

Erhöhung der Manageability durch SQL-Profile

Ein Erfahrungsbericht

Inhalt

1. Problemstellung
2. Der SQL-Tuning-Advisor (STA)
3. Anlegen und Implementieren von SQL-Profilen
4. Alternative Stored Outlines
5. Beispiele
6. Best Practice
7. Fazit

Problemstellung

Der DBA sieht sich oft in der Situation, daß er Performanceprobleme diagnostiziert, ja sogar die Lösung kennt, diese aber nicht anwenden kann, weil er keine SQL-Statements und/oder Initialisierungsparameter ändern kann bzw. darf. Auf die einen hat er keinen Zugriff, bei Änderung der anderen verliert er unter Umständen den Support. Besonders bemerkbar macht sich das Problem bei Release-Upgrades der Datenbank.

Beispiel: Migration Oracle eBusiness Suite 11.5.9 von Datenbank 9.2 nach 10.2

- Ø Kontraproduktive Hints: ordered, rule etc.
- Ø Vorgeschriebene suboptimale Parameterwerte: z.B. CURSOR_SHARING, OPTIMIZER_INDEX_COST_ADJ

Frage: Kann der SQL Tuning Advisor hier helfen?

Der SQL Tuning Advisor

- Ø Neu in Oracle 10g
- Ø Bestandteil des Tuning Packs (separat zu lizenzieren!!!)
- Ø Funktionsweise: **Cost Based Optimizer** im Tuning Mode (comprehensive mode) = mit vom Anwender konfigurierbarem Zeitlimit

Bedienbar über:

- Ø Enterprise Manager (Database oder Grid Control)
- Ø Package DBMS_ADVISOR
- Ø Package DBMS_SQLTUNE
- Ø Skript `$ORACLE_HOME/rdbms/admin/sqltrpt.sql`

Empfehlungen des SQL Tuning Advisor

- ∅ Auflistung von Objekten in den untersuchten SQL-Anweisungen, die keine oder veraltete Statistiken haben.
- ∅ Vorschläge für das Anlegen von Indizes, die die Performance verbessern können
- ∅ Empfehlungen für die Reformulierung der SQL
- ∅ Hinweise für bessere Ausführungspläne. Diese können dann als SQL-Profil implementiert werden. **Dazu ist keine Änderung der SQL oder von Parametern nötig!**

Die Empfehlungen werden **nicht** automatisch implementiert!

Anlegen und Implementieren von SQL-Profilen

1. Ermittlung des Input für den STA – Mögliche Quellen sind

- Manuelle Eingabe
- Cursor Cache (v\$sql)
- Top-Activity-Seite im Enterprise Manager
- Active Session History (ASH)
- Output des Automatic Database Diagnostic Monitor (ADDM)
- Snapshots des Automatic Workload Repository (AWR)

2. Erstellung eines Tuning Task mit

```
DBMS_SQLTUNE.CREATE_TUNING_TASK
```

3. Ausführen des Tuning Task mit

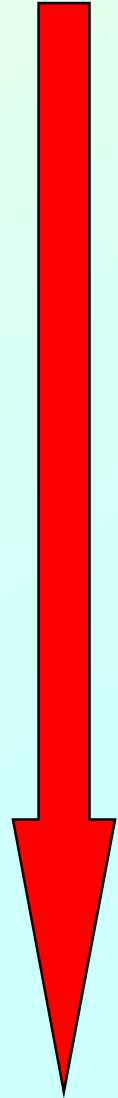
```
DBMS_SQLTUNE.EXECUTE_TUNING_TASK
```

4. Anzeigen des Reports

```
DBMS_SQLTUNE.REPORT_TUNING_TASK
```

5. Implementieren

```
DBMS_SQLTUNE.ACCEPT_SQL_PROFILE
```



Spezielle Einstellungen

- Ø Bildung von *SQL Tuning Sets* (STS): Berücksichtigung von Abhängigkeiten
- Ø SQL-Profile sind kategorisierbar (Standardkategorie DEFAULT): Reaktion auf verschiedene Lastsituationen
- Ø Es kann in der SQL mit Bindevariablen gearbeitet werden (Parameter `bind_list` von `DBMS_SQLTUNE.CREATE_TUNING_TASK`).
- Ø Man kann einstellen (Parameter `force_match => TRUE` in `DBMS_SQLTUNE.ACCEPT_SQL_PROFILE`), daß Anweisungen, die sich nur durch Literale unterscheiden, den gleichen Ausführungsplan verwenden. Dadurch wird für die einzelne Anweisung die Einstellung von `CURSOR_SHARING = EXACT` praktisch ignoriert.

