

**Was bringt
Extended SQL-Tracing
für das
Performance Tuning
?**

Ein Erfahrungsbericht

Inhalt

1. Motivation – Tuningziele und -bereiche
2. Defizite traditioneller Tuningmethoden
3. Was ist Extended SQL-Tracing?
4. Inhalt und Auswertung der Trace Files
5. Ein Fallbeispiel
6. Ausblick Oracle 10g
7. Fazit

Tuningbereiche

Geschäftsprozesse

Datendesign (Schema)

Anwendung (SQL und Forms)

Speichernutzung (SGA und PGA)

I/O-Subsystem (Hintergrundprozesse)

Zugriffskonflikte (Locks und Latches)

Betriebssystem

Netzwerk

Von Schritt zu Schritt steigender Aufwand bei sinkendem Effekt!

Tuningziele

Zufriedene Benutzer

- **Antwortzeit**
- **Durchsatz**

- **Trefferrate im Cache**
- **Cachennutzung**
- **Wiederverwendung von Code**
- **Geringster Blockzugriff**
- **Schnelles Lesen und Schreiben**
- **Kein Warten auf Ressourcen**
- **Minimale Beeinträchtigung durch Verwaltungsaufgaben**

Defizite traditionellen Performance Tunings

- Es gibt im eigentlichen Sinne keine zielführende Methode, nur Tips und Techniken, die gerade in komplexen Infrastrukturen häufig versagen..
- Fehlender Determinismus: Die Korrelation zwischen Aktionen und Wirkungen ist meist schwach. Trial and Error!
- Es muß häufig von systemweiten Summen auf Session-Details geschlossen werden, z.B. bei Nutzung von *Statspack* und *v\$-views*.
- Trace Files enthalten keine Informationen über *Waits* und *Binds*!
- Erfolgskriterium ist oft nicht die Antwortzeit, sondern mehr oder weniger relevante Benchmarks, Trefferquoten etc.
- Es gibt Probleme bei der Eingrenzung der Nutzeraktionen, die die wesentlichen Laufzeitprobleme verursachen.
- Das Tuning setzt nicht bei den Ereignissen an, die den größten Einfluß auf die Antwortzeit haben.

Was verspricht Extended SQL-Tracing?

1. Extended SQL-Tracing ermöglicht nicht nur SQL-Tuning, sondern unterstützt einen ganzheitlichen Ansatz.
2. Die Methode ist deterministisch: Die Auswirkungen von Tuningmaßnahmen auf die Performance sind vorhersagbar.
3. Es gibt nur zwei Kriterien: Antwortzeit und Kosten
4. Statistiken können spezifisch für die als problematisch ermittelte Nutzeraktion gesammelt werden.
5. Meßfehler sind eher beherrsch- bzw. eliminierbar.
6. Der Beitrag jedes einzelnen Database Calls zur Antwortzeit läßt sich ermitteln.
7. Die Performance-Implicationen eigentlich „unzugänglicher“ Nutzeraktionen (Fremdapplikationen) werden transparent.

Methoden für das Extended SQL-Tracing

Oracle's Pseudo-Error Debugging Events

- Fehler-Nummern von 10000 - 10999
- Eine komplette Liste findet sich in \$ORACLE_HOME/rdbms/mesg/oraus.msg (Leider nicht auf Windows-Plattformen!)
- **Event 10046 = Extended SQL-Tracing**
- **Statistik-Ebenen (level):**

Level	Funktion
0	keine Statistik
1	entspricht SQL_TRACE = TRUE
2	Derzeit identisch mit 1
4	1 + BINDS
8	1 + WAITs
12	1 + BINDS + WAITs

Starten und Stoppen des Extended SQL-Tracing

Eigene Session

```
ALTER SESSION SET EVENTS '10046 trace name context forever, level <level>';
```

>> *Benutzeraktionen*

```
ALTER SESSION SET EVENTS '10046 trace name context off';
```

