



SQLT und SQLHC

Performance-Analyse mit Bordmitteln

Uwe Küchler, Senior Consultant

OPITZ CONSULTING Deutschland GmbH

(NL Bad Homburg v.d.H.)

E-Mail: uwe.kuechler@opitz-consulting.com

Internet: www.opitz-consulting.com



DOAG-Konferenz Nürnberg, 21.11.2013



Mission

Wir entwickeln gemeinsam mit allen Branchen Lösungen, die dazu führen, dass sich diese Organisationen besser entwickeln als ihr Wettbewerb.

Unsere Dienstleistung erfolgt partnerschaftlich und ist auf eine langjährige Zusammenarbeit angelegt.

Leistungsangebot

- Business IT Alignment
- Business Information Management
- Business Process Management
- Anwendungsentwicklung
- SOA und System-Integration
- IT-Infrastruktur-Management

Märkte

- Branchenübergreifend
- Über 600 Kunden



Eckdaten

- Gründung 1990
- 400 Mitarbeiter
- 9 Standorte



Zur Person

- **Generation C=64**
- **Seit über 25 Jahren in der IT tätig**
- **1997-2000 bei Oracle Deutschland**
- **Seither durchgehend Oracle-Berater, im DBA- und Entwicklungs-Umfeld, Tutor**
- **Seit 09/2013 bei OPITZ CONSULTING**
- **Blog- und Buchautor**
- **Performance als „Steckenpferd“**



Agenda

1. Einsatzszenario

2. Wozu braucht die Welt noch ein Tool?

- Exkurs: SQL-Performance-Analyse
- Welche Tools gibt es am Markt?
- Welche Lücke füllen SQLT+SQLHC?

3. Funktionsweise von SQLT

4. Funktionsweise von SQLHC

5. Fallbeispiele

Einsatzszenario

- **Anwender: „Die Anwendung ist langsam!“**
 - Idealfall: Anwender grenzt Aussage auf einen Vorgang ein
- **Anwendungsverantwortlicher: „Die DB ist langsam!“**
- **DBA muss nun Beweis oder Gegenbeweis erbringen**
 - Allgemeines Performance-Problem?
 - Problem mit SQL des spezifischen Anwendungsvorgangs?
 - Bei fehlender Eingrenzung: Top-n SQL
- **„Gestern lief es noch“**
 - Verursacher ist mittlerweile identifiziert, aber
 - Gründe für schlechte Performance noch unklar.
 - Ggf. ergänzt durch „wir haben nichts am System geändert!“
- **Nun beginnt eine mehr oder weniger lange Detektivarbeit...**

SQL-Performance-Analyse

■ Etablierte Top-Down-Methodik:

1. Auf welche Systemressourcen wurde am längsten gewartet?
2. Was war für diesen Ressourcenverbrauch am meisten verantwortlich?
3. Warum kam es jeweils zu diesen Wartezeiten und lassen sich diese verkürzen?

■ Viele verfügbare Tools

- APM (AppDynamics, Foglight, New Relic, ...)
- DB-zentrierte Tools (OEM, Spotlight, ...)
- Bordmittel
 - Diagnostics + Tuning Pack (→ ASH + AWR, **lizenzpflichtig!**)
 - Statspack (funktioniert auch noch in 12c)
 - Autotrace
 - SQL- und Optimizer-Trace (Events 10043 und 10056)
 - Data Dictionary

Betrachtung vorhandener Tools

■ APM

- Alarmiert und grenzt Problem ein
- Beantwortet nicht, *warum* es zu dem Problem kam
- Zusätzliche Kosten

■ ASH, AWR, Statspack

- Bietet umfangreiche, historische Daten
- Gute Top-Down-Sicht in den Berichten
- Beantwortet i.d.R. nicht, *warum* es zu dem Problem kam
- Hohe Lizenzkosten

■ Real Time SQL Monitoring

- Detaillierte Sicht auf SQL-Ablauf
- Gegenüberstellung Optimizer-Schätzung und tatsächlicher Ablauf
- Beantwortet das „wo“ und „wie“, aber nicht unbedingt das „*warum*“
- Hohe Lizenzkosten, weil Teil des Tuning Packs

Betrachtung vorhandener Tools (2)

■ Autotrace

- SQL muss live ausgeführt werden; keine historische Betrachtung
- Läuft i.d.R. nicht im Applikations-Kontext

■ SQL-Trace

- Jeder einzelne Schritt der SQL-Ausführung nachvollziehbar
- Muss zur Laufzeit durchgeführt werden; keine historische Betrachtung
- Große Tracefiles + Overhead verbieten meist Einsatz in Produktion

■ Data Dictionary

- Äußerst detaillierte Sicht auf Performance-Metriken, Objekt-Statistiken, Optimizer-Einstellungen, Session-Konfiguration, SQL Area, ...
- Mit Zusatzlizenz auch längere, historische Betrachtung („DBA_HIST%“)
- Für manuelle Recherche Expertenwissen nötig
- Manuelle Recherche kann sehr zeitaufwendig werden
- Viele Möglichkeiten, ein wichtiges Indiz zu übersehen.

