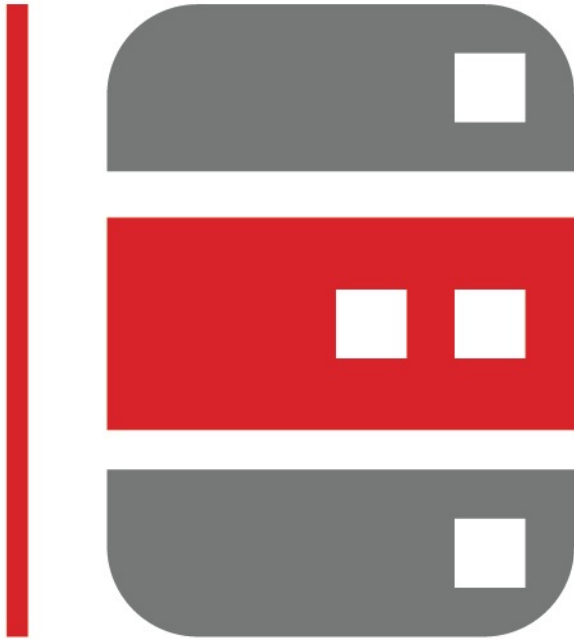


performing databases



Your reliability. Our concern.

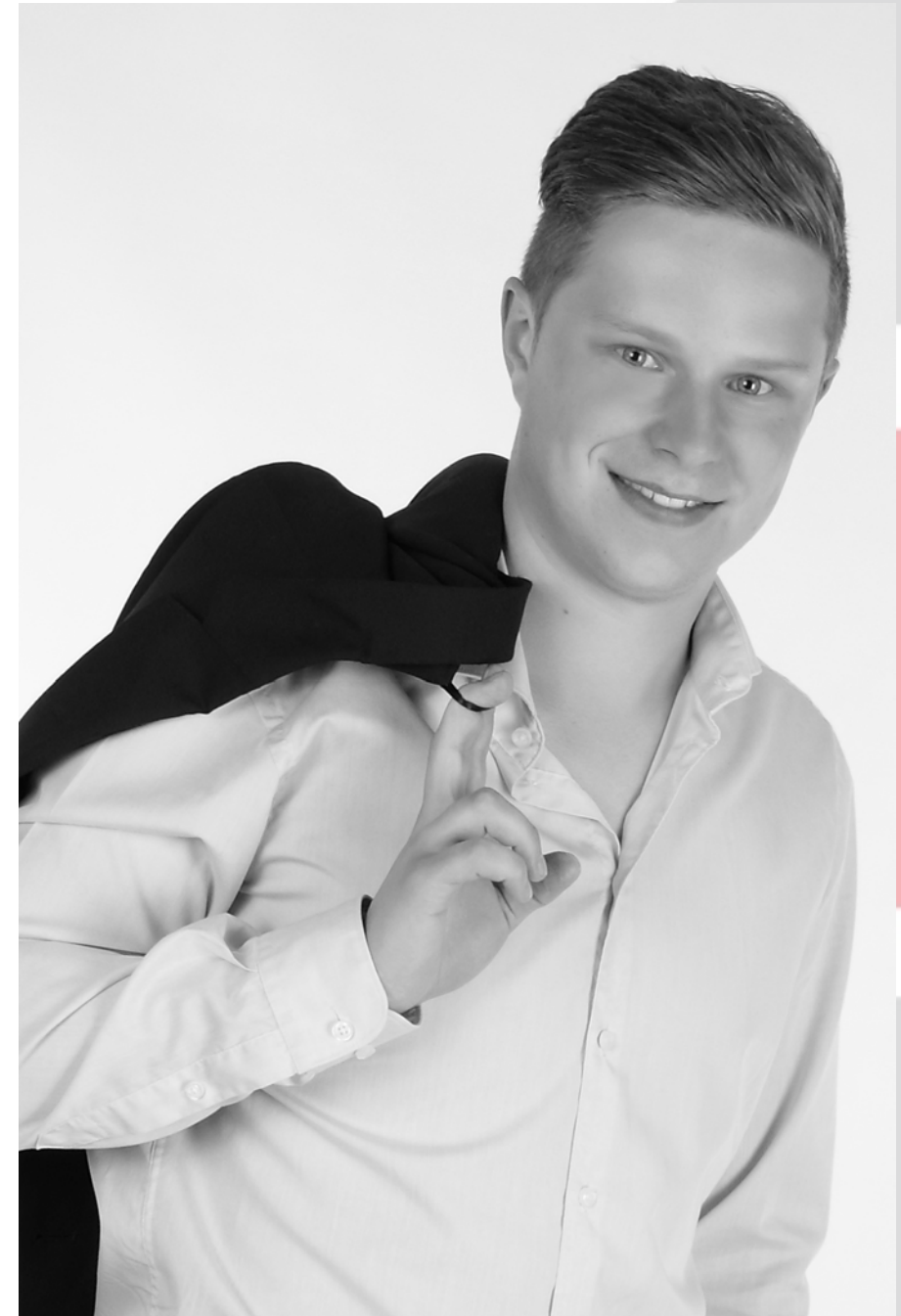
Kostenlos aber effektiv: Performanceanalyse mit Statspack und SQL*

