

Alternative Fakten - das Unbeeinflussbare beeinflussen

Stefan Winkler
merlin.zwo InfoDesign GmbH & Co. KG
Karlsruhe

Schlüsselworte

Manipulation SQL, Manipulation Execution Pläne, transparent für Applikation, Restriktionen überwinden, closed code, SQL Translation Framework, Eingriff in Deployment, Performance, Tuning, beeinflussen, ...

Einleitung

Du hast keine Chance – nutze sie! Was tun, wenn plötzliche Performanceeinbrüche oder fehlerhaftes SQL den Anwendungsbetrieb lahmlegen oder eine Migration gefährden und zudem vorgegebene Restriktionen scheinbar alle Lösungsmöglichkeiten verhindern? Im Vortrag werden verschiedene solcher Szenarien aus der Praxis mit unterschiedlich harten Restriktionen geschildert und mögliche Lösungen beschrieben. Der Referent erläutert anhand von Praxisbeispielen aus dem harten Realitätsalltag wie man bei problematischem und scheinbar fixen SQL-Code dennoch Lösungen sowohl für den Endkunden als auch seinen Lieferanten finden kann.

Der Schwerpunkt des Vortrags liegt bei der verschärften Variante mit „closed code“ und nutzt dabei unter anderem das SQL Translation Framework.

Ein typisches, sicherlich bekanntes Szenario aus dem ORACLE-Alltag,,

An einem nicht ganz so schönen Montagmorgen läuft die zentrale Anwendung nur noch in Zeitlupe und alle Beteiligten versichern glaubhaft, dass sie rein gar nichts geändert haben. Die eigene IT-Abteilung kann jedoch aus dem AWR schnell das verantwortliche Statement ermitteln, prüft auf etwaige Änderungen des Execution Plan, der Indizierung, Parametrisierung usw. Gemeinsam mit dem gerade verfügbaren externen ORACLE-Experten erarbeitet sie ein getunttes Statement und stimmt den Einbau mit dem Softwarehersteller für das Produktiv-Deployment am Abend ab. Unrealistisch?

Natürlich!

Das Oracle RDBMS ist ein sehr mächtiges System mit vielen Möglichkeiten zur Einflussnahme. Manche Bereiche lassen sich bei Problemen nur mit viel Aufwand und Einverständnis aller Beteiligten verändern (z.B. ORACLE-Versionen oder Patches); bei anderen sind der Aufwand und/oder die Restriktionen dagegen wesentlich kleiner (z.B. Statistiken, init.ora-Parametrisierung).

Dieser Vortrag schildert in der Praxis angetroffene Situationen und dafür gefundene Lösungen bei Performance- und Deploymentproblemen, sowohl für den kooperativen Fall (man kommt an den Code heran und darf ihn ändern), als auch für die verschärfte Variante, dass man an die Ursache (z.B. den Source-Code) NICHT heran kommt und/oder ihn NICHT ändern darf.

Die Frage ist also: Welche Restriktionen bestehen und wie kann man sie brechen oder zumindest umgehen, so dass eine akzeptable Lösung sowohl für den Endkunden als auch den Software-Lieferanten und die Betriebsverantwortlichen (typischerweise die IT-Abteilung) gefunden werden kann?

Alternative Fakten für die Ausführung von SQL-Befehlen

Der erste Teil des Vortrags konzentriert sich auf die Beeinflussung der Art und Weise wie das SQL ausgeführt wird. Zentrales Element ist dabei der Execution Plan und die Einflussnahme auf seine Bildung. Der häufigste Auslöser für den Wunsch zur Beeinflussung ist bei diesem Bereich der alternativen Fakten die Performance, die bekanntlich am wirkungsvollsten durch Indizierung, Parametrisierung (init.ora) und Statistiken beeinflusst werden kann.

Index-Probleme bei Migrationen sowie im Betrieb gehören hierbei sicherlich zu den häufigsten Themen. Für die gängigsten Aufgaben gibt es in der ORACLE Dokumentation und im Web viele HowTo's, z.B. für

- Größe und I/O: Basic/Advanced Compression
- Selektivität: Reverse Columns, Column Compression
- Performance: Partitionierung, optimizer_%-Parameter...

Was aber tun, wenn ein Softwarehersteller sein eigenes „Data Dictionary“ mit den von ihm erwarteten und gepflegten Objekten führt? Und noch verheerender, wenn er bei der Migration sein Data Dictionary überprüft, alles „Ungewollte = nicht eigene“ gnadenlos eliminiert und dieser Code nicht deaktiviert werden kann oder darf? Der Vortrag erläutert wie bei einer Kundenmigration für dieses Problem eine pragmatische Lösung gefunden wurde.

Im Vortrag wird mit alternativen Fakten eine Lösung für ein weiteres Kundenszenario vorgestellt: Tuning der Abfrage- und DML-Zugriffe auf eine sehr große Tabelle mit asymmetrisch verteilten Daten – ohne Änderungen am Applikationscode und ohne Partitioning Option! Die vorgestellte Lösung ist gemäß Kundenvorgabe 100% transparent für die Anwendung und benötigt nur Features der Standard Edition (u.a. View und INSTEAD OF Trigger). Wieder einmal überrascht hier der Optimizer beim Bau des „passenden“ Execution Plans.

Im nächsten Beispiel aus der Projektpraxis zeigt der Vortrag wie in einem sehr restriktiven Umfeld alternative Fakten für einen Execution Plan durch Oracle SQL-Profile geschaffen wurden.

Alternative Fakten für den SQL-Code

Nachdem im ersten Teil des Vortrags die Beeinflussung der Art und Weise wie ein gegebener SQL-Befehl ausgeführt wird, gezeigt wurde, wird im zweiten Teil die Manipulation des SQL-Befehl an sich – also des Codes! – ausführlich thematisiert. Hier bietet ORACLE seit einigen Versionen weitgehend unbekannte Möglichkeiten. Das SQL Translation Framework als eine dieser Möglichkeiten; es wird konzeptionell und basierend auf einem realen Einsatz in einem Kundenprojekt in vereinfachter Form vorgestellt. Es wird aufgezeigt welche Voraussetzungen zu schaffen sind, wie die Durchführung abläuft und welche Möglichkeiten es grundsätzlich gibt – wobei auch Monitoring, Tracing, das Handling von Bindevariablen sowie Code Injection thematisiert werden – aber auch welche Restriktionen existieren.

Kontaktadresse:

Stefan Winkler
merlin.zwo InfoDesign GmbH & Co. KG
Elsa-Brändström-Straße 14
D-76228 Karlsruhe

Telefon: +49 (0) 721-132 096-41
Fax: +49 (0) 721-132 096-99
E-Mail stefan.winkler@merlin-zwo.de
Internet: www.merlin-zwo.de

