



Datenbankstatistiken im Griff mit DBMS_STATS

DOAG 2012 Konferenz + Ausstellung

Nürnberg

21. November 2012

Herrmann & Lenz Services GmbH

Herrmann & Lenz Solutions GmbH

- Erfolgreich seit 1996 am Markt
- Firmensitz: Burscheid (bei Leverkusen)
- Beratung, Schulung und Betrieb/Fernwartung rund um das Thema Oracle Datenbanken
- Schwerpunktthemen: Hochverfügbarkeit, Tuning, Migrationen und Troubleshooting
- Herrmann & Lenz Solutions GmbH
 - Produkt: Monitoring Module
 - Stand Nr. 212 auf Ebene 2



Zwei Fakten über den Oracle Optimizer

1. Er ist nicht verantwortlich für schlechtes Datenbankdesign!
2. Er kann ohne Statistiken keine guten Ergebnisse liefern!



Ziele des Vortrags

- Zusammenhänge über das Sammeln von Statistiken verstehen
- Möglichkeiten zur gezielten Manipulation des Prozesses kennenlernen



Inhalt

- Überblick: Erstellung von Datenbankstatistiken
- Automatisierung
- Präferenzen für den Statistiklauf
- Diverses





Überblick: Erstellung von Datenbankstatistiken

DB-Statistiken

- Sammeln von DB-Statistiken möglich seit Oracle7 (mit Einführung CBO)
- Zunächst über ANALYZE-Kommando
- Ab Oracle8i auch mit DBMS_STATS
- Bis einschließlich Oracle9i R2 Diskrepanz:
 - Notwendigkeit von Statistiken für gute Ausführungspläne
 - Kein Automatismus in der DB!



Das ANALYZE-Kommando

- Wird u.a. benötigt für folgende Optionen:
 - VALIDATE (Strukturanalyse)
Untersuchung auf Korruptionen!
 - COMPUTE / ESTIMATE STATISTICS
Berechnung von *Chained Rows* (DBMS_STATS macht das nicht!)
 - LIST CHAINED ROWS
Nächste Stufe zur Analyse von Chained Rows
 - Freelist-Informationen
Bei ASSM-Tablespaces nicht mehr relevant!
- CBO-relevante Daten werden mit DBMS_STATS gesammelt!





Automatisierung

Automatische Statistiken seit Oracle 10g

- Scheduler Job (10g)
- Autotask-Infrastruktur (11g)
 - Neue APIs
 - Neue Views
 - Scheduler Jobs nur noch während der Laufzeit zu sehen
- Oft gehört:
 - „Ich deaktiviere den Automatismus und nutze meine eigenen Skripte!“
 - „Oracle 11g berechnet keine Statistiken – es gibt keinen Scheduler Job!“



Wo ist der Scheduler-Job zum Sammeln der Statistiken?

- Ab 11g „versteckt“ in der Autotask-Infrastruktur!
 - Views `DBA_AUTOTASK_CLIENT`,
`DBA_AUTOTASK_OPERATION`,
`DBA_AUTOTASK_WINDOW`
 - Admin-Schnittstelle: `DBMS_AUTO_TASK_ADMIN`
 - Ebenfalls wichtig: `DBMS_SCHEDULER`
 - Manueller Start des Statistiklaufs (z.B. nach Initialbefüllung einer DB):

`DBMS_AUTO_TASK_IMMEDIATE.GATHER_OPTIMIZER_STATS`



ENABLE / DISABLE der Autotask-Infrastruktur bzw. einzelner Clients

- Autotask Infrastruktur komplett:
EXEC dbms_auto_task_admin.enable
EXEC dbms_auto_task_admin.disable
- Client:
EXEC dbms_auto_task_admin.enable
('auto optimizer stats collection')

EXEC dbms_auto_task_admin.enable
('auto optimizer stats collection',
'WEDNESDAY_WINDOW')



Wann und wie wird der Statistiklauf gestartet (11g)?

- Zeitfenster: MONDAY_WINDOW, TUESDAY_WINDOW, ..., SUNDAY_WINDOW
- Erzeugen eines Einmal-Jobs für den Scheduler (Name: ORA\$AT_OS_OPT_SY_<nn>)
- View: DBA_SCHEDULER_JOB_RUN_DETAILS
- Stop des Jobs bei Ende des Zeitfensters



Welche Objekte werden analysiert?

- Noch nie analysierte
- $\frac{\#DMLs}{NUM_ROWS} * 100 \geq STALE_PERCENT$
- Default für STALE_PERCENT: 10
- Wichtige Views:
DBA_TAB_STATISTICS (NUM_ROWS,
STALE_STATS)
DBA_TAB_MODIFICATIONS (INSERTS,
UPDATES, DELETES)





Präferenzen für den Statistiklauf

Welche Parameter werden verwendet?