Fremde Session

1. ORADEBUG

```
oradebug setorapid <pid>
```

```
oradebug event 10046 trace name context forever, level <level>
```

>> *Benutzeraktionen*

```
oradebug event 10046 trace name context off
```

2. DBMS_SUPPORT

```
execute dbms_support.start_trace_in_session(<sid>, <serial#>,  
      waits => TRUE, binds => TRUE);
```

>> *Benutzeraktionen*

```
execute dbms_support.stop_trace_in_session(<sid>, <serial#>);
```

3. DBMS_SYSTEM

```
execute dbms_system.set_ev( <sid>, serial#>, 10046, 12);
```

>> *Benutzeraktionen*

```
execute dbms_system.set_ev(<sid>, <serial#>, 10046, 0);
```

Probleme

1. Von Oracle im Level > 1 offiziell nicht unterstützt (bis 10g)
2. Wird in der Oracle-Dokumentation und den meisten Tuning-Büchern nicht behandelt
3. Mißverständnisse über die Wirkungsweise selbst bei Experten
4. Es gibt keine Tools zum adäquaten Sammeln der Daten.
5. Unterschiedliche Maßeinheiten (8i cs vs. 9i μ s)
6. Kaum Standard-Tools zur Interpretation der Trace Files
 - TKPROF gibt keine totale Antwortzeit einer Session. Außerdem werden Ereignisse zwischen den Calls nicht richtig interpretiert und die Zuordnung der rekursiven SQL ist schwierig.
 - Der Trace Analyzer (TRCA) wird von Oracle offiziell nicht unterstützt.

Auswerten der Trace Files

Zu beachten sind dabei:

- Vorwärtsaggregation
- Rekursive Tiefe (Child Cursor)
- Ereignisse innerhalb der und zwischen den Calls
- Idle und nonidle waits

Elemente der Trace Files

- Session- und Anwendungsinformationen sowie Timestamp
- Cursor Identifikation: PARSING IN CURSOR
- Database Calls: PARSE, EXECUTE, FETCH
- Warteereignisse: WAIT
- Bindevariablen: BINDS
- Ausführungsplan (Row Source): STAT
- Transaktionsende: XTEND

Rekursive Tiefe

```
PARSING IN CURSOR #10 len=116 dep=1 uid=0 oct=3 lid=0 tim=4853113354 hv=431456802
  ad='7b7c78cc'
select o.owner#,o.name,o.namespace,o.remoteowner,o.linkname,o.subname,o.dataobj#,o.flags from
  obj$ o where o.obj#=1
END OF STMT
PARSE #10:c=0,e=2730,p=0,cr=0,cu=0,mis=1,r=0,dep=1,og=0,tim=4853113314
BINDS #10:
  bind 0: dty=2 mxl=22(22) mal=00 scl=00 pre=00 oacflg=08 oacfl2=1 size=24 offset=0
  bfp=06022f2c bln=22 avl=04 flg=05
  value=33814
EXEC #10:c=0,e=2035,p=0,cr=0,cu=0,mis=0,r=0,dep=1,og=4,tim=4853116393
FETCH #10:c=0,e=242,p=0,cr=3,cu=0,mis=0,r=1,dep=1,og=4,tim=4853116884
=====
PARSING IN CURSOR #1 len=50 dep=0 uid=59 oct=6 lid=59 tim=4853210150 hv=421188132
  ad='7bbb63c0'
update emp set job = 'GEHILFE' where empno = :zahl
END OF STMT
PARSE #1:c=220317,e=579625,p=5,cr=68,cu=0,mis=1,r=0, dep=0,og=0,tim=4853210107
```

Warteereignisse

BINDS #1:

bind 0: dty=2 mxl=22(22) mal=00 scl=00 pre=00 oacflg=03 oacfl2=0 size=24 offset=0
bfp=0600f600 bln=22 avl=03 flg=05
value=7499

