

# IT-Tage 2015

Schwerpunkt: Datenbanken  
14.12. - 18.12.2015  
Frankfurt am Main

Norbert Rieger:  
Optimierung der Performance bei Oracle-  
Datenbanken "nur" mit der Standard Edition



## Optimierung der Performance bei Oracle-Datenbanken "nur" mit der Standard Edition

IT-Tage Frankfurt 2015



MANAGED DATABASE SERVICES 24x7

Referent: Norbert Rieger  
Senior DBA bei dbaservices® GmbH in Neu-Isenburg





## Datenbank Administration ist unsere Kernkompetenz

- Seit 2001 (ehemals allnabout GmbH)
- Team mit 15 Mitarbeiter
- ORACLE- und MSSQL-Datenbanksysteme
- Beratung und Implementierung
- 24/7 Monitoring und Bereitschaft
- 3rd Level Support
- Managed Database Outtasking Modelle
- >800 Datenbankprojekte
- ITIL v.3
- ISO 9001 und 27001 i.V.



## Agenda

- **Optimierung Performance mit der Standard Edition**
- **Vor- und Nachteile der Tools**
- **Best Practice**



## Tools

- **Enterprise Manager**
- **AWR**
- **utlbstat/utlestat – Vorgänger vor 8.1.6 für statspack**
- **statspack – Periodische Snaps – Reports nach Bedarf**
- **Weitere Tools**



## Details

- **Enterprise Manager -> aber ohne Diagnostics und Tuning Pack!**
- **AWR -> nur mit Enterprise Edition**



## utlstat/utlestat

Die beiden SQL-Skripte befinden sich in dem Verzeichnis  
ORACLE\_HOME/rdbms/admin.

Die Analyse wird gestartet mit dem Skript utlstat.sql und wird beendet mit dem Skript utlestat.sql, welches den Report report.txt erstellt.



## Abschnitte im Report utlbstat/utlestat

- 1) Library cache Statistiken
- 2) Anzahl angemeldeter User
- 3) Statistiken wie CPU, DB TIME, db block gets etc.
- 4) Wait Events ohne/nur Hintergrundprozesse
- 5) Statistiken der Latche
- 6) Buffer busy wait Statistiken
- 7) Liste der init.ora-Parameter
- 8) Summe der I/O Operationen auf den Tablespaces





## Beispiele aus dem Report

### AD 1) Library cache Statistiken

18 LIBRARY	GETS	GETHITRATI	PINS	PINHITRATI	RELOADS	INVALIDATI
19 -----	-----	-----	-----	-----	-----	-----
20 SQL AREA	796	.991	2836	.993	0	0
21 TABLE/PROCED	130	1	656	1	0	0
22 BODY	17	1	260	1	0	0
23 TRIGGER	1	1	70	1	0	0
24 INDEX	0	1	0	1	0	0



## AD 3) Statistiken wie CPU, DB TIME, db block gets etc.

97 Statistic	Total	Per Transact	Per Logon	Per Second
98 -----	-----	-----	-----	-----
99 CPU used by this session	45	1.29	.16	1.15
100 CPU used when call started	35	1	.12	.9
101 CR blocks created	2	.06	.01	.05
102 DB time	9538	272.51	33.94	244.56



## statspack – Periodische Snaps – Reports nach Bedarf

**Bewertung:** Dieses Werkzeug ist für eine periodische maschinelle Analyse sehr nützlich. Es sind allerdings Vorarbeiten erforderlich.

**Beschreibung im Metalink:**

[https://docs.oracle.com/cd/B10500\\_01/server.920/a96533/statspac.htm](https://docs.oracle.com/cd/B10500_01/server.920/a96533/statspac.htm)

**Voraussetzungen:**

Ausführung des SQL-Skripts `@?/rdbms/admin/spcreate.sql`



## Installation: Das Skript spcreate.sql ruft folgende Skripte auf

1. SPCUSR.SQL: Legt den User an und teilt die Privilegien zu
2. SPCTAB.SQL: Erstellt die Tabellen
3. SPCPKG.SQL: Erstellt die Packages

### Empfehlung

Separaten Tablespace (perfstat) anlegen, damit dieser später leichter gelöscht werden kann.

Aufruf des Skripts im Batch (hier Linux):

```
SQL> CONNECT / AS SYSDBA
SQL> define default_tablespace='PERFSTAT'
SQL> define temporary_tablespace='TEMP'
SQL> define perfstat_password='<my_perfstat_password>'
SQL> @?/rdbms/admin/spcreate.sql
```



## Statistiken sammeln

Diese werden mit dem Befehl snap gesammelt. Der Aufruf kann manuell oder per Job erfolgen.