SQL Profile im Enterprise Manager erstellen

Oracle Enterprise Manager (SYSMAN) - Top Aktivität - Windows Internet Explorer

http://gandall.wincor-nordorf.com:7779/em/console/database/instance/awaitDetails?event=doLoad&target=REPORT&type=oracle_database&waitClass=D&view&datasource=SQL

Oracle Enterprise Manager (SYSMAN) - Top Aktivität

Betrügsüberwachung ist aktiv.

ORACLE Enterprise Manager 10g

Grid Control

Standardverzeichnis | Ziele | Deployments | Alerts | Setup | Voreinstellungen | Hilfe | Abmelden

Hosts | Datenbanken | Application Server | Web-Anwendungen | Services | Systeme | Gruppen | Alle Ziele | Oracle Applications

Datenbank-Instance: REPORT > Angemeldet als SYSTEM

Top Aktivität

Ziehen Sie das schattierte Feld, um den Zeitraum für den Detail-Abschnitt unten zu ändern.

Daten anzeigen: Echtzeit: Aktualisierung alle 15 Sekunden

Detail für ausgewähltes Fünf-Minuten-Intervall

Startzeit: 25.10.2007 14.10 Uhr CEST

ASH-Bericht ausführen

Top-SQL

SQL Tuning Advisor planen | SQL Tuning Set erstellen

Alles auswählen | Nichts auswählen

Auswählen	Aktivität (%)	SQL ID	SQL-Typ
<input checked="" type="checkbox"/>	65,57	dvd8j02crvfxg	SELECT
<input type="checkbox"/>	5,04	9gfyvhu81mw68	PL/SQL EXECUTE
<input type="checkbox"/>	3,51	dt7j2a614sby0	PL/SQL EXECUTE
<input type="checkbox"/>	3,07	7ttgrx1mqxg7	PL/SQL EXECUTE
<input type="checkbox"/>	2,41	8pg7fuwh68pn	PL/SQL EXECUTE
<input type="checkbox"/>	2,19	51a5dgwd185aa	PL/SQL EXECUTE
<input type="checkbox"/>	2,19	5a4h23y5aav13	UNKNOWN
<input type="checkbox"/>	1,97	g83hmsy2kdv7v	SELECT
<input type="checkbox"/>	1,97	qfm7s34970d5j	PL/SQL EXECUTE
<input type="checkbox"/>	1,75	bs3301zdyw68	PL/SQL EXECUTE

Gesamtanzahl von Samples: 456

Top Sessions

Anzeigen: Top Sessions

Aktivität (%)	Session-ID	Benutzername	Programm
63,62	151	DATAMANAGER	sqlplus.exe
2,55	118	APEX_PUBLIC_USER	httpd@crm0160 (TNS V1-V3)
2,55	75	APEX_PUBLIC_USER	httpd@crm0160 (TNS V1-V3)
1,70	108	APEX_PUBLIC_USER	httpd@crm0160 (TNS V1-V3)
1,70	139	APEX_PUBLIC_USER	httpd@crm0160 (TNS V1-V3)
1,49	79	APEX_PUBLIC_USER	httpd@crm0160 (TNS V1-V3)
1,49	113	APEX_PUBLIC_USER	httpd@crm0160 (TNS V1-V3)
1,49	121	APEX_PUBLIC_USER	httpd@crm0160 (TNS V1-V3)
1,49	99	APEX_PUBLIC_USER	httpd@crm0160 (TNS V1-V3)
1,28	130	APEX_PUBLIC_USER	httpd@crm0160 (TNS V1-V3)

Gesamtanzahl von Samples: 470

Zusätzliche Überwachungs-Links

Sie finden Top Sessions und Top SQL-Daten aus ASH auf der Seite Top-Aktivität.

