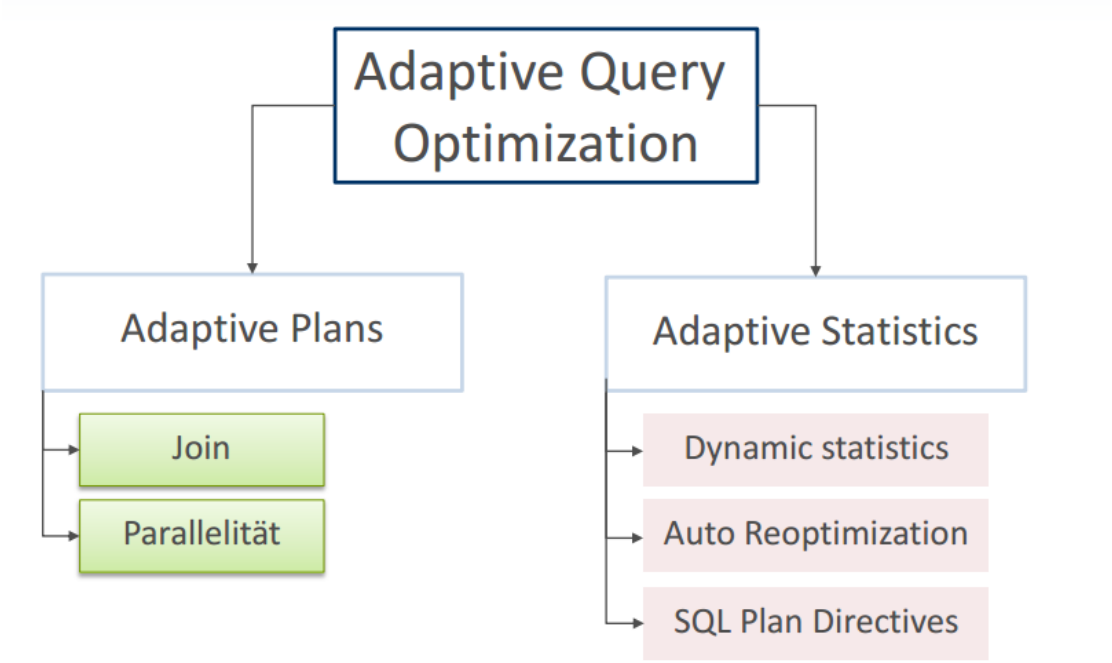

Abb. 1: Unterteilung der Adaptiven Features

Voraussetzungen zur Nutzung der neuen Optimizer Features sind die Oracle Version 12c und folgende Initialisierungsparameter:

```
COMPATIBLE = 12.1

OPTIMIZER_ADAPTIVE_FEATURES = TRUE

OPTIMIZER_ADAPTIVE_REPORT_ONLY = FALSE
```

Adaptive Pläne
Adaptive Plans ermöglichen es dem Optimizer quasi während der Ausführung seinen Ausführungsplan zu wechseln. Dabei ist die Möglichkeit des Optimizers auf den Grad der Parallelität (bei paralleler Ausführung) und auf den Wechsel zwischen NESTED LOOP und HASH JOIN begrenzt.

Die Planänderung wirkt sich allerdings nur auf die erste Ausführung des Statements aus. Ist der Ausführungsplan gecacht und als IS_RESOLVED_ADAPTIVE_PLAN markiert, erfolgt keine erneute Optimierung – auch nicht bei Änderung der Datenlage.
Den vollständigen Ausführungsplan (inkl. der entfernten Zeilen) kann man sich per DBMS_XPLAN mit format =>‘adaptive‘ anzeigen lassen.

```
SQL> SELECT * FROM TABLE(DBMS_XPLAN.display_cursor(format => 'adaptive'));

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------------------------------------
------------------------------------------------------
SQL_ID  007snhj87caq0, child number 0
-------------------------------------
SELECT b.bestellung,        p.artikel FROM   bestellungen b        JOIN
positionen p ON p.bs_id = b.id WHERE  b.nummer = 1

Plan hash value: 3368533918

-------------------------------------------------------------------------------------------------------------------
|   Id  | Operation                                | Name              | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------------------------------------------
|     0 | SELECT STATEMENT                         |                   |       |       |     3 (100)|          |
|-  *  1 |  HASH JOIN                               |                   |    25 |   650 |     3   (0)| 00:00:01 |
|     2 |   NESTED LOOPS                           |                   |    25 |   650 |     3   (0)| 00:00:01 |
|     3 |    NESTED LOOPS                          |                   |    25 |   650 |     3   (0)| 00:00:01 |
|-    4 |     STATISTICS COLLECTOR                 |                   |       |       |            |          |
|     5 |      TABLE ACCESS BY INDEX ROWID BATCHED | BESTELLUNGEN      |     1 |    14 |     2   (0)| 00:00:01 |
|  *  6 |       INDEX RANGE SCAN                   | BESTELLUNGEN_NR_I |     1 |       |     1   (0)| 00:00:01 |
|  *  7 |     INDEX RANGE SCAN                     | POS_BES_FK_I      |    25 |       |     0   (0)|          |
|     8 |    TABLE ACCESS BY INDEX ROWID           | POSITIONEN        |    25 |   300 |     1   (0)| 00:00:01 |
|-    9 |   TABLE ACCESS FULL                      | POSITIONEN        |    25 |   300 |     1   (0)| 00:00:01 |
-------------------------------------------------------------------------------------------------------------------

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

   1 - access("P"."BS_ID"="B"."ID")
   6 - access("B"."NUMMER"=1)
   7 - access("P"."BS_ID"="B"."ID")

Note
-----
   - this is an adaptive plan (rows marked '-' are inactive)
```

Abb.2: Ausführungsplan eines adaptiven Plans

Adaptive Statistiken

Unter dem Begriff *Dynamic Sampling* konnte der Optimizer auch in älteren Versionen bereits Schätzungen über den Füllstand und Werteverteilung innerhalb von Tabellen vor der eigentlichen Ausführung abschätzen, um einen optimalen Ausführungsplan zu ermitteln.