Wozu SQLT / SQLHC?

- **Automatisierung der „Detektivarbeiten“**
- **Plötzliche Änderungen am Ausführungsplan**
- **Änderungen an den Optimizer-Statistiken der vom SQL benutzten Objekte**
- **Grundsätzlich schlechte oder fehlende Optimizer-Statistiken**
- **Nicht mehr benutzte (z.B. weil invalide oder gelöschte) Indizes**
- **Aufruf des SQLs mit außergewöhnlichen Bind-Werten**
- **Hinweise auf versionsspezifische Bugs**

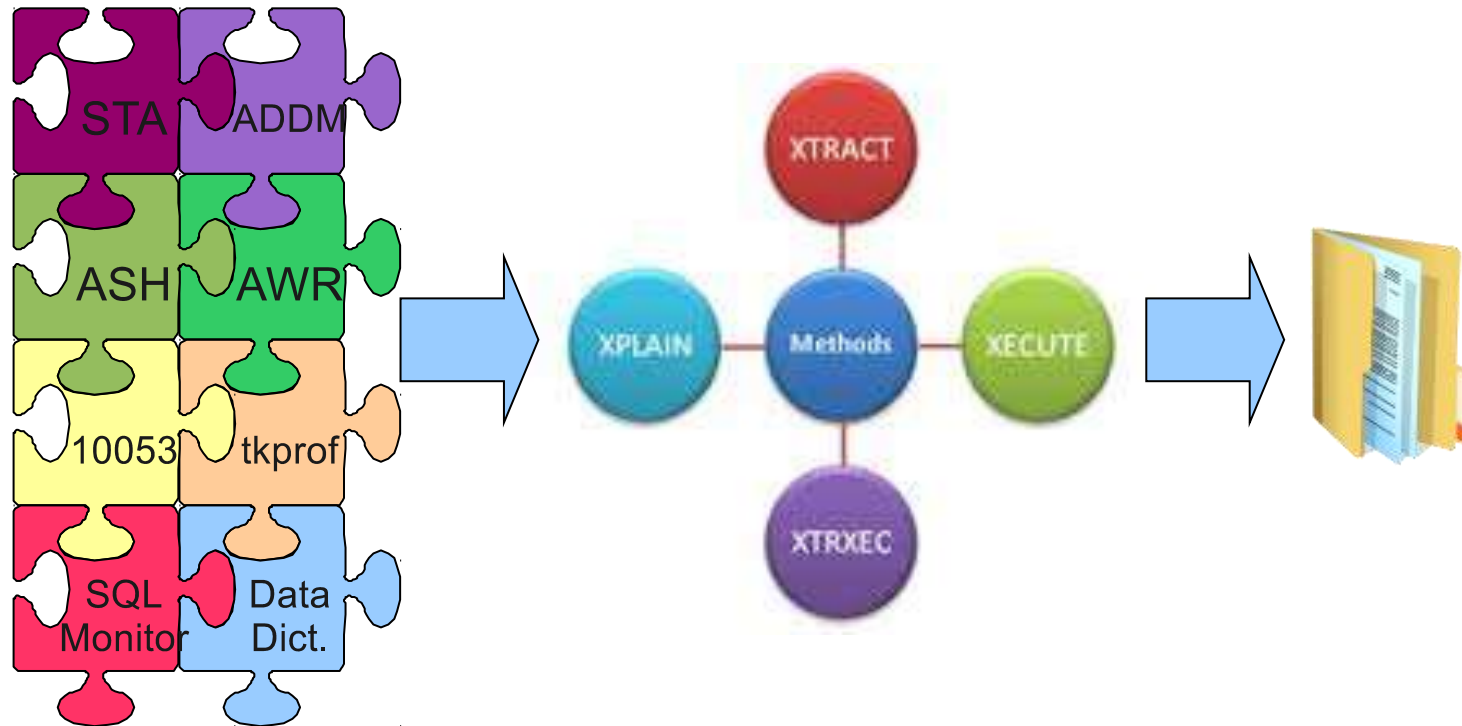
3

Funktionsweise von SQLT

Was ist SQLT?