- Top Consumer
- Ähnliches SQL
- Blockierende Sessions
- Hängen-Analyse
- Instance-Sperren
- Instance-Aktivität
- Normalisierte Baseline-Metriken
- Sessions suchen
- Snapshots
- SQL Tuning Sets

Standardverzeichnis | Ziele | Deployments | Alerts | Konformität | Jobs | Berichte | Setup | Voreinstellungen | Hilfe | Abmelden

Copyright © 1996, 2007, Oracle. All rights reserved. Alle Rechte vorbehalten.
Oracle, JD Edwards, PeopleSoft, und Retek sind eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen. Andere Namen und Bezeichnungen können Marken ihrer jeweiligen Inhaber sein.
Info: Oracle Enterprise Manager

Empfehlungen im Enterprise Manager

Oracle Enterprise Manager (SYSMAN) - Empfehlungen für SQL-ID:dyd8j02crvfxq - Windows Internet Explorer

http://gandalf.winco-nixdorf.com:7779/em/console/database/instance/sqltune?event=viewtrn&task_id=214&objectid=1&target=REPORT&type=oracle_database

Oracle Enterprise Manager (SYSMAN) - Empfehlungen

Betrugsüberwachung ist aktiv.

ORACLE Enterprise Manager 10g

Grid Control

Standardverzeichnis | Ziele | Deployments | Alerts | Konformität | Jobs | Berichte

Hosts | Datenbanken | Application Server | Web-Anwendungen | Services | Systeme | Gruppen | Alle Ziele | Oracle Applications

Datenbank-Instance: REPORT > Zentrales Advisor > SQL Tuning-Ergebnisse:SQL_TUNING_1193313829476 > Empfehlungen für SQL-ID:dyd8j02crvfxq

Angemeldet als SYSTEM

Nur eine Empfehlung sollte implementiert werden.

SQL-Text

```
SELECT KW '|| o461377.report_week AS c_4. (o149329.machine_id) || ' '|| (o149329.machine_location) || ' '|| (o149329.customer_category_level_2) ...
```

Empfehlung auswählen

Original-Explain-Plan (mit Anmerkung versehen)

Implementieren

Auswählen	Typ	Ergebnisse	Empfehlungen	Begründung	Neuer Vorteil Explain-Plan (%)	Explain-Pläne vergleichen
<input checked="" type="radio"/>	SQL-Profil	Ein potenziell besserer Ausführungsplan wurde für diese Anweisung gefunden.	Sie sollten das empfohlene SQL-Profil akzeptieren.		33.83	
<input type="radio"/>	SQL restrukturieren	Ein aufwändiger Vorgang "UNION" wurde in Zeilen-ID 15 des Ausführungsplans gefunden.	Sie sollten "UNION ALL" anstelle von "UNION" verwenden, wenn Duplikate zulässig sind oder Eindeutigkeit gewährleistet ist.	"UNION" ist ein aufwändiger und blockierender Vorgang, weil doppelte Zeilen eliminiert werden müssen. "UNION ALL" ist eine weniger aufwändige Alternative, vorausgesetzt Duplikate sind zulässig oder Eindeutigkeit ist gewährleistet.		
<input type="radio"/>	SQL restrukturieren	Das Prädikat TO_NUMBER("SYS_ALIAS_2" "REPORT_WEEK")>=TO_NUMBER(CHAR(SYSDATE@!-56 'YYYYIW'), das in Zeilen-ID 11 des Ausführungsplans benutzt wird, enthält einen Ausdruck auf der indizierten Spalte "REPORT_WEEK". Dieser Ausdruck hindert den Optimizer daran, Indizes auf Tabelle "WNRP_DATA" "XWVN_CUST_AVAILABILITY_SUM" auszuwählen.	Schreiben Sie das Prädikat in eine entsprechende Form um, damit Indizes genutzt werden können. Alternativ erstellen Sie einen funktionsbasierten Index auf dem Ausdruck.	Der Optimizer kann einen Index nicht benutzen, wenn das Prädikat eine Ungleichheits-Bedingung ist oder wenn auf der indizierten Spalte ein Ausdruck oder eine Konvertierung des impliziten Datentyps vorhanden ist.		
<input type="radio"/>	Verschiedene	Der Optimizer konnte die View in Zeilen-ID 14 des Ausführungsplans nicht konsolidieren.				

