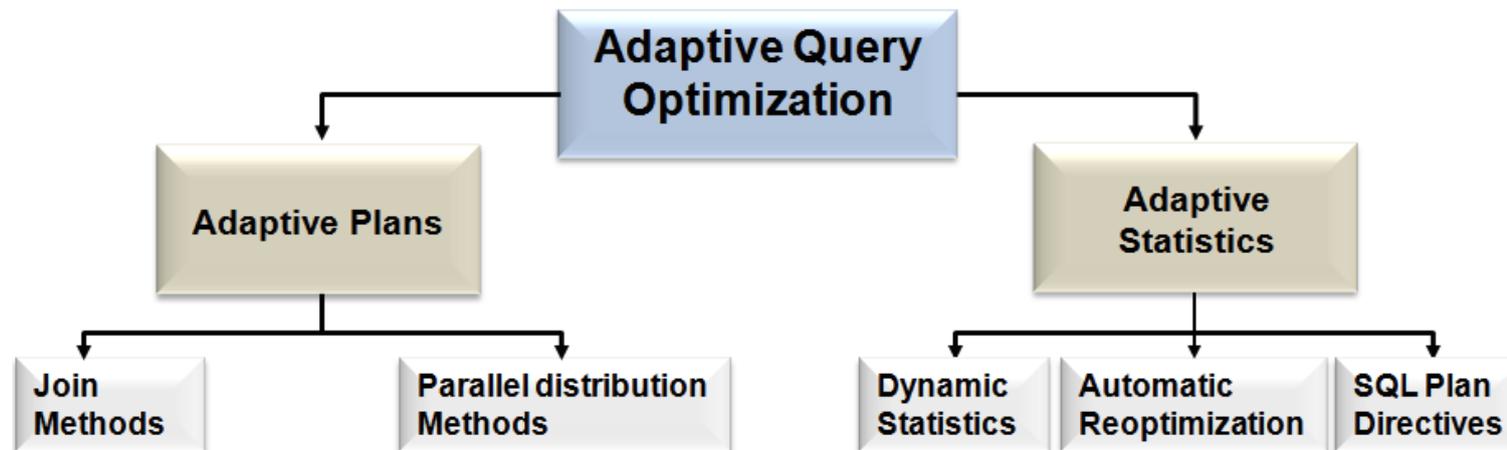


Selbstlernender Optimizer in Oracle 12

Vortrag DOAG-Regiogruppe Freiburg/Südbaden 25.10.16

Adaptive Query Optimization

- neue und erweiterte Funktionen zur dynamischen Anpassung von Ausführungsplänen
- vor allem bei fehlenden oder unzureichenden Statistiken
- zwei Schwerpunkte
 - adaptive Ausführungspläne
 - adaptive Statistiken



Quelle: Oracle[1]

Adaptive Pläne / Join Methods (1)

- Optimizer berechnet Ausführungsplan incl. alternativen (Teil-)Plänen
- vor joins wird ein „statistic collector“ eingefügt
- bei der ersten Ausführung vergleicht der „statistic collector“ fortlaufend die Anzahl „gesehener“ Zeilen mit der vom Optimizer geschätzte Zahl
- bei „signifikanter“ Abweichung wird auf den alternativen Plan gewechselt
- nach der ersten Ausführung wird der statistic collector deaktiviert und der tatsächlich verwendete Plan als „endgültig“ markiert
- derzeit: nur Wechsel zwischen „nested loop join“ und „hash join“
- Zusammenfassung:
„nested loop join“ vs. „hash join“ wird erst während der ersten Ausführung entschieden

Adaptive Pläne / Join Methods (2)

- `V$SQL.IS_RESOLVED_ADAPTIVE_PLAN`
 - NULL kein adaptiver Plan vorhanden
 - Y adaptive Pläne vorhanden, finaler Plan entschieden
 - N adaptive Pläne vorhanden, noch nicht entschieden
 - R adaptive Pläne vorhanden, nur Reporting
- „explain plan“ zeigt den default Ausführungsplan
- „display_cursor()“ zeigt den finalen Ausführungsplan
- „display_cursor(format=>'adaptive')“ zeigt alle Pläne
 - inaktive Teile sind mit „-“ gekennzeichnet
 - Fussnote enthält Hinweis „adaptive plan“
 - Voraussetzung:
 - Hint `/* gather_plan_statistics */`
 - `alter session set statistics_level=all`