- **Geschrieben ab 2001 von Carlos Sierra (Oracle Support)**
- **Begann als persönliche Sammlung von Skripten zur Erleichterung der Arbeit beim Troubleshooting**
 - Ursprünglicher Name: „SQLTXPLAIN“
- **Ab ca. 2010 offiziell vom Arbeitgeber unterstützt**
 - Einerseits mit Arbeitszeit für die Weiterentwicklung
 - Offizielles Tool zum Erstellen von Test Cases für Oracle Support
- **Seither teils wöchentliche Updates; aktuelle Version: 12.1.03**
- **Verantwortung 2013 übergeben an Mauro Pagano (Oracle)**

Funktionsweise von SQLT



Funktionsweise von SQLT

- **Sammlung von SQL-Skripten, PL/SQL-Packages und Repository-Tabellen in eigenem Schema**
- **Sammelt Informationen zum Tuning eines SQL-Statements ein, wahlweise aus historischen Informationen oder durch Ausführung des Statements**
- **Führt über 100 Health Checks rund um das untersuchte Statement aus**
- **Kann aus einem Anwendungs-Schema heraus ausgeführt werden**
- **Läuft auf Single Instance, RAC und Exadata**
- **Läuft unter Oracle 8i ++**

Funktionsweise von SQLT

- **Wenn Tuning- oder Diagnostics-Pack lizenziert sind:**
 - Wird SQL Tuning Advisor (STA) für das Statement ausgeführt.
 - Werden ASH- und SQL-Monitor-Berichte erzeugt.
 - Können SQL Tuning Test Cases extrahiert werden, die dann in einem Testsystem zu eingehenderen Analysen verwendet werden können.
 - SQLC TC und TCX
 - 11g Test Case Builder (TCB)
- **Beschleunigt den SQL-Tuning-Prozess durch Automatisierung von ansonsten manuell ausgeführten Recherchen.**

Funktionsweise von SQLT: Installation / Deinstallation

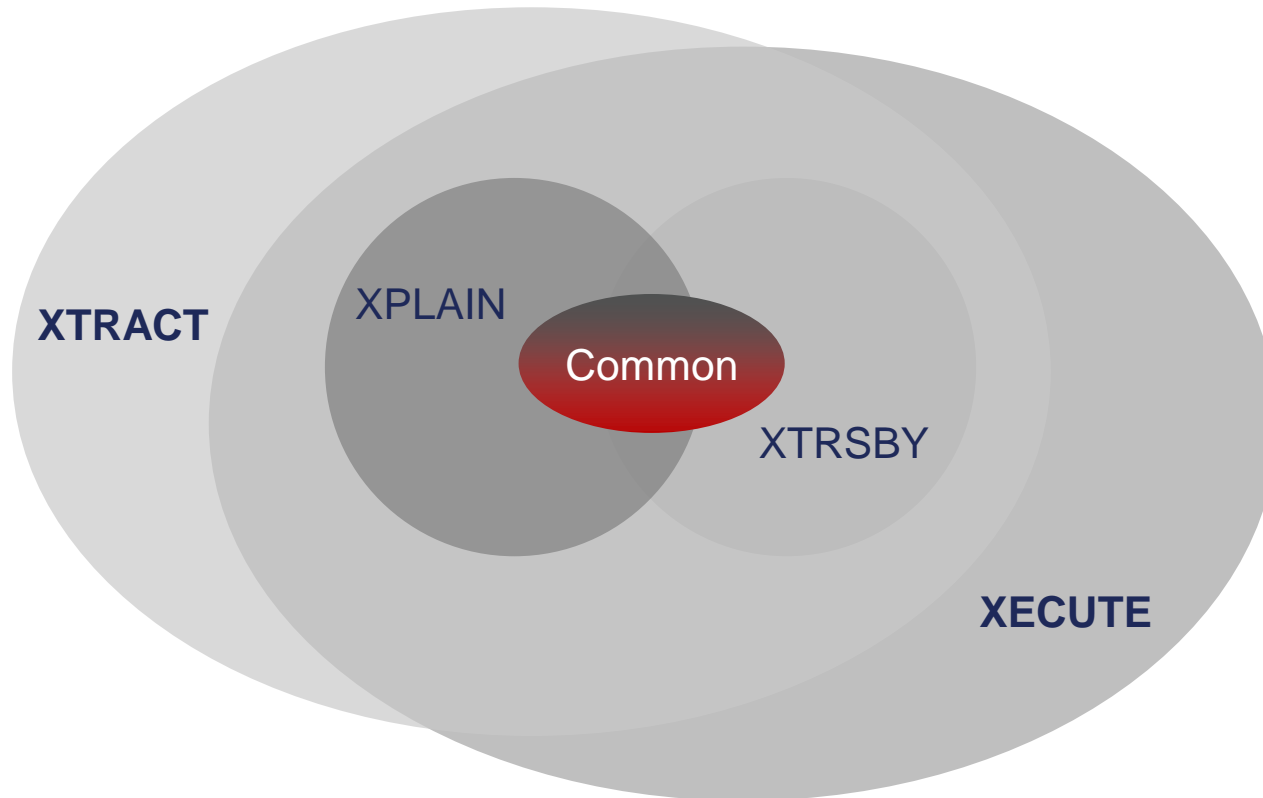
- **Download:** Doc ID [215187.1](#)
- **Als SYS:**
`sqlt/install/sqdrop.sql`
- **Als SYS:**
`sqlt/install/sqcreate.sql`
- **Eingabewerte:**
 - Passwort
 - Default Tablespace
 - Temp Tablespace
 - Benutzerkonto, das SQLT ausführen darf
 - Lizenzierte Optionen (**T**uning, **D**iagnostics oder **N**one).