Standardverzeichnis | Ziele | Deployments | Alerts | Konformität | Jobs | Berichte | Setup | Voreinstellungen | Hilfe | Abmelden

Copyright © 1996, 2007, Oracle. All rights reserved. Alle Rechte vorbehalten.
Oracle, JD Edwards, PeopleSoft, und Retek sind eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen. Andere Namen und Bezeichnungen können Marken ihrer jeweiligen Inhaber sein.
[Info Oracle Enterprise Manager](#)

Zur Beachtung

Nach Implementierung von SQL-Profilen wird zur Laufzeit trotzdem der CBO im Normalmodus verwendet.

Das Verfahren ist im Gegensatz zu Stored Outlines flexibel, d.h. wenn sich die Bedingungen gravierend ändern, kommt unter Umständen auch ein anderer Ausführungsplan zur Anwendung!

Daraus ergibt sich:

Wenn sich die Antwortzeit signifikant verschlechtert, muß der STA erneut ausgeführt werden.

Vergleich mit Stored Outlines

Anstelle von SQL-Profilen können Ausführungspläne als Stored Outline im Data Dictionary abgelegt werden. Auch diese können kategorisiert werden. Der **Workflow** ist folgender:

- Der problematische Parameter wird auf einen nicht unterstützten Wert gestellt.
- Für die SQL wird ein Stored Outline erzeugt.
- Der Parameter wird zurückgestellt.
- Die Verwendung des Stored Outline wird eingeschaltet.

Der **Vorteil** gegenüber den SQL-Profilen ist, daß sich die Ausführungspläne im laufenden Betrieb nicht ändern können. Die SQL wird *immer* nach dem gleichen Plan ausgeführt, solange das Outline nicht geändert, überschrieben, gelöscht oder seine Verwendung unterdrückt wird.

Nachteile von Stored Outlines

- ∅ Parameter müssen für die Erstellung des Outline wenigstens zeitweise geändert werden.
- ∅ SQL muß geändert werden, da z.B. Hints bei der Outline-Erstellung greifen. (Widerspricht der Aufgabenstellung)
- ∅ Es gibt keine Möglichkeit, das Cursor Sharing ohne dauerhafte Änderung des Parameters `CURSOR_SHARING` zu erzwingen.
- ∅ Der Optimizer arbeitet bei der Erstellung der Outlines nur im Normalmodus, d.h. er findet gegebenenfalls nicht den besten Ausführungsplan. Es hat sich gezeigt, daß die Kosten für die Ausführung des vom STA generierten Plans deutlich geringer sein können als die für die Ausführung des Stored Outline.

STA: Allgemeine Angaben

Statement:

```
SELECT /*+NO_INDEX(MITARBEITER MITARB_PK) */ *  
FROM mitarbeiter WHERE mitarbeiter_nr = 1001;
```

Empfehlung STA:

```
DBMS_SQLTUNE.REPORT_TUNING_TASK('NEW_TASK')
```

```
-----  
GENERAL INFORMATION SECTION  
-----
```

```
Tuning Task Name           : new_task  
Tuning Task Owner         : SCOTT  
Scope                     : COMPREHENSIVE  
Time Limit(seconds)       : 600  
Completion Status         : COMPLETED  
Started at                 : 08/28/2007 10:12:32  
Completed at              : 08/28/2007 10:12:33  
Number of SQL Profile Findings : 1  
Schema Name: SCOTT  
SQL ID                    : dcb7dx00j744v  
SQL Text                   : SELECT /*+NO_INDEX(MITARBEITER MITARB_PK) */ * FROM mitarbeiter  
                           WHERE mitarbeiter_nr = 1001
```

STA: Beispiel SQL-Code (Irreführender Hint)

FINDINGS SECTION (1 finding)

1- SQL Profile Finding (see explain plans section below)

Recommendation (estimated benefit: 49,97%)

- Sie sollten das empfohlene SQL-Profil akzeptieren.