Adaptive Pläne / Join Methods (3)

```
SQL> select * from table(dbms_xplan,display_cursor(format=>'adaptive'));

PLAN_TABLE_OUTPUT
-----
SQL_ID d3mzkmzxn264d, child number 0
-----
select /*+ gather_plan_statistics */ p.product_name from order_items2
o, product_information p where o.unit_price = 15 and o.quantity > 1
and p.product_id = o.product_id

Plan hash value: 2886494722

-----
| Id | Operation | Name | Rows | Bytes | Cost (%CPU)|
-----
| 0 | SELECT STATEMENT | | | | 7 (100)|
| * 1 | HASH JOIN | | 4 | 128 | 7 (0)|
|- 2 | NESTED LOOPS | | | | |
|- 3 | NESTED LOOPS | | 4 | 128 | 7 (0)|
|- 4 | STATISTICS COLLECTOR | | | | |
| * 5 | TABLE ACCESS FULL | ORDER_ITEMS2 | 4 | 48 | 3 (0)|
|- * 6 | INDEX UNIQUE SCAN | PRODUCT_INFORMATION_PK | 1 | | 0 (0)|
|- 7 | TABLE ACCESS BY INDEX ROWID | PRODUCT_INFORMATION | 1 | 20 | 1 (0)|
| 8 | TABLE ACCESS FULL | PRODUCT_INFORMATION | 1 | 20 | 1 (0)|
-----

Predicate Information (identified by operation id):
-----
 1 - access("P"."PRODUCT_ID"="0"."PRODUCT_ID")
 5 - filter(("O"."UNIT_PRICE"=15 AND "O"."QUANTITY">1))
 6 - access("P"."PRODUCT_ID"="0"."PRODUCT_ID")

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

Quelle: Oracle[1]

Adaptive Pläne / Join Methods (4)

- Parameter `_OPTIMIZER_ADAPTIVE_PLANS`
 - adaptive Pläne (de)aktivieren (default: TRUE)
 - auf System- oder Session-Ebene möglich
 - als Hint: `ADAPTIVE_PLAN` bzw. `NO_ADAPTIVE_PLAN`
- Parameter `_OPTIMIZER_ADAPTIVE_REPORTING_ONLY`
 - TRUE die Informationen für adaptive Pläne werden gesammelt und können ausgezeigt werden, werden aber nicht angewendet (d.h. es wird immer der default plan angewendet)
 - default: FALSE

Adaptive Pläne / Parallel Distribution Method

- wie adaptive Pläne, jedoch bzgl. der Verteilung der Daten bei paralleler Verarbeitung
- ohne Relevanz für das dem Vortrag zu Grunde liegende Projekt
- Details siehe Oracle[1]

Adaptive Statistics / Dynamic Statistics (1)

- Oracle 11.2: „dynamic sampling“
- Optimizer ersetzt fehlende oder unzureichende Statistiken durch dynamisch erstellte Stichproben-Statistiken
- soll die Schätzung der Kardinalitäten verbessern
- für Zugriffe auf Basis-Tabellen, Indices, ..
- neu: auch für joins, group by und komplexe Prädikate
- Umfang der erstellten Statistiken hängt ab von
 - der Komplexität des SQL
 - den vorhandenen Statistiken
 - der voraussichtlichen Ausführungszeit
 - den verfügbaren Ressourcen
 - Parameter OPTIMIZER_DYNAMIC_SAMPLING
- bei OPTIMIZER_DYNAMIC_SAMPLING=11 können dynamische Statistiken erstellt werden auch wenn alle notwendigen Statistiken vorhanden sind

Adaptive Statistics / Dynamic Statistics (2)

- Ergebnisse werden im Cache gespeichert und können für beliebige Cursor genutzt werden
- der Zeitaufwand für die Erstellung der Statistiken verlängert die Zeit für das Parsen!
- der Zeitaufwand für die Erstellung der Statistiken ist begrenzt - insgesamt und pro Statistik (komplexe Formel)
- Parameter
 - OPTIMIZER_DYNAMIC_SAMPLING
 - 0 keine dynamische Statistiken
 - 2 default
 - 11 Optimizer entscheidet nach eigenem Gutdünken
(Seiteneffekte! z.B. OPTIMIZER_FEATURES_ENABLE wird ignoriert)als Hint: DYNAMIC_SAMPLING(0) usw.
 - _OPTIMIZER_ADS_TIME_LIMIT (default: 10 Sekunden)
 - _OPTIMIZER_ADS_USE_RESULT_CACHE (default: TRUE)
 - _OPTIMIZER_DSDIR_USAGE_CONTROL

Adaptive Statistics / Automatic Reoptimization

- adaptive Pläne: Anwendung nur bei der ersten Ausführung
- automatic Reoptimization: Informationen aus einer Ausführung werden gespeichert und bei nachfolgenden Ausführungen verwendet
- Cursor wird dazu invalidiert (`V$SQL.IS_REOPTIMIZABLE = 'Y'`) und bei der nächsten Ausführung ein neuer „hard parse“ durchgeführt
- kann u.U. mehrfach wiederholt werden
- soll Pläne mit schlechten Schätzungen verbessern
- derzeit v.a. „statistic feedback“

Adaptive Statistics / Statistic Feedback (1)

- vormalig: „cardinality feedback“
- wird angewendet bei
 - fehlenden Statistiken
 - mehrfachen Prädikaten auf Tabellen
 - Prädikaten mit komplexen Operatoren
- nach einer Ausführung werden die tatsächlichen Kardinalitäten („A-Rows“) mit den vom Optimizer geschätzten Kardinalitäten („E-Rows“) verglichen