WAIT #1: nam='db file sequential read' **ela= 5876** p1=1 p2=50474 p3=1

WAIT #1: nam='db file sequential read' **ela= 2103** p1=1 p2=50466 p3=1

EXEC #1:c=0,**e=11438**,p=2,cr=1,cu=2,mis=0,r=1,dep=0,og=4,tim=4853222037

WAIT #1: nam='SQL*Net message to client' **ela= 18** p1=1413697536 p2=1 p3=0

WAIT #1: nam='SQL*Net message from client' **ela= 8744235** p1=1413697536 p2=1 p3=0

STAT #1 id=1 cnt=1 pid=0 pos=1 obj=0 op='UPDATE (cr=1 r=2 w=0 time=9390 us)'

STAT #1 id=2 cnt=1 pid=1 pos=1 obj=33814 op='INDEX UNIQUE SCAN EMP_EMPNO_PK
(cr=1 r=1 w=0 time=6227 us)'

=====
PARSING IN CURSOR #1 len=8 dep=0 uid=59 oct=45 lid=59 tim=4861969778 hv=35058394
ad='7bb7412c'

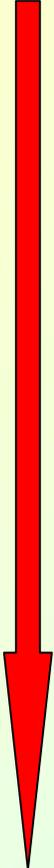
rollback

END OF STMT

PARSE #1:c=0,**e=1583**,p=0,cr=0,cu=0,mis=1,r=0,dep=0,og=4,tim=4861969735

XCTEND rlbk=1, rd_only=0

Allgemeine Vorgehensweise

- 
- Ermittlung der performance-kritischsten (teuersten) Nutzeraktionen der Anwendung
 - Überwachtes Tracing der Nutzer-Session
 - Erstellung eines Ressourcenprofils der Session (Wo bleibt die Antwortzeit?)
 - Bestimmung der kostenintensivsten SQL
 - Welche Calls der Nutzeraktion tragen wie und womit (CPU, Warteereignisse etc.) zur Antwortzeit bei?
 - Wo ist der größte Tuning-Effekt zu erwarten?
 - Wie kann das erreicht werden?
 - Was kostet das?

Ausblick auf Oracle 10g

Erweiterte Möglichkeiten:

- Neue Spalten bekannter Views (BLOCKING_SESSION in *v\$session*)
- Neue Views (*v\$session_wait_class*, *v\$event_histogram*, *v\$sess_time_model*)
- Historie der session-bezogenen Warteereignisse (*v\$active_session_history*, *dba_hist_active_session_hist*)
- Betriebssystemstatistiken (*v\$osstat*)

Probleme:

- Nur bestimmte Statistiken sind auf Session-Ebene verfügbar.
- Wenig komfortable Angabe der Zeiten : cs, ms, μ s
- Die Identifizierung der problematischen Nutzeraktion ist nach wie vor nicht trivial.
- „Session Time Model“ bietet keine Korrelation zu den Warteereignissen
- Die Zuordnung der Ereignisse zu den Calls fehlt.

Fazit:

- Es geht auch in 10g nicht ohne Extended SQL-Tracing.
- Neues Package **DBMS_MONITOR** (z.B. **SESSION_TRACE_ENABLE**)

Literatur

- C. V. Millsap, J. L. Holt: Optimizing Oracle Performance.
O'Reilly, Sebastopol (CA) 2003, 390 pp.
- R. J. Niemiec: Oracle 9i Performance Tuning: Tips & Techniques.
Mc Graw Hill, New York 2003, 826 pp.
- A. Nanda: Perform without Waiting.
Oracle Magazin, July/August 2004, pp.79-82
- K. Floss: Tracing SQL in Oracle Database 10g.
Oracle Magazin, September/October 2004, pp.79-80
- R. Shee, K. Deshpande, K. Gopalakrishnan: Oracle Wait Interface:
A Practical Guide to Performance Diagnostics & Tuning.
Mc Graw Hill, New York 2004, pp.
- <http://metalink.oracle.com>
- <http://www.hotsos.com>
- Oracle9i Database Online Documentation, Release 2

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

ORACLE

**CERTIFIED
PROFESSIONAL**