execute dbms_sqltune.accept_sql_profile(task_name => 'new_task', replace => TRUE);

EXPLAIN PLANS SECTION

1- Original With Adjusted Cost

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		1	39	2 (0)
* 1	TABLE ACCESS FULL	MITARBEITER	1	39	2 (0)

2- Using SQL Profile

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		1	39	1 (0)
1	TABLE ACCESS BY INDEX ROWID	MITARBEITER	1	39	1 (0)
* 2	INDEX UNIQUE SCAN	MITARB_PK	1		0 (0)

Beispiel Parameter **OPTIMIZER_INDEX_COST_ADJ**

1. Original (51 Zeilen) - **OPTIMIZER_INDEX_COST_ADJ = 100**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		1	333	8878 (1)
...					
* 45	TABLE ACCESS FULL	tab	1363K	41M	8521 (1)

2. Mit SQL-Profil (65 Zeilen) - **OPTIMIZER_INDEX_COST_ADJ = 100**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		1	333	117 (3)
...					
* 53	TABLE ACCESS BY INDEX ROWID	tab	1	32	3 (0)
* 54	INDEX RANGE SCAN	ind	1		2 (0)

3. Zum Vergleich Stored Outline (52 Zeilen) - **OPTIMIZER_INDEX_COST_ADJ = 10**

Id	Operation	Name	Rows	Bytes	Cost (%CPU)
0	SELECT STATEMENT		200	66600	1840 (1)
...					
* 36	TABLE ACCESS BY INDEX ROWID	tab	1	32	1 (0)
...					
* 46	INDEX RANGE SCAN	ind	1		1 (0)

Beispiel OPTIMIZER_FEATURES_ENABLE

Explain-Pläne vergleichen

Neuer Explain-Plan mit SQL-Profil
 Alle einblenden | Alle ausblenden

Vorgang	Zeilen-ID	Objekt	Objektyp	Reihenfolge	Zeilen	Größe (KB)	Kostenfaktor
SELECT STATEMENT	0			15	1	0,240	19.797
SORT ORDER BY	1			14	1	0,240	19.797
VIEW	2			13	1	0,240	19.796
WINDOW SORT	3			12	1	0,238	19.796
SORT GROUP BY	4			11	1	0,238	19.796
FILTER	5			10			
NESTED LOOPS OUTER	6			9	1	0,238	19.794
NESTED LOOPS	7			6	1	0,185	19.793
HASH JOIN	8			3	1	0,157	19.792
TABLE ACCESS FULL	9	EUL.XXWN_IBASE_PQ	TABLE	1	4.491	346,474	5.838
TABLE ACCESS FULL	10	EUL.XXWN_INCIDENTS_PQ	TABLE	2	40.565	3.248,369	13.953
TABLE ACCESS BY INDEX ROWID	11	EUL.XXWN_INCIDENTS_MV	TABLE	5	1	0,027	1
INDEX RANGE SCAN	12	EUL.XXWN_INCIDENTS_MV_N1	INDEX	4	1		1
TABLE ACCESS BY INDEX ROWID	13	EUL.XXWN_MATERIAL_MV	TABLE	8	1	0,054	1
INDEX RANGE SCAN	14	EUL.XXWN_MATERIAL_MV_U2	INDEX	7	1		1

Original-Explain-Plan (mit Anmerkung versehen)
 Gibt eine Anpassung von dem Originalplan durch SQL Tuning Advisor an
 Als Nächstes folgt der Original-Explain-Plan für die SQL-Anweisung, die optimiert wird.
 Alle einblenden | Alle ausblenden