```
SQL> select /*+gather_plan_statistics*/ c.cust_first_name, c.cust_last_name, sum(s.amount_sold)
  2 from customers c, sales s
  3 where c.cust_id=s.cust_id
  4 and   c.cust_city='Los Angeles'
  5 and   c.cust_state_province='CA'
  6 and   c.country_id='US'
  7 and   s.time_id='18-FEB-00'
  8 group by c.cust_first_name, c.cust_last_name;
```

Id	Operation	Name	Starts	E-Rows	A-Rows
0	SELECT STATEMENT		1		0
1	HASH GROUP BY		1	1	0
* 2	HASH JOIN		1	1	0
3	PARTITION RANGE SINGLE		1	5	40
* 4	TABLE ACCESS FULL	SALES	1	5	40
* 5	TABLE ACCESS FULL	CUSTOMERS	1	1	13

Initial Cardinality estimates are more than 8X off

Quelle: Oracle[1]

Adaptive Statistics / Statistic Feedback (2)

- bei „signifikanter“ Abweichung von „E-Rows“ und „A-Rows“
 - werden die tatsächlichen Kardinalitäten im Cache gespeichert
 - für den Cursor IS_REOPTIMIZABLE='Y' gesetzt
 - bei der nächste Ausführung ein erneuter „hard parse“ durchgeführt und dabei die gespeicherten Kardinalitäten verwendet
 - eine SQL Plan Directive erzeugt
- wenn die (ursprünglichen) Schätzungen „korrekt“ waren, dann wird das Monitoring abgeschaltet und der Plan beibehalten (IS_REOPTIMIZABLE='N')
- Anwendung des statistischen Feedbacks sichtbar in Fussnote bei „display_cursor“

Note

- statistics feedback used for this statement

Quelle: Oracle[1]

- Parameter OPTIMIZER_USE_FEEDBACK
 - steuert nur die Verwendung des Feedbacks beim Parsen
 - SQL Plan Directive werden immer erzeugt!

Adaptive Statistics / Performance Feedback