Benedikt Nahlovsky

Performing Databases GmbH
Mitterteich

Referent

- Benedikt Nahlovsky
- Troubleshooter, Database integrator, Performance Firefighter
- Fachliche Schwerpunkte
 - Performanceoptimierung / Tuning
 - hochverfügbare Systeme
 - Cluster und Replikation
- Kontakt:
benedikt.nahlovsky@performing-db.com
- Weblog: <http://oradbn.wordpress.com>



Unternehmen

- Spezialisten für Datenbanktechnik
 - Konzeptberatung und Vergabekompetenz
 - Architektur- und Systemplanung
 - Lizenzierung
 - Realisierung und Troubleshooting
- Kontakt
 - Performing Databases GmbH
Wiesauer Straße 27
95666 Mitterteich
 - Web: <http://www.performing-databases.com>
 - Mail: info@performing-db.com
 - Twitter: @PerformingDB



Was ist Performance?

- Performance = Leistungserlebnis des Users
- Die Bewertung hängt von dem zu bewertenden System oder der einzelnen Komponente ab
- Performance ist mess- und vergleichbar



Oracle Statspack

- Kostenloses Tool zur Sammlung und Auswertung von Datenbankinformationen (seit Oracle 8i)
- (Performance-)Informationen werden über viele Bereiche der Oracle DB gesammelt
- Snapshots der V\$VIEWS
- Auswertung über verschiedene Zeitintervalle möglich
- Seit Oracle 10g: AWR (Automatic Workload Repository) —> kostenpflichtig (Enterprise Edition)

Statspack

Kompatibilität

- nicht abwärtskompatibel
- Dokumentation:

`$ORACLE_HOME/rdbms/admin/spdoc.txt`

Database Release	12.1	11.1	10.2	10.1	9.2	9.0	8.1.7	8.1.6
12.1	Y	-	-	-	-	-	-	-
11.1	-	Y	-	-	-	-	-	-
10.2	-	-	Y	-	-	-	-	-
10.1	-	-	-	Y	-	-	-	-
9.2	-	-	-	-	Y	-	-	-
9.0	-	-	-	-	-	Y	-	-
8.1.7	-	-	-	-	-	-	Y	-
8.1.6	-	-	-	-	-	-	-	Y

Statspack Historie

8.1.6 —> 8.1.7

- Load Profile (executions / transactions per second)
- Execute to Parse %
- sppurge.sql —> Purgen alter Snapshots

8.1.7 —> 9.0

- Latches und wait_time (cumulative spent waiting for the latch)
- Snap Level 6
- keine Installation von Statspack in SYSTEM Tablespace

9.0 —> 9.2

- Summary Page Instance Report
- Module Information bei SQL's
- Top 5 Wait Events —> Top 5 Timed Events

9.2 —> 10.1

- Report Intervall (Tage)
- Error Handling bei Reporterstellung
- SQL ordered by Cluster Wait Time (RAC - DB)
- SQL_ID in stats\$sql* hinzugefügt

10.1 —> 10.2

- Host CPU + Memory Configuration
- Avg Waits für Top 5 Timed Events
- Cache / Shared Pool Größen (Anfang / Ende) - wobei das Ende nur gefüllt wird, wenn eine Änderung im Report Intervall statt fand

