Stabile Ausführungspläne auch in der Standard Edition

Dierk Lenz

DOAG 2019 Konferenz und Ausstellung

Herrmann & Lenz Services
Herrmann & Lenz Solutions

Herrmann & Lenz Services

Herrmann & Lenz Solutions

- 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 / Monitoring
- Herrmann & Lenz Solutions GmbH
  - Produkte: Monitoring Module, Taskzone
  - Stand auf Ebene 2 (236)

Worum geht es?

SQL Performance

- Optimizer: versucht, ***DEN*** besten Plan zu bestimmen
- Geling oft
- Aber (mit dem Ziel, Pläne weiter zu verbessern):
  - Neuberechnung von Statistiken
  - Patches, Updates, Migrationen
  - Adaptive Features
- In einigen Fällen: Umkippen von Ausführungsplänen → erhebliche Verschlechterung
- Gesucht: Möglichkeit zum Festlegen eines *guten* Plans

Festlegen von Plänen

- Umfasst diverse Techniken
  - Stored Outlines
  - SQL Plan Management
  - Hints
- Verspricht Sicherheit
- Risiko:
  - Größenverhältnisse von Tabellen ändern sich
  - Werteverteilungen ändern sich
  - Neue Versionen bieten neue Optimizer-Verfahren

➔ Optimizer kann / darf nicht reagieren!

Stored Outlines: Ganz schön lange dabei…

- Mit Oracle8i eingeführt
- Einfaches Konzept:
  - Speicherung eines Ausführungsplans zu einem SQL Statement
  - Methode: Hints
- Deprecated ab Oracle Database 11gR2 (immer noch mit 19c…)
- Oracle‘s Empfehlung: bitte stattdessen SQL Plan Management nutzen
- OK…
- Aber: Enterprise Edition Feature!!!
- Neu in 19c: mit Einschränkungen in SE2 nutzbar!

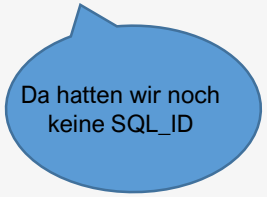
Stored Outlines

Implementierung

- Oracle-Schema `OUTLN` mit den Tabellen `OL$`, `OL$HINTS` und `OL$NODES`
- Stored Outlines zwischen Datenbanken austauschen: Export, Import

Erzeugen von Stored Outlines

- Session-Parameter
  ```
  create_stored_outlines
      = TRUE | FALSE | <category>
  ```
- Alle folgenden Parses werden Outlines
- (Vorsicht! Wenn Statement in der SQL Area → kein Parse!)
- Für einzelne Statements mit
  ```
  dbms_outln.create_stored_outline
      (<hash>, <child>, <category>)
  ```



Nutzen von Stored Outlines

- System- bzw. Session-Parameter
  `use_stored_outlines = TRUE | FALSE | <category>`
- Kurios: kein Initialisierungsparameter, d.h. funktioniert nicht in init.ora bzw. spfile.ora
- Oracle brauchte wohl ein Beispiel für einen Database Startup Trigger

Demo Stored Outlines

Der Trick mit dem Plantausch

- Situation: Plan ist gekippt
- Aufzeichnung ergibt den „falschen“ Plan
- Lösung:
  1. Statement z.B. durch Hints tweaken bis Plan OK
  2. Ebenfalls aufzeichnen
  3. Tausch der Rows zu den Outlines (`UPDATE` auf `ol$hints.ol_name`)

SQL Plan Management

SQL Plan Management

- Komplexes Framework zur Verwaltung von SQL Plan Baselines
  → Menge von akzeptierten Plänen für ein SQL Statement
- Auch schon länger dabei!
- Mit Plan Evolution: Prüfen neuer Pläne (z.B. bei neuen Versionen), dann Hinzufügen zur Baseline
- Voraussetzung für SQL Plan Management: Enterprise Edition!
- Ab 19c: Eingeschränkt auch für Standard Edition:
  - Nur eine SQL Plan Baseline pro SQL Statement
  - Plan Evolution ist deaktiviert
  - Einige weitere Funktionen nicht möglich, z.B. `dbms_spm.load_plans_from_awr`

Erzeugen von SQL Plan Baselines

- `optimizer_capture_sql_plan_baselines = TRUE`
- Einzelne Pläne aus der SQL Area:
  `dbms_spm.load_plans_from_cursor_cache (sql_id => …)`
- Migration von Stored Outlines:
  `dbms_spm.migrate_stored_outline`

Vorteile von SQL Plan Management

- Filter beim Capture von Plänen:
  dbms_spm.configure
- Filter-Parameter:
  - `AUTO_CAPTURE_SQL_TEXT`
  - `AUTO_CAPTURE_PARSING_SCHEMA_NAME`
  - `AUTO_CAPTURE_MODULE`
  - `AUTO_CAPTURE_ACTION`
- Systemparameter
  `optimizer_use_sql_plan_baselines [ = TRUE ]`
  (auch als Initialisierungsparameter!)

Nachteil

- Direkter Plantausch wie bei Stored Outlines nicht möglich
- Aktuell noch machbar:
  1. Trick mit Stored Outlines durchführen
  2. Dann die richtige Stored Outline als SQL Plan Baseline migrieren

Demo SQL Plan Management

Fazit

- Mit 19c endlich wieder ein supportetes Verfahren zum Festlegen von Plänen in der Standard Edition vorhanden
- Möglichkeit zur Migration von Stored Outlines
- ***Retter in der Not***-Feature – kein Allheilmittel
- Eher für einzelne Pläne, nicht für die gesamte Anwendung

Und sonst so in 12.2/18c/19c?

Einige Statements brauchen VIEL länger…

- OR-Optimierung bis einschließlich 12.1 mit dem alten rule-based Verfahren
- Neues Verfahren ab 12.2
- Führt bei manchen SQLs mit OR-Klauseln (auch IN-Klauseln) zu langen Laufzeiten
- Nachweis im Projekt, dass das nicht so sein muss: mit Stored Outline!
- Lösung: Abschalten der neuen OR-Optimierung:
  `"_optimizer_cbqt_or_expansion" = off`

  (MOS Note 2536570.1: *Bad Execution Plan With OR Query After Update To 12.2.0.1*)

Starten mit Pluggable Databases in 19c

- Ab 19c sind 3 PDBs pro CDB erlaubt
- *In der nächsten Version* soll die *alte Architektur* nicht mehr supported sein
- Alle Beispiele dieser Präsentation wurden mit einer PDB in einer 19c SE2 CDB erstellt

Weitere H&L Vorträge

- **Die Top Fehler beim Datenbank-Monitoring vermeiden**
  Sebastian Röhrig
  Mittwoch 9:00 Seoul
- **Boost (Not Only) My ODA: Die GI als Ressourcenmanager**
  Susanne Jahr
  Donnerstag 13:00 Seoul

Fragen & Kontakt

- Mail: dierk.lenz@hl-services.de
- Web: www.hl-services.de
- Blog: blog.hl-services.de
- Twitter: @ora1578
- Live: Hier in der Ausstellung, Stand 236

  Ebene 2 (gelb), 2. Stand links von der Rolltreppe