

Performance Tuning mit Oracle 12c

Lutz Fröhlich
IT Consulting
Darmstadt

Schlüsselworte

Performance-Optimierung

Adaptive Statistics

Adaptive Execution Plans

SQL-Plandirektiven

Konkurrierendes Sammeln von Statistiken

Spaltengruppenstatistiken

Private Session-Statistiken

Dynamic Sampling

SQL Realtime Monitoring

Einleitung

Oracle 12c hält eine breite Palette neuer Features für die SQL-Optimierung bereit. Eines der herausragenden Features sind der Adaptive Ausführungsplan sowie Adaptive Statistiken. Ein bedeutsamer Schritt zur Verlagerung der Optimierung auf den Ausführungszeitpunkt. Darüber hinaus wurde der Optimizer in der Version 12c in weiteren Punkten verbessert. Voraussetzung dafür sind neue Typen von Histogrammen sowie SQL-Plandirektiven. Nicht zu vergessen ist die „Automatische Wiederoptimierung (Automatic Reoptimization)“. Die Idee die sich dahinter verbirgt ist, die Erfahrungen aus der erstmaligen Ausführung einer SQL-Anweisung für zukünftige Ausführungen zu nutzen.

Eine Weiterentwicklung fand in den Bereichen „SQL Plan Management“, „Konkurrierendes Sammeln von Statistiken“ sowie „SQL Real Time Monitoring“ statt.

Adaptive Optimierung von SQL-Anweisungen

In Oracle 12c verfügt der Optimizer über die Fähigkeit, Anpassungen während der Ausführung von SQL-Anweisungen vorzunehmen, um einen besseren Ausführungsplan zu finden. Dazu kann es erforderlich sein, die vorliegenden Statistiken anzupassen. Das Feature nennt sich „Adaptive Statistiken“.

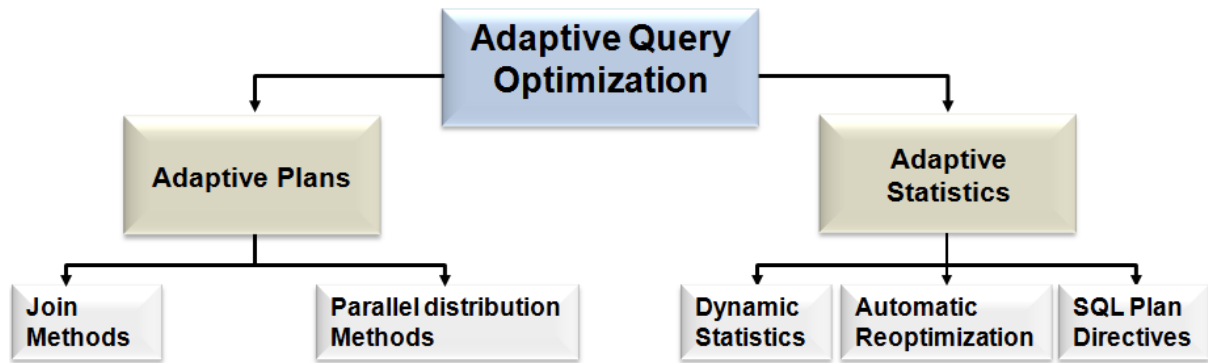


Abb. 1: Übersicht der Komponenten für Adaptive SQL-Optimierung

Adaptive Ausführungspläne

Die Anpassung von Ausführungsplänen zur Laufzeit kann Einfluss auf alle Entscheidungen haben, die der Optimizer im Vorfeld getroffen hat. Die schließt eine erneute Evaluierung der Join-Methoden ein. Dabei arbeitet der Optimizer mit mehreren Sub-Plänen, die bereits im Vorfeld evaluiert wurden. So kann der Optimizer die Entscheidung treffen, von der Nested Loop-Methode zur Hash Join-Methode für einen Sub-Plan zu wechseln.

