

Oracle-DB

Beeinflussen der Ausführungspläne von SQL-Statements ohne Code-Anpassung

Verschiedene Verfahren und ihre Unterstützung durch Panorama



Otto Group Solution Provider Dresden GmbH

www.osp.de



Gründung:

März 1991

Muttergesellschaft:

OTTO Group

Standorte:

Dresden, Hamburg, Burgkunstadt, Taipeh, Bangkok

Mitarbeiterzahl:

Ca. 250

Geschäftsführer:

Dr. Stefan Borsutzky, Norbert Gödicke, Jens Gruhl

zur Person



Peter Ramm

Teamleiter strategisch-technische Beratung bei OSP Dresden

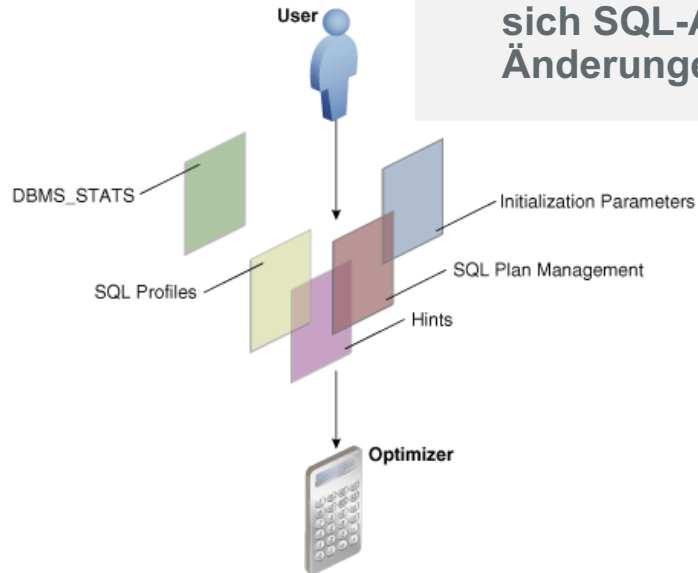
> 20 Jahre Historie in IT-Projekten

Schwerpunkte:

- Entwicklung von OLTP-Systemen auf Basis von Oracle-Datenbanken
- Architektur-Beratung bis Trouble-Shooting
- Performance-Optimierung bestehender Systeme

Motivation

- Einfluss auf problematische SQL-Ausführung erfordert oft ein Ausrollen der angepassten Software
- In kritischen Produktions-Szenarien ist teils eine schnellere Lösung nötig.
- Dieser Vortrag demonstriert, mit welchen Verfahren der Oracle-DB sich SQL-Ausführungen auch ohne Ausrollen von Software-Änderungen ad hoc beeinflussen lassen.



Dabei werden im folgenden betrachtet:

- SQL Plan Baselines
- SQL Profiles
- SQL Patches
- SQL Translations

Verfahren zur Beeinflussung des Optimizers

Verfahren	Wirkungsweise	Erforderliche Lizenz
SQL Plan Baseline	Vorgabe des Plan-Hash-Wertes des zu verwendenden Ausführungsplans	EE (+Tuning Pack für Erstellung per Panorama aus AWR-Daten)
SQL Profile	Injizierung von Optimizer-Hints	EE + Diagnostics and Tuning Pack
SQL Patch	Injizierung von Optimizer-Hints ähnlich SQL-Profile	Keine (auch anwendbar in Standard Edition)
SQL Translation	Ersetzen des vollständigen SQL-Textes	EE

Panorama für Oracle: freies Analyse-Tool

Freies Werkzeug für Performance-Analyse
von Oracle-DB

Gehostet für Nutzung innerhalb otto group:

<http://sbfappserver.ov.otto.de:8080/Panorama>

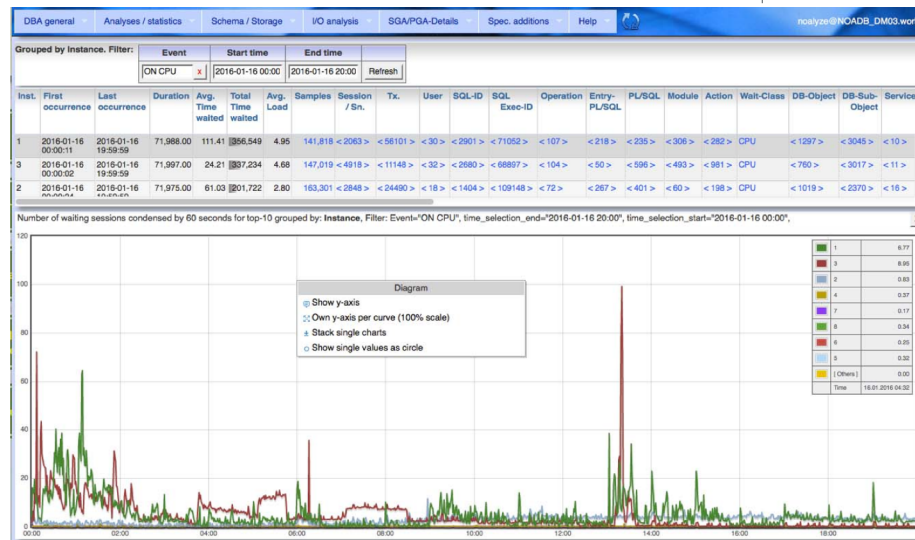
<http://panorama.osp-dd.de:8080/Panorama>

https analog auf Port 8443

Weitere Doku + Download:

<http://rammpeter.github.io>

<https://hub.docker.com/r/rammpeter/panorama>



- „Leichte“ Machbarkeit der Analysen mittels GUI-Workflow, für DBA als auch gezielt für Entwickler
- Senken der Hemmschwelle, Problemen tatsächlich detailliert auf den Grund zu gehen
- Identifikation der konkrete Ursachen für unzureichende Applikations-Performance bei Datenzugriffen
- Aufbereitung komplexer Zusammenhänge der DB ohne tiefes Insider-Wissen

SQL Plan Baseline

- Erlaubt es, einen konkreten Ausführungsplan verbindlich vorzugeben, wenn ein syntaktisch identisches SQL-Statement ausgeführt wird (SQL-ID Hash).
- Es wird hierbei nicht der konkrete Ausführungsplan vorgegeben, sondern der Plan Hash Value des zu verwendenden Plans. D. h., der Optimizer muss selbst diesen Plan mit dem zu erzielenden Plan Hash Value ermitteln können.

Support durch Panorama:

- Generieren von PL/SQL-Code zum Erzeugen einer SQL Plan Baseline für einen in Vergangenheit aktiven konkreten Ausführungsplan eines SQL
- Generierung ausführbar über Button „SQL Plan Baseline“ in historischer SQL-Detail-Ansicht

Vorteil in der Anwendung:

- Beim Auftreten eines geänderten problematischen Ausführungsplanes kann über Vergleich der bisherigen Ausführungen in AWR-Historie ein evtl. besserer Plan identifiziert werden
- Durch Generieren der Baseline-Erzeugung auf Basis des besseren Plans kann das Problem erstmal in Minuten gefixt werden ohne das SQL überhaupt inhaltlich verstanden zu haben

SQL Plan Baseline: generiertes Script (Extrakt)

```
DECLARE
    cur          sys_refcursor;
    hit_count    PLS_INTEGER;
BEGIN

    -- Create new SQL Tuning Set (STS)
    DBMS_SQLTUNE.CREATE_SQLSET(sqlset_name => 'PANORAMA_STS',
                               description => 'Panorama: SQL Tuning Set for SQL Plan Baseline');

    -- Populate STS from AWR, using a time duration when the desired plan was used
    OPEN cur FOR
        SELECT VALUE(P)
        FROM TABLE(
            DBMS_SQLTUNE.Select_Workload_Repository(Begin_Snap=>116673, End_Snap=>116769,
                                                    Basic_Filter=>'sql_id = ''7fbk69tj7ak43'',
                                                    attribute_list=>'ALL')
            ) p;
        DBMS_SQLTUNE.LOAD_SQLSET( sqlset_name=> 'PANORAMA_STS', populate_cursor=>cur);
    CLOSE cur;

    -- Load desired plan from STS as SQL Plan Baseline, Filter explicitly for the plan_hash_value here
    hit_count := DBMS_SPM.LOAD_PLANS_FROM_SQLSET(sqlset_name => 'PANORAMA_STS',
                                                  Basic_Filter=>'plan_hash_value = ''3026395079''');

END;
/
```


- Erlaubt es, bei Ausführung eines über SQL-ID identifizierten SQLs weitere Optimizer-Hints von außen zu injizieren, die bei der Ermittlung des Ausführungsplanes berücksichtigt werden.
- Über diverse SQL-Tuning-Funktionen des Oracle Enterprise Managers (bzw. neu Cloud Control) können SQL-Profiles erzeugt werden.

Support durch Panorama:

- Das Generieren von SQL Profiles wird durch Panorama nicht unterstützt, da die identische Funktion auch durch SQL Patches erreicht werden kann ohne Limitierung bzgl. Edition und Option Pack

- SQL-Patches (verfügbar ab Release 11.1) erlauben es, bei Ausführung eines über SQL-ID identifizierten SQLs weitere Optimizer-Hints von außen zu injizieren, die bei der Ermittlung des Ausführungsplanes berücksichtigt werden.
- SQL-Patches können ohne weitere Lizenz genutzt werden auch in Standard Edition.

