Tipps & Tricks: März 2002

Bereich:	DBA	Erstellung:	03/2003 HW
Versionsinfo:	8.2 9.2 10.2 11.1	Letzte Überarbeitung:	05/2009 BK

Tracing einer Datenbank Session

Wie kann man eine Session in der Datenbank tracen

Oft steht man als DBA vor der Aufgabe, an genauere Informationen über alle SQL Statements und Ausführungspläne einer bestimmten Session zu gelangen. Dies geht oft nur über die von der Datenbank bereitgestellte Funktionalität des Session-Tracing (nicht zu verwechseln mit DBMS_TRACE). Hierbei wird von der Datenbank eine Datei erzeugt, die nach dem Tracing ausgewertet werden kann.

Hinweis:

Um aussagekräftige Ergebnisse zu erhalten empfiehlt es sich, den Parameter TIMED_STATISTICS auf TRUE zu setzen. Dies kann für eine Session oder für das ganze System über das ALTER Kommando geschehen, ohne die Instanz durchstarten zu müssen.

Die aktuelle Session tracen

Die eigene Session zu tracen ist recht einfach, da man dazu nur über das Systemprivileg ALTER SESSION verfügen muss.

So können Sie das Tracing starten:

```
ALTER SESSION SET SQL_TRACE = TRUE;
Session altered.

ALTER SESSION SET EVENTS '10046 trace name context forever, level 1';
Session altered.
```

So können Sie das Tracing wieder stoppen:

```
ALTER SESSION SET SQL_TRACE = FALSE;
Session altered.
```

Eine fremde Session tracen

Etwas anders vorgehen muss man, wenn es eine andere Session betrifft. Oracle stellt für das Session-Tracing Prozeduren über das undokumentierte Package SYS.DBMS_SYSTEM zur Verfügung. Zum Ausführen benötigt man das Systemprivileg EXECUTE ON DBMS_SYSTEM, das bis zur Version Oracle8i jeder DBA hatte - ab Oracle9i jedoch nur mehr SYS.

- SET_SQL_TRACE_IN_SESSION (SID , SERIAL#, <TRACE_STATUS>)

Zum Ausführen dieser Prozedur benötigen Sie die SID und SERIAL# der jeweiligen Session. Sie erhalten diese Angaben aus der View V$SESSION:

```
SELECT S.SID,S.SERIAL#,S.USERNAME,S.MACHINE,S.OSUSER,P.SPID
FROM V$SESSION S, V$PROCESS P
```

```
WHERE S.PADDR = P.ADDR
AND BACKGROUND IS NULL;

SID    SERIAL# USERNAME   MACHINE    OSUSER    SPID
---- ---------- ---------- ---------- --------- ---------
   7       1754 SYSTEM     ultra60    oracle    20179
   8        150 HR         ultra60    oracle    20218
```

So können Sie nun das Tracing der SID 8 starten:

```
EXEC SYS.DBMS_SYSTEM.SET_SQL_TRACE_IN_SESSION(8,150,TRUE)
PL/SQL procedure successfully completed.
```

So können Sie das Tracing wieder stoppen:

```
EXEC SYS.DBMS_SYSTEM.SET_SQL_TRACE_IN_SESSION(8,150,FALSE)
PL/SQL procedure successfully completed.
```

Alle Sessions tracen

Um alle Sessions in einer Datenbank tracen zu können, müssen Sie einen Prameter in der Datei INIT.ORA eintragen und die Instanz neu starten.
So können Sie das Tracing über die Datei INIT.ORA aktivieren:

- SQL_TRACE = TRUE

oder

- EVENT = "10046 trace name context forever, level 1"

Zum Stoppen des gesamten Tracing müssen Sie die neuen Zeilen wieder entfernen und die Instanz erneut durchstarten.

Auswerten der Tracedatei

Durch das Tracing einer Benutzer-Session wird im Verzeichnis USER_DUMP_DEST eine Datei mit dem Namen >ORACLE_SID>_ora_<Process_ID>.trc (Unix) erzeugt. Diese Datei enthält Informationen über:

- alle ausgeführten SQL Statements
- Zeiten in 1/100 Sekunden für die Phasen: PARSE, EXECUTE und FETCH
- ORA-Fehlernummern
- Commit/Rollback
- Ausführungspläne

Hinweis:

Das Tracen eines Datenbankjobs erzeugt eine Tracedatei im Verzeichnis BACKGROUND_DUMP_DEST!

Die Tracedatei kann mit dem Oracle Tool TKPROF in ein lesbares Format gebracht werden.

