



# Erweiterte 10046 Tracefileanalyse

mit QueryAdvisor

[www.queryadvisor.com](http://www.queryadvisor.com)



State of the art Database Analysis

# QueryAdvisor



## Frank Beutelschiess

- Dipl.-Informatiker aus Leidenschaft
- 17 Jahre freiberuflich tätig
- 10+ Jahre Siebelerfahrung
- Seit 2 Jahren QueryAdvisor.com

# Aktivierung



```
-- All versions.
SQL> ALTER SESSION SET sql_trace=TRUE;
SQL> ALTER SESSION SET sql_trace=FALSE;

SQL> EXEC DBMS_SESSION.set_sql_trace(sql_trace => TRUE);
SQL> EXEC DBMS_SESSION.set_sql_trace(sql_trace => FALSE);

SQL> ALTER SESSION SET EVENTS '10046 trace name context forever, level 8';
SQL> ALTER SESSION SET EVENTS '10046 trace name context off';

SQL> EXEC DBMS_SYSTEM.set_sql_trace_in_session(sid=>123, serial#=>1234, sql_trace=>TRUE);
SQL> EXEC DBMS_SYSTEM.set_sql_trace_in_session(sid=>123, serial#=>1234, sql_trace=>FALSE);

SQL> EXEC DBMS_SYSTEM.set_ev(si=>123, se=>1234, ev=>10046, le=>8, nm=>' ');
SQL> EXEC DBMS_SYSTEM.set_ev(si=>123, se=>1234, ev=>10046, le=>0, nm=>' ');

-- Available from SQL*Plus since 8i (commandline utility prior to this.
SQL> CONN sys/password AS SYSDBA; -- User must have SYSDBA.
SQL> ORADEBUG SETMYPID;           -- Debug current session.
SQL> ORADEBUG SETOSPID 1234;     -- Debug session with the specified OS process.
SQL> ORADEBUG SETORAPID 123456;  -- Debug session with the specified Oracle process ID.

SQL> ORADEBUG EVENT 10046 TRACE NAME CONTEXT FOREVER, LEVEL 12;
SQL> ORADEBUG TRACEFILE_NAME;    -- Display the current trace file.
SQL> ORADEBUG EVENT 10046 TRACE NAME CONTEXT OFF;

-- All versions, requires DBMS_SUPPORT package to be loaded.
SQL> EXEC DBMS_SUPPORT.start_trace(waits=>TRUE, binds=>FALSE);
SQL> EXEC DBMS_SUPPORT.stop_trace;

SQL> EXEC DBMS_SUPPORT.start_trace_in_session(sid=>123, serial=>1234, waits=>TRUE, binds=>FALSE);
SQL> EXEC DBMS_SUPPORT.stop_trace_in_session(sid=>123, serial=>1234);
```