Sobald zwei Snaps ausgeführt wurden, kann ein Report erstellt werden.

```
SQL> CONNECT perfstat/<my_perfstat_password>  
SQL> EXECUTE statspack.snap;
```



# Report erstellen

Dieser wird mit dem SQL-Skript spreport.sql erstellt.

```
SQL> connect perfstat/<my_perfstat_password>  
SQL> @?/rdbms/admin/spreport.sql
```

Ablauf:

Output-1 -> Liste der Snaps

Input-1 -> <snap-start>

Input-2 -> <snap-ende>

Input-3 (optional) -> Reportname (default = sp\_<snap-start>\_<snap\_ende>.lst)

Output-2 – Anzeige/Speicherung Report



# Struktur eines Reports

**Die Struktur des Reports entspricht in weiten Teilen der Struktur des Reports, welcher mit utlbstat/utlestat erstellt wurde.**



## Weitere Tools

### Aktivierung von Traces

#### Metalink:

[http://docs.oracle.com/cd/E25054\\_01/server.1111/e16638/sqltrace.htm#autoId9](http://docs.oracle.com/cd/E25054_01/server.1111/e16638/sqltrace.htm#autoId9)

#### 1. Instanz-Ebene

```
EXECUTE DBMS_MONITOR.DATABASE_TRACE_ENABLE  
(waits => TRUE, binds => FALSE, instance_name => '<instance>');
```

```
EXECUTE DBMS_MONITOR.DATABASE_TRACE_DISABLE  
(instance_name => '<instance>');
```

Auswertung mit trcess und TKPROF





## 2. Session-Ebene

```
EXECUTE DBMS_MONITOR.SESSION_TRACE_ENABLE(session_id => <nr1>, serial_num  
=> <nr2>, waits => TRUE, binds => FALSE);  
EXECUTE DBMS_MONITOR.SESSION_TRACE_DISABLE(session_id => <nr1>, serial_num  
=> <nr2>);
```

## 3. SQL-Plus – autotrace

Metalink:

[http://docs.oracle.com/cd/B10500\\_01/server.920/a96533/autotrac.htm](http://docs.oracle.com/cd/B10500_01/server.920/a96533/autotrac.htm)

```
SET AUTOTRACE OFF  
SET AUTOTRACE ON {, EXPLAIN, STATISTICS}  
SET AUTOTRACE TRACEONLY
```



# Systemstatistiken

## 1.) Voraussetzungen

Parameter = STATISTICS\_LEVEL = TYPICAL oder ALL

## 2.) Erstellen

Package = DBMS\_STATS.GATHER\_SCHEMA\_STATS()

Link: [http://wikis.gm.fh-koeln.de/wiki\\_db/Datenbanken/Statistiken](http://wikis.gm.fh-koeln.de/wiki_db/Datenbanken/Statistiken)

## 3.) Auswerten

View: DBA\_TABLES (u.a.) – Spalte „ LAST\_ANALYZED „

## 4.) Dictionary-Views

DBA\_TABLES, DBA\_OBJECT\_TABLES, DBA\_TAB\_STATISTICS, DBA\_TAB\_COL\_STATISTICS u.a.



# Waits

## 1.) Wait-Events

View: V\$EVENT\_NAME

Link: [http://docs.oracle.com/cd/B19306\\_01/server.102/b14237/waitevents003.htm#BGGIBDJI](http://docs.oracle.com/cd/B19306_01/server.102/b14237/waitevents003.htm#BGGIBDJI)

## 2.) Wait-Klassen

Hinweis: Jeder Wait-Event gehört zu einer Wait-Klasse.

View: V\$SYSTEM\_WAIT\_CLASS

Link: [https://docs.oracle.com/cd/B19306\\_01/server.102/b14237/waitevents001.htm#BGGHJGII](https://docs.oracle.com/cd/B19306_01/server.102/b14237/waitevents001.htm#BGGHJGII)



# Optimizer – CBO – Ausführungsplan erstellen

## 1.) SQLPLUS (Beispiel)

```
set autotrace on  
SELECT name FROM v$database;
```

## 2.) Package (Beispiel)

```
EXPLAIN PLAN FOR SELECT name FROM v$database;  
SELECT PLAN_TABLE_OUTPUT FROM TABLE(DBMS_XPLAN.DISPLAY());
```



## Vielen Dank für die Aufmerksamkeit

Referent: Norbert Rieger, Senior DBA  
norbert.rieger@dbaservices.de  
www.dbaservices.de