Beispiel:

```
$ tkprof mp9i_ora_20218.trc mp9i_ora_20218.out explain=hr/hr
$ tkprof mp9i_ora_20218.trc mpo9i_ora_20218.out explain=hr/hr
sort=exeela,fchela,prsela sys=no
```

Beispiel für einen Ausschnitt aus der Tracedatei:

```
PARSING IN CURSOR #1 len=187 dep=0 uid=42 oct=3 lid=42 tim=1015181988171510
hv=166937731 ad='90381d4c'
     SELECT
        EMPLOYEE_ID,
        FIRST_NAME,
        LAST_NAME,
        JOB_ID,
        e.MANAGER_ID,
        e.DEPARTMENT_ID
     FROM HR.EMPLOYEES E, HR.Departments d
     where e.DEPARTMENT_ID = d.DEPARTMENT_ID

END OF STMT
PARSE #1:c=30000,e=45398,p=0,cr=2,cu=0,mis=1,r=0,dep=0,og=4,tim=1015181988171485
EXEC #1:c=0,e=166,p=0,cr=0,cu=0,mis=0,r=0,dep=0,og=4,tim=1015181988172438
FETCH #1:c=0,e=259,p=0,cr=2,cu=2,mis=0,r=1,dep=0,og=4,tim=1015181988173012
FETCH #1:c=0,e=469,p=0,cr=2,cu=0,mis=0,r=15,dep=0,og=4,tim=1015181988174799
FETCH #1:c=10000,e=472,p=0,cr=2,cu=0,mis=0,r=15,dep=0,og=4,tim=1015181988190290
FETCH #1:c=0,e=394,p=0,cr=2,cu=0,mis=0,r=15,dep=0,og=4,tim=1015181988205483
FETCH #1:c=0,e=403,p=0,cr=2,cu=0,mis=0,r=15,dep=0,og=4,tim=1015181988220868
FETCH #1:c=0,e=409,p=0,cr=2,cu=0,mis=0,r=15,dep=0,og=4,tim=1015181988236238
FETCH #1:c=0,e=444,p=0,cr=3,cu=0,mis=0,r=15,dep=0,og=4,tim=1015181988251485
FETCH #1:c=0,e=446,p=0,cr=3,cu=0,mis=0,r=15,dep=0,og=4,tim=1015181988266728
FETCH #1:c=0,e=28,p=0,cr=0,cu=0,mis=0,r=0,dep=0,og=4,tim=1015181988281569
STAT #1 id=1 cnt=106 pid=0 pos=0 obj=0 op='NESTED LOOPS '
STAT #1 id=2 cnt=107 pid=1 pos=1 obj=30826 op='TABLE ACCESS FULL EMPLOYEES '
STAT #1 id=3 cnt=106 pid=1 pos=2 obj=30822 op='INDEX UNIQUE SCAN '
```

Beispiel für den zugehörigen Ausschnitt aus der TKPROF-Datei:

```
SELECT
       EMPLOYEE_ID,
       FIRST_NAME,
       LAST_NAME,
       JOB_ID,
       e.MANAGER_ID,
       e.DEPARTMENT_ID
FROM HR.EMPLOYEES E, HR.Departments d
WHERE e.DEPARTMENT_ID = d.DEPARTMENT_ID

call     count       cpu    elapsed       disk      query    current
------- ------  -------- ---------- ---------- ---------- ----------
Parse        1      0.03       0.03          0          0          0
Execute      1      0.00       0.00          0          0          0
Fetch        9      0.01       0.00          0         18          2
------- ------  -------- ---------- ---------- ---------- ----------
total       11      0.04       0.03          0         18          2

     rows
----------
         0
```

```
          0
       106
----------
      106

Misses in library cache during parse: 1
Optimizer goal: CHOOSE
Parsing user id: 42  (HR)

Rows     Row Source Operation
-------  ---------------------------------------------------
    106  NESTED LOOPS
    107   TABLE ACCESS FULL EMPLOYEES
    106   INDEX UNIQUE SCAN (object id 30822)

Rows     Execution Plan
-------  ---------------------------------------------------
       0    SELECT STATEMENT   GOAL: CHOOSE
    106    NESTED LOOPS
    107    TABLE ACCESS   GOAL: ANALYZED (FULL) OF 'EMPLOYEES'
    106    INDEX (UNIQUE SCAN) OF 'DEPT_ID_PK' (UNIQUE)
```

Eine Übersicht über alle Parameter zu TKPROF erhalten Sie, wenn Sie TKPROF ohne Parameter starten!