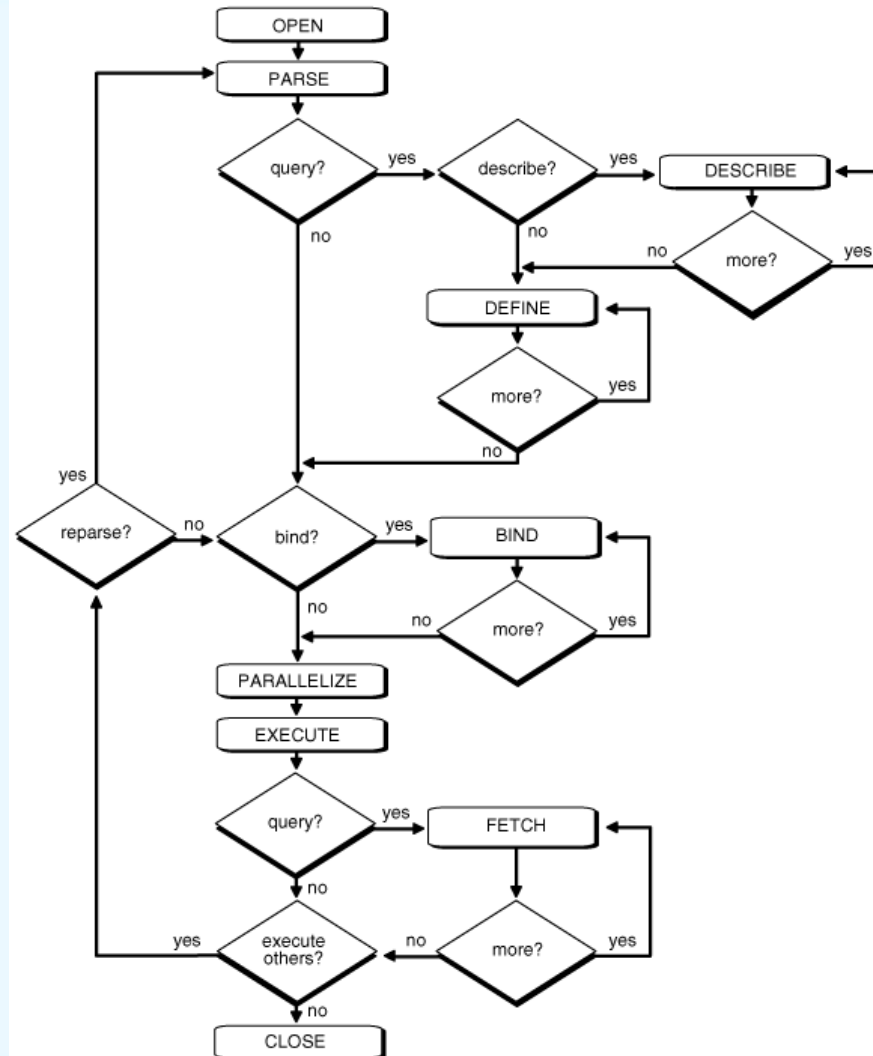
# Detail Level



## Level Values

Level	Available Events
1	Execute, Fetch, Parse, XCTEND
4	Bind, Execute, Fetch, Parse, XCTEND
8	Execute, Fetch, Parse, Wait, XCTEND
12	Bind, Execute, Fetch, Parse, Wait, XCTEND

# Abarbeitung SQL



# APPNAME/Bind



APPNAME	
name	can be modified with <code>dbms_application_info</code>
mod	module name
mh	module hash
act	action
ah	action hash value
Bind	
dty	OCI datatype
avl	length
value	value
mxl	maximal possible length of this datatype

# Parsing in Cursor / WAIT



PARSING IN CURSOR / WAIT	
nam	nam
len	length of SQL
dep	recursive depth
uid	userid calling
oct	Oracle command type
lid	userid owning
ela	elapsed time
tim	time
hv	statement id
ad	library cache address from V\$SQL
P1,P2,P3	parameter1...3 from V\$EVENT_NAME

# EXEC / FETCH / ERROR / ...



EXEC / FETCH / ERROR / ERROR UNMAP / PARSE / UNMAP	
c	CPU time
e	elapsed time
p	physical reads
cr	consistent reads
cu	current reads
mis	hardparse due to library cache miss
r	rows processed/returned
dep	recursive depth
og	optimizer goal (1=all/2=first/3=rule/4=choose)
tim	time



# STAT / XCTEND



STAT	
id	unique id in set
cnt	number of rows returned
pid	parent id
pos	?
obj	object id
op	operation
cr	consistent mode reads
r	number of blocks read with OS calls
w	number of blocks written with OS calls
tim	elapsed duration in micro
parent	stats include children stats
XCTEND	
rlbk	1 = rollback, 0 = otherwise
rd_only	1 = transaction changed no data



Der Speicherort wird durch den Prozess festgelegt. Der Trace von Benutzer-Sessions befindet sich somit im UDUMP Verzeichnis.

```
SQL> ALTER SYSTEM SET user_dump_dest = '/app/oracle/admin/o102/udump' SCOPE=both;
```

Im Fall von parallel ausgeführten Queries kommen Backgroundprozesse hinzu. Erwartungsgemäß werden die Tracefiles in dem BDUMP Verzeichnis abgelegt.

Der Bezug der abhängigen Prozesse kann über den Session Identifier des erzeugenden Prozesses hergestellt werden.

Der Vollständigkeit halber soll noch kurz auf die Existenz des CDUMP Verzeichnis (C=Core) hingewiesen werden, wo sich hoffentlich keine Datei befindet.

# Rekursives SQL



Automatisch erzeugtes Statement gegen das Data Dictionary um andere Kommandos zu unterstützen.