Mit Oracle 12c wird aus *Dynamic Sampling Dynamic Statistics*. Zudem wird die Funktionalität des Optimizers durch ein neues Level 11 (OPTIMIZER_DYNAMIC_SAMPLING) erweitert.

Damit kann der Optimizer auch bei vorhandenen Statistiken einen besseren Ausführungsplan ermitteln. Die Ergebnisse der Abschätzungen können im Result-Cache der Datenbank zwischengepuffert werden und für weitere, ähnliche Abfragen wiederverwendet werden.

```
PLAN_TABLE_OUTPUT
-------------------------------------------------------------------------------------------------
SQL_ID  g6zqaad4rpka5, child number 0
-------------------------------------
select /*+ gather_plan_statistics */ count(*)  from
REF_ADR_POSTLEITZAHLEN where reg_id = 'SN' and plz like '012%'

Plan hash value: 1348772103

-----------------------------------------------------------------------------------------------
| Id  | Operation         | Name            | Starts | E-Rows | A-Rows |   A-Time   | Buffers |
-----------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |                 |      1 |        |      1 |00:00:00.01 |       3 |
|   1 |  SORT AGGREGATE   |                 |      1 |      1 |      1 |00:00:00.01 |       3 |
|*  2 |   INDEX RANGE SCAN| RADRP_PLZ_REG_I |      1 |     81 |     82 |00:00:00.01 |       3 |
-----------------------------------------------------------------------------------------------

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

   2 - access("PLZ" LIKE '012%' AND "REG_ID"='SN')
       filter(("REG_ID"='SN' AND "PLZ" LIKE '012%'))

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

```
----------------------
PARSING IN CURSOR #140520530525080 len=559 dep=1 uid=328 oct=3 lid=328 tim=1566696108 hv=305954488 ad='73fbd728' sqlid='b94cbvb93szps'
SELECT /* DS_SVC */ /*+ dynamic_sampling(0) no_sql_tune no_monitoring optimizer_features_enable(default) no_parallel result_cache(snapshot=3600) OPT_EST
OM (SELECT /*+ qb_name("innerQuery") INDEX( "REF_ADR_POSTLEITZAHLEN" "RADRP_PLZ_I")  */ COUNT(*) AS C1, 4294967295 AS C2, SUM(CASE WHEN ("REF_ADR_POSTLE
" "REF_ADR_POSTLEITZAHLEN" WHERE ("REF_ADR_POSTLEITZAHLEN"."PLZ" LIKE '012%')) innerQuery
END OF STMT
```

Abb. 3:Ausfühungsplan dynamic statistics

Automatic Reoptimization

Darunter versteht man die Möglichkeit des Optimizers, die geschätzten Werte je Zeile in einem Ausführungsplan mit den tatsächlichen Werten zu vergleichen und bei Bedarf einen besseren Ausführungsplan bei der zweiten Ausführung des Statements zu ermitteln.

```
select /*+ gather_plan_statistics */ count(*)  from
REF_ADR_POSTLEITZAHLEN where reg_id = 'SN' and plz like '012%'

Plan hash value: 1348772103

-------------------------------------------------------------------------------------------------
| Id  | Operation         | Name            | Starts | E-Rows | A-Rows |   A-Time   | Buffers |
-------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |                 |      1 |        |      1 |00:00:00.01 |       3 |
|   1 |  SORT AGGREGATE   |                 |      1 |      1 |      1 |00:00:00.01 |       3 |
|*  2 |   INDEX RANGE SCAN| RADRP_PLZ_REG_I |      1 |     82 |     82 |00:00:00.01 |       3 |
-------------------------------------------------------------------------------------------------

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

   2 - access("PLZ" LIKE '012%' AND "REG_ID"='SN')
       filter(("REG_ID"='SN' AND "PLZ" LIKE '012%'))

Note
-----
   - statistics feedback used for this statement
```

Abb. 4:Ausfühungsplan statistics feedback

SQL Plan Directives

SQL Plan Directives sind zusätzliche Informationen für den Optimizer. Diese werden automatisch bei der Ausführung generiert und aller 15 Minuten im SYSAUX persistiert. Außerdem können dadurch Extended Statistiken, also Statistiken über mehrere Spalten, automatisch generiert werden.

Kontaktadresse:
Thomas Lehmann
Robotron Datenbank-Software GmbH
Stuttgarter Straße 29
D-01189 Dresden

Telefon: +49 (0) 351-25859 2782
Fax: +49 (0) 351-25859 3696
E-Mail Thomas.Lehmann@robotron.de
Internet: www.robotron.de