Um alle Operationen in einem Adaptiven Plan zu sehen, muss der Parameter „format“ mit dem Wert „+adaptive“ angegeben werden:

```
SQL> SELECT * FROM TABLE(dbms_xplan.display_cursor(
sql_id => 'cvhk2j2gymzhd',format => '+adaptive'));
```

```

-----
|  Id  | Operation                                | Name          |
-----+-----+-----+
|  0   | SELECT STATEMENT                          |               |
|- *  1   | HASH JOIN                                  |               |
|     2   | NESTED LOOPS                              |               |
|-   3   | STATISTICS COLLECTOR                      |               |
|- *  4   | HASH JOIN                                  |               |
|     5   | NESTED LOOPS                              |               |
|-   6   | STATISTICS COLLECTOR                      |               |
| *   7   | HASH JOIN                                  |               |
|- *  8   | HASH JOIN                                  |               |
|     9   | NESTED LOOPS                              |               |
|    10  | NESTED LOOPS                              |               |
|-   11  | STATISTICS COLLECTOR                      |               |
|    12  | NESTED LOOPS                              |               |
|- *  13  | HASH JOIN                                  |               |
|    14  | NESTED LOOPS                              |               |
|-   15  | STATISTICS COLLECTOR                      |               |
|    16  | NESTED LOOPS                              |               |
|    17  | TABLE ACCESS BY INDEXID                  | USER$         |
| *  18  | INDEX UNIQUE SCAN                          | I_USER1       |
-----

```

Note

```
-----
```

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

Adaptive Statistiken

Die Qualität des Ausführungsplans wird entscheidend durch die Qualität der Statistiken bestimmt. Die Inhalte von Tabellen können sich zum Ausführungszeitpunkt gegenüber dem Zeitpunkt des Sammelns der Statistiken geändert haben. Während der Parsing-Phase entscheidet der Optimizer, ob ein Teil der Statistiken aktualisiert oder neu gesammelt wird. Dieser Prozess ist als „Dynamic Sampling“ bekannt. In Oracle 12c wurde das Dynamic Sampling signifikant verbessert und erweitert.

Automatic Reoptimization

Bei der Reoptimization fließen die Erfahrungen aus der erstmaligen Ausführung einer SQL-Anweisung an den Optimizer zurück. Er verwendet diese Informationen, um für folgende Ausführungen derselben Anweisung möglicherweise einen besseren Plan zu finden.

Dafür werden folgende Feedbacks an den Optimizer gegeben:

- Statistisches Feedback
- Performance Feedback

SQL Plan-Direktiven

SQL Plan-Direktiven werden automatisch auf Basis der Informationen, die im Rahmen der „Automatic Reoptimization“ erstellt wurden. Dabei handelt es sich um zusätzliche Informationen, die der Optimizer für die Erstellung eines adaptiven Ausführungsplans verwendet. EINE SOLCHE Direktive kann zum Beispiel den Optimizer veranlassen, dynamische Statistiken in bestimmten Bereichen zu erstellen.

Gespeicherte SQL Plan-Direktiven können mit Hilfe der folgenden Views abgefragt werden:

DBA_SQL_PLAN_DIRECTIVES

DBA_SQL_PLAN_DIR_OBJECTS

```
SELECT TO_CHAR(p.directive_id), o.owner, o.object_name, p.state,
p.reason
FROM dba_sql_plan_directives p, dba_sql_plan_dir_objects o
WHERE p.directive_id = o.directive_id
```

```
TO_CHAR(P.DIRECTIVE_  OWN OBJEC STATE      REASON
-----
5854062426711973735 SYS USER$ NEW          JOIN CARDINALITY MISESTIMATE
8030769718917252823 SYS USER$ HAS_STATS  SINGLE TABLE CARDINALITY MIS
1404992050398360524 SYS USER$ PERMANENT JOIN CARDINALITY MISESTIMATE
```

Private Session-Statistiken

Obwohl Global Temporary Tables von allen Sessions benutzt werden können, ist der Inhalt privater Natur. In Oracle 12c ist es nun möglich, mit Session-spezifischen zu arbeiten. Dieses Feature kann durch die Einstellung „GLOBAL_TEMP_TABLE_STATS“ im Paket DBMS_STATS beeinflusst werden. Die Standardeinstellung ist „SESSION“, d.h. Oracle erstellt Session-spezifische Statistiken. Ist die Einstellung auf „SHARED“ gesetzt, dann verwendet der Optimizer die globalen Statistiken für die Tabelle.

Konkurrierendes Sammeln von Statistiken

Das konkurrierende Sammeln von Statistiken wurde in 11g eingeführt. Oracle verwendet dafür den Job Scheduler sowie das Feature „Advanced Queueing“. In der Version 12c wurde das Verfahren optimiert um jeden einzelnen Job besser auszunutzen.

Kontaktadresse:

Lutz Fröhlich
IT Consulting
Sandstr. 38
D-64283 Darmstadt

Telefon: +49 (0) 6151-5208530
Fax: +49 (0) 6151-5208531
E-Mail: Lutz@LutzFroehlich.de
Internet: www.lutzfroehlich.de