Beim Erstellen einer neuen Tabell via Create wird automatisch ein Insert Statement gegen das Data Dictionary erzeugt, um den entsprechenden Eintrag in SYS.TAB\$ anzulegen. Dieses Statement ist im Tracefile ersichtlich.

Weitere Szenarien mit rekursiven SQL Statements sind z.B.:

- Trigger
- SQL innerhalb von PL/SQL

# Gemeinsame Zeitbasis



Vor Oracle 9i beziehen sich alle Zeitangaben auf 1/100s. Mit 9i wurde die dargestellte Auflösung auf 1/1000.000s erhöht.

Der angezeigte Wert leitet sich von V\$TIMER ab. Der Timer beginnt bei einer Breite von 4 Byte mit dem Hochfahren der Instanz zu loszulaufen und hat ca. alle 497 Tage einen Überlauf.

Eine genaue Beschreibung findet sich in Metalink Note 39817.1.

Es besteht somit nur die Möglichkeit anhand der in den einzelnen Tracefiles angeführten absoluten Zeiten eine geeignete Kalibrierung durchzuführen.

# Mergen von Session



Bei einem Dedicated Server entspricht 1 Benutzersession genau einem Prozess. Alle Aktivitäten dieses Prozesses werden in einem Tracefile dokumentiert.

Bei einem Shared Server kann 1 Benutzersession über mehrere Prozesse verteilt nacheinander ausgeführt werden. Somit werden die Aktivitäten über mehrere Tracefiles verteilt und müssen wieder zusammengeführt werden. Dies kann z.B. wieder über den Session Identifier gemacht werden.

Oracle bietet ein Tool namens TRCSESS an, welches zur Konsolidierung von mehreren Tracefiles verwendet werden kann.

```
trcsess [output=output_file_name] [session=session_id] [clientid=client_id]  
[service=service_name] [action=action_name] [module=module_name]  
[trace_files]
```

# TKPROF von Oracle



```
$ tkprof orcl102_ora_3064.trc output.prf EXPLAIN=scott/tiger SYS=NO
```

```
Tkprof: Release 9.2.0.1.0 - Production on Tue Dec 24 15:32:43 2002  
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.
```

```
Trace file: ORCL102_ora_3064.trc
```

```
...
```

```
Sort options: default
```

```
select * from employee where emp_id = 3737
```

call	count	cpu	elapsed	disk	query	current	rows
Parse	10	0.00	0.03	0	0	0	0
Execute	10	0.00	0.00	0	0	0	0
Fetch	20	0.34	0.35	72	4730	0	10
total	40	0.34	0.39	72	4730	0	10

```
Misses in library cache during parse: 1
```

```
Optimizer goal: CHOOSE
```

```
Parsing user id: 59
```

```
Rows  Row Source Operation
```

```
-----  
1 TABLE ACCESS FULL EMPLOYEE
```

# TRCANLZR von Oracle



Trace Analyzer ist über den Oracle Support erhältlich.

Die wichtigsten Vorteile gegenüber TKPROF sind:

- Aktueller Wert der Bindvariablen des Statements.
- Auflistung „heißer“ Blöcker, Optimizerstatistiken für Indices und Tabellen.
- Auteilung in user- und systemgenerierte rekursive Statements.
- Umfangreichere Informationen bei Wait Events.
- I. A. wesentlich ausführliches Reporting.

Im Gegensatz zu TKPROF benötigt Trace Analyzer Objekte in der Datenbank, d.h. für jede auszuwertende Datenbank muß eine Installation durchgeführt werden. Desweiteren ist die TKPROF schneller als Trace Analyzer, da weniger Auswertungen durchgeführt werden.