Funktionsweise von SQLT: Methoden

Methoden	Script	Features
SQLT XTRACT	sqlt/run/sqltxtract.sql	<ul style="list-style-type: none"> <input type="checkbox"/> Am häufigsten verwendete Methode <input type="checkbox"/> Eingabewert: SQL_ID <input type="checkbox"/> SQL-Statement wird nicht ausgeführt
SQLT XECUTE	sqlt/run/sqltxecute.sql	<ul style="list-style-type: none"> <input type="checkbox"/> Eingabe: Name eines Scripts mit <ul style="list-style-type: none"> ✓ ALTER SESSION -Anweisungen (wenn nötig) ✓ Binds mit Wertzuweisungen (wenn nötig) ✓ SQL-Statement <input type="checkbox"/> SQL-Statement wird ausgeführt
SQLT XTRXEC	sqlt/run/sqltxtrxecsqli	<ul style="list-style-type: none"> <input type="checkbox"/> Kombiniert XTRACT und XECUTE <input type="checkbox"/> Eingabewert: SQL_ID
SQLT XPLAIN	sqlt/run/sqltxplain.sql	<ul style="list-style-type: none"> <input type="checkbox"/> Eingabe: Dateiname mit SQL-Statement <input type="checkbox"/> Wenn SQL Binds enthält: <ul style="list-style-type: none"> <input type="checkbox"/> Unverändert lassen, ODER mit Literalen vom gleichen Datentyp ersetzen
SQLT XTRSBY	sqlt/run/sqltxtrsbysqli	<ul style="list-style-type: none"> <input type="checkbox"/> Für Read-only-Datenbanken <input type="checkbox"/> Ausführung auf Primary, Verbindung zu Stdbys. <input type="checkbox"/> Wie XTRACT <input type="checkbox"/> Eingabe: SQL_ID und DBLINK

Funktionsweise von SQLT: Methoden



Funktionsweise von SQLT: Bericht (Hauptseite)

215187.1 SQLT XTRACT 11.4.4.7 Report: sqlt_s92947_main.html

Global

- Observations
- SQL Text
- SQL Identification
- Environment
- CBO Environment
- Fix Control
- CBO System Statistics
- DBMS_STATS Setup
- Initialization Parameters
- NLS Parameters
- I/O Calibration
- Tool Configuration Parameters

Cursor Sharing and Binds

- Cursor Sharing
- Adaptive Cursor Sharing
- Peeked Binds
- Captured Binds

SQL Tuning Advisor

- STA Report
- STA Script

Plans

- Summary
- Performance Statistics
- Performance History (delta)
- Performance History (total)
- Execution Plans

Plan Control

- Stored Outlines
- SQL Profiles
- SQL Plan Baselines

SQL Execution

- Active Session History
- AWR Active Session History
- SQL Statistics
- SQL Detail ACTIVE Report
- Monitor Statistics
- Monitor ACTIVE Report
- Monitor HTML Report
- Monitor TEXT Report
- Segment Statistics
- Session Statistics
- Session Events
- Parallel Processing

Tables

- Tables
- Statistics
- Statistics Versions
- Modifications
- Properties
- Physical Properties
- Constraints
- Columns
- Indexed Columns
- Histograms
- Partitions
- Indexes