Vorgang	Zeilen-ID	Objekt	Objektyp	Reihenfolge	Zeilen	Größe (KB)	Kostenfaktor
SELECT STATEMENT	0			16	1	0,240	72.0
SORT ORDER BY	1			15	1	0,240	72.0
VIEW	2			14	1	0,240	72.0
WINDOW SORT	3			13	1	0,238	72.0
SORT GROUP BY	4			12	1	0,238	72.0
FILTER	5			11			
NESTED LOOPS OUTER	6			10	1	0,238	72.0
NESTED LOOPS	7			7	1	0,185	72.0
NESTED LOOPS	8			4	1	0,157	72.0
TABLE ACCESS FULL	9	EUL.XXWN_INCIDENTS_PQ	TABLE	1	40.565	3.248,369	13.9
TABLE ACCESS BY INDEX ROWID	10	EUL.XXWN_IBASE_PQ	TABLE	3	1	0,077	
INDEX RANGE SCAN	11	EUL.XXWN_IBASE_PQ_N2	INDEX	2	102		
TABLE ACCESS BY INDEX ROWID	12	EUL.XXWN_INCIDENTS_MV	TABLE	6	1	0,027	
INDEX RANGE SCAN	13	EUL.XXWN_INCIDENTS_MV_N1	INDEX	5	1		
TABLE ACCESS BY INDEX ROWID	14	EUL.XXWN_MATERIAL_MV	TABLE	9	1	0,054	
INDEX RANGE SCAN	15	EUL.XXWN_MATERIAL_MV_U2	INDEX	8	1		

Resultate

SQL-Code:

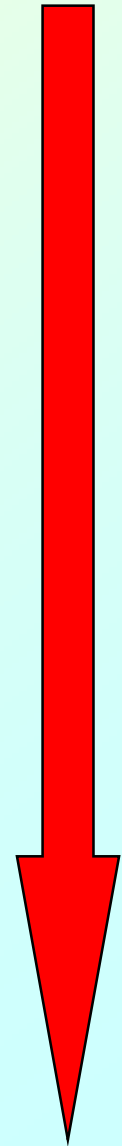
Die Indexverwendung wurde mit dem Hint `no_index` unterdrückt. Der STA empfiehlt einen besseren Ausführungsplan mit Indexverwendung. Nach Implementierung des SQL-Profiles wird der Index verwendet, **ohne** daß am Code eine Veränderung vorgenommen werden mußte.

Initialisierungsparameter:

Die *geforderte* Einstellung von `OPTIMIZER_INDEX_COST_ADJ = 100` erzwingt einen Ausführungsplan, der einen Full Table Scan durchführt, wo ein Index Scan günstiger wäre. Der STA schlägt einen Ausführungsplan mit Indexverwendung vor. Zusätzlich ändert sich auch die Joinverarbeitung drastisch. Der STA prognostiziert eine Laufzeitverbesserung von mehr als 98%. Die *vorgeschriebene* Einstellung von `OPTIMIZER_FEATURES_ENABLE=10.2.0.3` verlängert bei bestimmten Statements die Antwortzeit von Sekunden bzw. Minuten auf mehrere Stunden. Mit der Anwendung des STA läßt sich ein Ausführungsplan erzeugen, der zu einer akzeptablen Verarbeitungszeit führt. Nach Implementierung des SQL-Profiles wurden die besseren Ausführungspläne verwendet, **ohne** daß Parameter geändert worden wären.

Workflow zur Verwendung von SQL-Profilen

1. Ermitteln der problematischen SQL
2. Bilden eines STS (optional)
3. Export der aktuellen Schemastatistiken vom Produktivsystem auf das Testsystem
4. Anwenden des STA im Tuning Mode auf dem Testsystem
5. Implementieren und Test der SQL-Profile auf dem Testsystem
6. Speichern der SQL-Profile in einer Staging Table
7. Export der Staging Table und Import im Produktivsystem
8. Anwenden der SQL-Profile im Produktivsystem
9. Überprüfen, ob der durch die SQL-Profile gegebene Performancegewinn auch noch nach längerer Zeit gegeben ist und eventuell erneuter Einsatz des STA



Fazit

SQL-Profile stellen eine sehr gute Möglichkeit dar, mit der Eingangs geschilderten Situation produktiv umzugehen.

Literatur:

- Ø Oracle 10g Release 2 Documentation: Performance Tuning Guide
- Ø Oracle 10g Release 2 Documentation: PL/SQL Packages and Types Reference
- Ø Metalink 262687.1 How to use the Sql Tuning Advisor
- Ø Metalink 271196.1 Automatic SQL Tuning – SQL Profile
- Ø Metalink 276103.1 Performance Tuning Using 10g Advisors and Manageability

Dr. Frank Haney
info@it-haney.de
Tel.: 03641-210224