- wie „statistic feedback“, jedoch bei der Verteilung der Daten bei paralleler Verarbeitung
- ohne Relevanz für das dem Vortrag zu Grunde liegende Projekt
- Details siehe Oracle[1]

Adaptive Statistics / SQL Plan Directives (1)

- werden automatisch erzeugt im Rahmen der „automatic reoptimization“
- enthalten zusätzliche Informationen und Hinweise für den Optimizer
- derzeit: nur „dynamic sampling“ (z.B. bei join über mehrere Spalten mit breiter Streuung der Werte)
- beziehen sich nicht auf SQL-Statements oder DB-Objekte, sondern auf Ausdrücke
- verwendbar für alle SQL-Statements mit „passenden“ Ausdrücken
- Beispiele:

Tabelle A	Spalte 1	COLUMN	DYNAMIC_SAMPLING	USABLE	SINGLE TABLE CARDINALITY MISESTIMATE
Tabelle A	Spalte 2	COLUMN	DYNAMIC_SAMPLING	USABLE	SINGLE TABLE CARDINALITY MISESTIMATE
Tabelle B		TABLE	DYNAMIC_SAMPLING	USABLE	JOIN CARDINALITY MISESTIMATE
Tabelle C		TABLE	DYNAMIC_SAMPLING	SUPERSEDED	JOIN CARDINALITY MISESTIMATE
Tabelle D		TABLE	DYNAMIC_SAMPLING	SUPERSEDED	SINGLE TABLE CARDINALITY MISESTIMATE

- kann dazu führen, dass beim nächsten Statistik-Lauf eine entsprechende „column group“ erstellt und diese dann an Stelle der SQL Plan Directive verwendet wird

Adaptive Statistics / SQL Plan Directives (2)

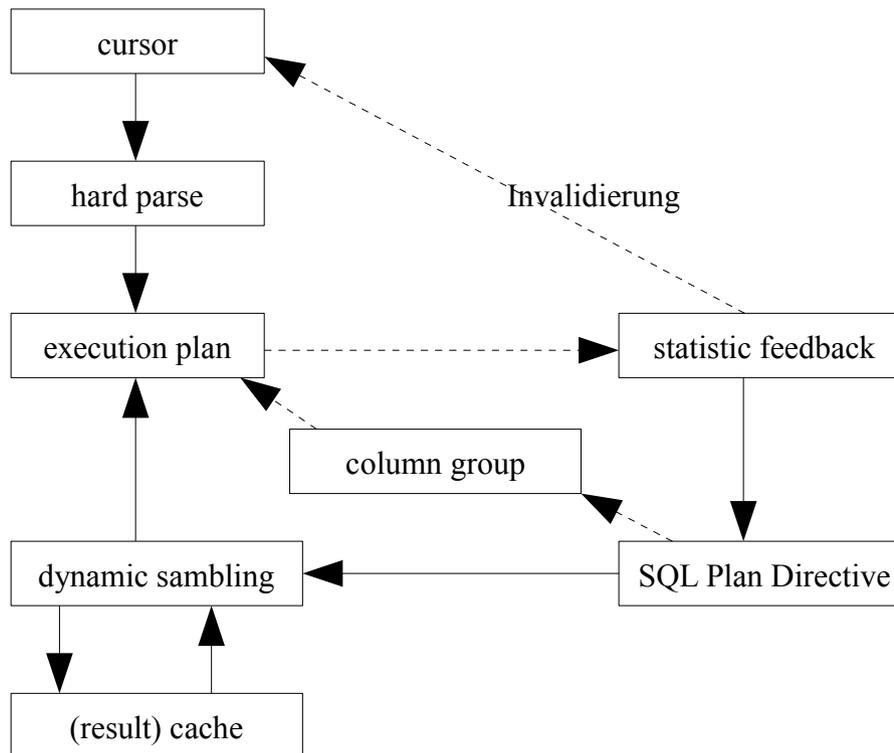
- SQL Plan Directive sind sichtbar in
 - DBA_SPLAN_DIRECTIVES und DBA_SQL_PLAN_DIR_OBJECTS
 - „explain plan“: alle anwendbaren SQL Plan Directiven
 - „display_cursor“: die Anzahl tatsächlich verwendeter Directiven