Objects

- Objects
- Dependencies
- Fixed Objects
- Fixed Object Columns
- Nested Tables
- Policies
- Audit Policies
- Tablespaces
- Metadata



Funktionsweise von SQLT: SQLT Test Case

- **Sammlung von DML und Statistiken, um ein SQL unter gleichen Bedingungen in einer Testumgebung auszuführen**
 - Enthält alles benötigte DML
 - Enthält alle benötigten Objekt-Statistiken
 - Enthält alle benötigten System-Statistiken
 - Setzt alle benötigten Parameter
 - Keine Inhalte notwendig!
 - Einfacher Import ins Zielsystem per Script „xpress.sql“ (als SYS)
- **Ermöglicht**
 1. Reproduzieren des Ausführungsplans der Quelle
 2. Anpassen der Optimizer-Einstellungen, um das Verhalten zu beobachten
 3. Anpassen der System-Einstellungen, um das Verhalten zu beobachten
 4. Anpassen der Session-Einstellungen, um das Verhalten zu beobachten
 5. Anpassen von Objekteigenschaften, um das Verhalten zu beobachten



4

Funktionsweise von SQLHC

Funktionsweise von SQLHC

■ Ein einzelnes Script (sqlhc.sql)

- Download unter Document ID [1366133.1](#)
- Erfordert keine Installation
- Kann daher sofort auf eine Datenbank losgelassen werden
- Hinterlässt keinen „Fußabdruck“ in der DB
- Ausführen als SYS (niedriger privilegierter User möglich)

■ Entspricht Methode „XTRACT“ von SQLT

- SQL wird *nicht* ausgeführt
- Gegenüber XTRACT *kein* STA, ADDM, 10046
- Aber, falls verfügbar bzw. lizenziert:
 - Historische Ausführungspläne aus AWR
 - Historische Binds
 - SQL-Monitor-Bericht (Flash-Version)

Funktionsweise von SQLHC: Bericht (Hauptseite)

1366133.1 SQLHC 11.4.3.7 Report:

sqlhc_V1122_host01_11.2.0.2.0

155923.html

Tables Summary

#	Table Name	Owner	Num Rows	Table Sample Size	Indexes	Avg Index Sample Size	Table Columns	Columns with Histogram	Avg Column Sample Size
1	CUSTOMER	QTUNE	10751	10751	4	10751	5	2	6405
2	ORDER_LINE	QTUNE	239152	4933	4	178042	8	0	2932
3	PART	QTUNE			2		6	0	
4	SALES_ORDER	QTUNE	33895	33895	3	33895	5	1	28204

Tabellen und Statistiken, z.B. zum Vergleichen zwischen Test und Prod.

Observations

#	Type	Name	Observation	More
1	CBO PARAMETER	DB_FILE_MULTIBLOCK_READ_COUNT	CBO initialization parameter "db_file_multiblock_read_count" with a per	Review the correctness of this non-default value "8". Unset this parameter unless there is a strong reason for keeping its current value. Default value is "98" as per V\$SYS_OPTIMIZER_ENV.
2	CBO PARAMETER	DB_FILE_MULTIBLOCK_READ_COUNT	"8" overriding its	The default value of this parameter is a value that corresponds to the maximum I/O size that can be performed efficiently. This value is platform-dependent and is 1MB for most platforms. Because the parameter is expressed in blocks, it will be set to a value that is equal to the maximum I/O size that can be performed efficiently divided by the standard block size.
3	CBO PARAMETER	HASH_AREA_SIZE		Review the correctness of this non-default value "1048576". Unset this parameter unless there is a strong reason for keeping its current value. Default value is "131072" as per V\$SYS_OPTIMIZER_ENV.

Handlungsempfehlungen und Links zu MOS

Kategorien von Erkenntnissen (Init-Parameter, CBO-Einstellungen, Statistiken, u.v.m.)

Der SQL-Text des untersuchten Statements findet sich am Ende des Berichts.

5

Praxisbeispiele

Literatur

- **SQLTXPLAIN Download und Doku: Doc ID [215187.1](#)**
- **SQLHC Download und Dokumentation: Doc ID [1366133.1](#)**
- **SQLT FAQ: MOS Doc ID [1454160.1](#)**
- **Webinar-Aufzeichnungen: Doc ID [740966.1](#)**
 - Navigieren zu „Oracle Database“ – „Archived 2012“
- **Blog von Carlos Sierra, dem Schöpfer von SQLT+SQLHC: <http://carlos-sierra.net/>**
- **Stelios Charalambides: Oracle SQL Tuning with Oracle SQLTXPLAIN, Apress 2013, ISBN13: 978-1-4302-4809-5**