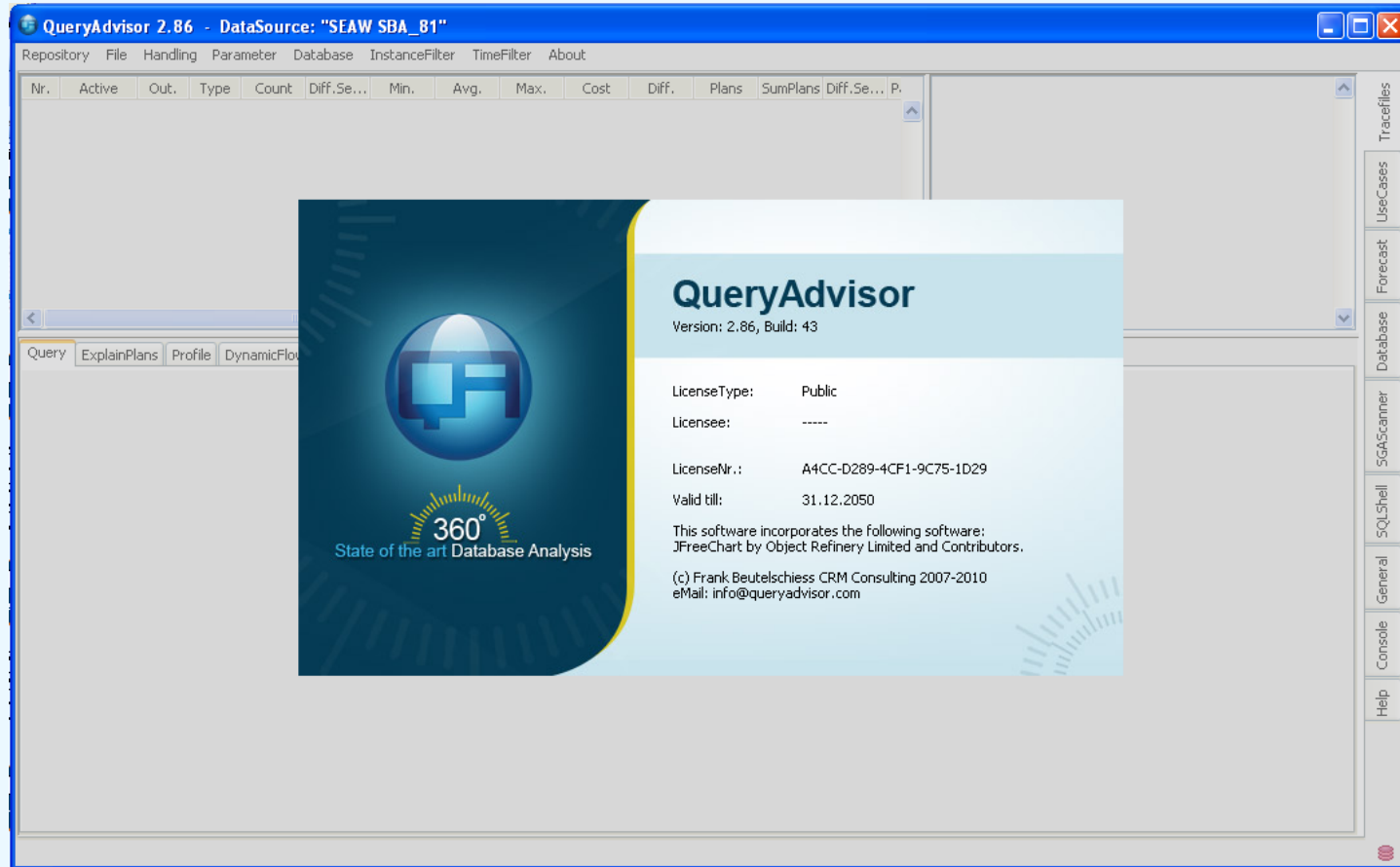
# Probleme & Anomalien



- Asymmetrisches Verhalten im RAC, d.h. die auf verschiedenen Knoten verarbeiteten Statements haben ein unterschiedliches Laufzeitverhalten.
- Sessionabhängig (verschiedene Ausführungspläne & Peek Binding).
- Releaseabhängig (verschiedene Umgebungen und Datenbankversionen).



# 3rd Party Tool QueryAdvisor



<http://www.QueryAdvisor.com> - Demonstration & Diskussion



QueryAdvisor soll einem umfangreichen Betatest unterzogen werden. Es werden noch Tester gesucht. Tester erhalten eine kostenfreie Entwicklerlizenz ohne Einschränkungen bei der Nutzung. Im Gegenzug wird um ein Feedback und eventuell auftretende Fehlerfälle gebeten. Der Betatest soll Anfang Mai mit dem kommenden Release beginnen.

Anmeldung bitte über das Kontaktformular „Contact Us“ auf <http://www.QueryAdvisor.com> oder auch direkt über <http://www.queryadvisor.com/globalPages/contactPage.html>.

Vielen Dank für das Interesse!