Support durch Panorama:

- Generieren von PL/SQL-Code zum Erzeugen eines SQL Patch
- Dieser PL/SQL-Code muss nach Generierung um die gewünschten Optimizer-Hints erweitert werden
- Generierung ist ausführbar über Button „SQL Patch“ in historischer und aktueller SQL-Detail-Ansicht
- Unterschiedliche PL/SQL-API je DB-Release werden durch Generator berücksichtigt

SQL Patch: generiertes Script

Durch Panorama generiertes Script:

```
BEGIN
sys.DBMS_SQLDiag_Internal.i_create_patch(
    sql_text      => 'SELECT * FROM Hugo h WHERE h.Filter_ID = :A1',
    hint_text     => '< my personal hint>',
    name          => 'Panorama-Patch 7fbk69tj7ak43',
    description   => 'My personal description for patch'
);

END;
/
```

Mit Hint angepasstes Script:

```
BEGIN
sys.DBMS_SQLDiag_Internal.i_create_patch(
    sql_text      => 'SELECT * FROM Hugo h WHERE h.Filter_ID = :A1',
    hint_text     => 'INDEX(@SEL$1 h@SEL$1, IDX_Hugo_Neu)',
    name          => 'Panorama-Patch 7fbk69tj7ak43',
    description   => 'Soll den genannten Index verwenden'
);

END;
/
```

Für Positionierung der Hints in komplexen SQLs Query-Block-Name in Hint angeben

- Das SQL Translation Framework erlaubt ab Version 12.1 das Hinterlegen von alternativen SQL-Texten zu einem SQL-Statement.
Vor der Ausführung des SQLs wird der SQL-Text ersetzt durch den hinterlegten alternativen SQL-Text.
- Bedingung ist, dass das alternative SQL die selbe Result-Struktur besitzt und die selben Bindevariablen unterstützt wie das originale SQL.

Support durch Panorama:

- Generieren von PL/SQL-Code zum Erzeugen einer SQL Translation über den Button "SQL-Translation" der SQL-Detail-Anzeige (aktuell oder historisch)

Vorteil in der Anwendung:

- Diese Funktion erlaubt eine totale Einflussnahme auf das ausgeführte SQL.
Nicht nur, dass dem SQL Optimizer-Hints zugefügt werden können, es können auch Ergebnisspalten ausgetauscht werden, JOIN-Beziehungen entfernt oder hinzugefügt werden, aus anderen Quellen selektiert werden etc. .

SQL Translation: Generiertes Script

- Das generierte Script ist etwas komplexer als bei den vorgenannten Varianten und enthält drei Abschnitte:
 - SYSDBA grantet User die Erzeugung von Translation Profiles
 - User erzeugt Translation Profile mit altem und neuem SQL-Test
 - SYSDBA erzeugt Logon-Trigger, der die Nutzung der Translation in User-Session aktiviert
- Im generierten Script muss die neue Variante des SQL entsprechend angepasst werden

Risiko bei Verwendung der genannten Verfahren

- Da die Zuordnung der Verfahren zu einem SQL-Statement auf der SQL-ID bzw. dem SQL-Text basiert, geht diese Beziehung bei Änderung oder Erweiterung der SQL-Syntax verloren
- Daher sollten diese Maßnahmen eher als Quick-Fix bis zum nächsten Software-Rollout gesehen werden und nicht als langfristige Lösung

Support durch Panorama:

- Die Existenz eines der vier Verfahren für ein SQL-Statement wird in Panorama in der SQL-Detail-Ansicht signalrot angezeigt.
Ohne diese Information besteht das Risiko der Konfusion darüber, wie der Optimizer zu dem konkret verwendeten Ausführungsplan kommt
- Menü „DBA/SGA-Details“ / „SQL plan management“ gibt einen Überblick über alle existierenden Maßnahmen inkl. ihrer wirklichen Nutzung

Letztes Wort

Panorama greift nur lesend auf die Datenbank zu und benötigt keine eigenen PL/SQL-Objekte.

Ihr könnt die Funktionen also ohne Risiko testen und verstehen. Probiert es gern aus.

Beschreibung inkl. Download-Link:

Docker-Image

Blog zum Thema:

<http://rammpeter.github.io/panorama.html>

<https://hub.docker.com/r/rammpeter/panorama>

<http://rammpeter.blogspot.com>

Vielen Dank für Euer Interesse

Fragen ?

Otto Group Solution Provider (OSP) Dresden GmbH

Freiberger Str. 35 | 01067 Dresden

T +49 (0)351 49723 0 | F +49 (0)351 49723 119

[osp.de](https://www.osp.de)

Backup

Weitere Links zum Thema:

- [Einführung in Oracle-Performance-Analyse mit Panorama](#)
- [Ulrike Schwinn: Ausführungspläne beeinflussen mit SQL Plan Management](#)
- [Oracle white paper: SQL Plan Management in Oracle Database 11g](#)
- [Oracle white paper: SQL Plan Management in Oracle Database 12c Release 2](#)
- [SQL Tuning Guide 12.2: Influencing the Optimizer](#)
- [SQL Tuning Guide 12.2: SQL Controls and Plan Baselines](#)