```
Note
-----
- dynamic statistics used: dynamic sampling (level=2)
- 2 Sql Plan Directives used for this statement
```

- die in einem Ausführungsplan tatsächlich verwendeten Directiven können nirgends angezeigt werden!
- nicht verwendete SQL Plan Directive werden nach 53 Wochen gelöscht
- SQL Plan Directive können manuell deaktiviert werden, aber kompliziert (werden ggf. automatisch wieder erzeugt ...)
- SQL Plan Directive können nicht komplett deaktiviert werden bzw. nur über OPTIMIZER_ADAPTIVE_FEATURES (??)

Zusammenfassung

- das ungefähre Zusammenwirken der Komponenten:



Projekterfahrungen (1)

- „select .. from view1 union select .. from view2“
- ca. 20 Basis-Tabellen, davon ca. 10 via outer join
- mehrere Tabellen mit 10 - 20 Mio. Datensätzen

- 25 unterschiedliche Ausführungspläne mit Laufzeiten zwischen 0.3 und 5 sec
(+ ein Ausreißer mit 20 sec)

- Ausführungspläne mit ca. 450 Zeilen (incl. adaptiven Plänen)
- ca. 90 anwendbare SQL Plan Directive
- in den Ausführungsplänen 1 bis 7 SQL Plan Directive genutzt

Projekterfahrungen (2)

- eingebaute Intelligenz ..
 - wenn diese funktioniert wie geplant, dann alles „super“
 - wenn nicht ... dann echte Probleme
- im positiven Fall
 - die Ausführungspläne werden schrittweise verbessert
 - die Ausführungspläne werden automatisch an geänderte Daten angepasst
 - sind die Auswirkungen nicht „sichtbar“, da „schneller“ keine Tickets auslöst
- im negativen Fall (wie im Projekt erlebt!) kann dies jedoch dazu führen, dass
 - sich die Ausführungspläne täglich ändern (z.B. wechselnde Anzahl SQL Plan Directiven, wechselnde join-Reihenfolgen, ..)
 - die Ausführungspläne schlechter werden (durch „falsche“ Entscheidungen)
 - die Ausführungszeiten instabil werden
 - die Auswirkung manueller Änderungen nicht mehr verlässlich beurteilt werden kann

Projekterfahrungen (3)

- Abhilfen
 - Deaktivieren einzelner Komponenten (s.o.)
 - Deaktivieren aller Komponenten
OPTIMIZER_ADAPTIVE_FEATURES=false
 - Profile
 - Baselines
- zuverlässig berechenbares Verhalten nur durch Baselines (oder SQL Profile)

Quellen

Oracle[1]: Oracle White Paper „Optimizer with Oracle Database 12c”

<http://www.oracle.com/technetwork/database/bi-datawarehousing/twp-optimizer-with-oracledb-12c-1963236.pdf>

Weiter interessante Links:

https://blogs.oracle.com/optimizer/entry/dynamic_sampling_and_its_impact_on_the_optimizer

<http://blog.dbi-services.com/sql-plan-directive-disabling-usage-and-column-groups/>

http://www.soug.ch/fileadmin/user_upload/SIGs/SIG_150521_Tuning_R/Christian_Antognini_AdaptiveDynamicSampling_trivadis.pdf

https://www.informatik-aktuell.de/fileadmin/templates/wr/pics/Seminare/Pr%C3%A4sentationen/Udo_Fohrmann_IT-Tage_2015.pdf

<https://jonathanlewis.wordpress.com/2014/11/05/cardinality-feedback-2/>

http://kerryosborne.oracle-guy.com/papers/12c_Adaptive_Optimization.pdf

<http://www.peasland.net/2015/11/05/adaptive-dynamic-stats-kills-performance-in-12-1-0-2-rac/>

https://bdrouvot.wordpress.com/2014/10/17/watch-out-for-optimizer_adaptive_features-as-it-may-have-a-huge-negative-impact/

Fragen?