- ESTIMATE_PERCENT:
DBMS_STATS.AUTO_SAMPLE_SIZE

Größe der Probe abhängig von der Beschaffenheit der Daten

- 5.500 Rows, 55.000 Rows, 550.000 Rows, ...
- Achtung bei AUTO_SAMPLE_SIZE:
NUM_ROWS = SAMPLE_SIZE = COUNT(*) in
DBA_TAB_STATISTICS

- METHOD_OPT: FOR ALL COLUMNS SIZE AUTO

Histogramme bei Bedarf



Welche Parameter werden verwendet?

- **CASCADE: DBMS_STATS . AUTO_CASCADE**

Abhängige Indizes wenn erforderlich

- **NO_INVALIDATE:
DBMS_STATS . AUTO_INVALIDATE**

Invalidierung von bestehenden Cursors über Rolling Invalidation

- Hard Re-Parse kann bis zu fünf Stunden dauern! (MOS Dokument 557661.1)



Welche Parameter werden verwendet?

- `STALE_PERCENT: 10`
- `DEGREE: DBMS_STATS . AUTO_DEGREE`

Oracle bestimmt DOP

- `PUBLISH: TRUE`

Statistiken sofort wirksam; bei `FALSE` Test bzw. Verwerfen der Statistiken möglich



Welche Parameter werden verwendet?

- GRANULARITY: AUTO

Tabellen / Partitionen / Subpartitionen

ALL | GLOBAL | GLOBAL AND PARTITION |
PARTITION | SUBPARTITION

- INCREMENTAL: FALSE

Globale Statistiken mit Full Table Scan berechnen



Präferenzen für DBMS_STATS

- Beispiel:

```
BEGIN
    dbms_stats.set_table_prefs
    ('SCOTT', 'EMP',
     'NO_INVALIDATE', 'FALSE');
END;
/
```



Präferenzen für DBMS_STATS

- SET_XXX_PREFS setzt Präferenzen für existierende Tabellen, bei
 - XXX = TABLE für die angegebene Tabelle
 - XXX = SCHEMA für die Tabellen im angegebenen Schema
 - XXX = DATABASE für die Tabellen in allen Nicht-internen Schemas
- Keine Voreinstellung für Schema oder Datenbank
- Neue Tabellen werden nicht berücksichtigt!



Präferenzen für DBMS_STATS

- Globale Präferenzen mit `SET_GLOBAL_PREFS`
- Gültig für alle Objekte in der DB
 - Es sei denn, es existieren Präferenzen auf Tabellenebene
- Abfrage mit `DBMS_STATS.GET_PREFS`
...oder direkt
 - `DBA_TAB_STAT_PREFS`
 - `SYS.OPTSTAT_HIST_CONTROL$`



Empfehlungen für Präferenzen

- `NO_INVALIDATE = FALSE`
Meist ist das Neu-Parsen nicht schädlich.
- Bei großen Tabellen
`STALE_PERCENT = 5`
(oder noch kleiner)



Empfehlungen für Präferenzen

- `INCREMENTAL = TRUE`
Kein FTS bei partitionierten Tabellen für globale Statistiken
 - Benötigt mehr Platz in `SYSAUX`
 - Keine Komplettberechnung der globalen Statistiken mehr
 - Vor.: Defaults für `PUBLISH`, `ESTIMATE_PERCENT`, `GRANULARITY`
- `ESTIMATE_PERCENT = ...`
für riesige Objekte einschränken (Laufzeit)





Diverses

Erweiterte Statistiken

- Histogramme pro Spalte
- Oft falsche Berechnung der Kardinalitäten bei korrelierten Spalten
Beispiel: ADRESSEN.PLZ \equiv ADRESSEN.ORT
- Hilfe durch erweiterte Statistiken (Extended Statistics)



Beispiel

```
DECLARE
    rva1 VARCHAR2(100);
BEGIN
    rva1 := dbms_stats.create_extended_stats
        ('SCOTT', 'ADRESSEN', '(PLZ, ORT)');
END;
/
```



Systemstatistiken

- Eigenschaften der Infrastruktur (CPU, IO)
- Im normalen Betrieb sammeln, z.B.:

```
BEGIN
```

```
dbms_stats.gather_system_stats  
    ('INTERVAL', 60);
```

```
END;
```

```
/
```



Dictionary-Statistiken

- Sammeln mit:

```
BEGIN
```

```
    dbms_stats.gather_dictionary_stats;
```

```
END;
```

```
/
```

- Nicht Bestandteil der Autotask-Statistiksammlung
- Gelegentlich durchzuführen



Autotask vs. Manuelles Sammeln

- Abschalten von Autotask nur in Ausnahmefällen
- Besser: Anpassung von Zeitfenstern und Präferenzen
- Behandlung von Ausnahmen über manuelles Sammeln oder Dynamic Sampling



Nach dem Neuberechnen der Statistiken ist alles langsam!!!

- Zurück auf ...

```
BEGIN
```

```
  dbms_stats.restore_schema_stats  
    ('SCOTT', TO_TIMESTAMP(...));  
END;  
/
```

- Auch: restore_table_stats,
 restore_database_stats,...



Informationsquellen

- Oracle Database Dokumentation
 - Performance Tuning Guide
 - Reference
 - PL/SQL Packages and Types Reference



Vielen Dank für Ihre Aufmerksamkeit!