10.2 —> 11.1

- Platform Name
- DB Time, DB CPU
- Physical Memory wird in GB angezeigt
- DB Time, DB CPU in Load Profile hinzugefügt
- PGA Memory Advisor zeigt erwartete Zeit für die Verarbeitung der Daten

11.1 —> 12.1

- Idle Events SQL*Net
- Statspack Installation für Pluggable Databases
- Installation und Reporting im Root Container nicht supported (CDB\$ROOT)

Statspack Installation

- Anlegen eines separaten PERFSTAT Tablespace
- Installation von Oracle Statspack

```
SQL> @?/rdbms/admin/spcreate.sql  
-> Passwort für PERFSTAT User eingeben  
-> Default Tablespace  
-> Temporary Tablespace
```


Snapshot

- Verbunden mit PERFSTAT
- Generieren eines Statspack Snapshot

```
SQL> execute statspack.snap;
```

```
PL/SQL procedure successfully completed.
```

```
Elapsed: 00:00:01.18
```

- Snapshot Level ändern (Default: 5)

```
SQL> execute statspack.snap(i_snap_level=>7);
```

Snapshot Level

- 0: Ausführungsstatistiken, grundlegende Events, SGA-Daten
- 5: Top SQL
- 6: SQL Pläne und Plan-Nutzungsstatistiken
- 7: Segmentstatistiken, Locks, Buffer Busy Waits
- 10: Erweiterte Latching Statistiken (Child Latches)

```
SQL> select * from stats$level_description;
```

Snapshot Job erstellen

- Datenbank Scheduler Job um Snapshot alle 10 Minuten zu erzeugen

```
SQL> conn / as sysdba
Connected.
SQL> BEGIN
sys.dbms_scheduler.create_job(
job_name => '"PERFSTAT"."PERFSTAT_SNAP"',
job_type => 'PLSQL_BLOCK', job_action => ' begin

execute immediate ('Alter session set "_cursor_plan_unparse_enabled" =
false'); statspack.snap(i_snap_level => 7, i_modify_parameter => 'false');
end;',

repeat_interval => 'FREQ=MINUTELY; INTERVAL=10;', start_date => systimestamp at
time zone 'Europe/Berlin', job_class => 'DEFAULT_JOB_CLASS',
comments => 'Perfstat snapshot job',
auto_drop => FALSE,
enabled => TRUE); END;
/
```

Useful Queries

```
SQL> select name,snap_id,to_char(snap_time,'DD.MM.YYYY HH24:MI:SS') "Date/Time"  
      from stats$snapshot,v$database  
      where snap_time > sysdate -1 order by 3 desc;
```

NAME	SNAP_ID	Date/Time
-----	-----	-----
ORAVM11	174	11.11.2014 21:26:12
ORAVM11	173	11.11.2014 21:16:12
ORAVM11	172	11.11.2014 21:06:12
ORAVM11	171	11.11.2014 20:56:12
ORAVM11	170	11.11.2014 20:46:12
ORAVM11	169	11.11.2014 20:36:12
ORAVM11	168	11.11.2014 20:26:12
ORAVM11	167	11.11.2014 20:16:12

Useful Queries

```
SQL> select dbid,to_char(startup_time,'DD.MM.YYYY HH24:MI:SS')"Startup Time",  
instance_name, host_name  
from stats$database_instance  
where startup_time > sysdate - 5 order by startup_time;
```

DBID	Startup Time	INSTANCE_NAME	HOST_NAME
----	-----	-----	-----
3111692151	10.11.2014 20:24:08	ORAVM11	oravm11gR2
3111692151	11.11.2014 10:03:15	ORAVM11	oravm11gR2
3111692151	11.11.2014 17:38:34	ORAVM11	oravm11gR2

Statspack Report

- Statspack Report generieren

```
SQL> @?/rdbms/admin/spreport.sql
```

- Report Intervall wählen

Listing all Completed Snapshots

Instance	DB Name	Snap Id	Snap Started	Snap Level	Comment
ORAVM11	ORAVM11	1	04 Nov 2014 18:06	5	
		2	04 Nov 2014 18:16	5	
		3	04 Nov 2014 18:26	5	

Specify the Begin and End Snapshot Ids

~~~~~

Enter value for begin\_snap: 1

Begin Snapshot Id specified: 1

Enter value for end\_snap: 3

# Report

STATSPACK report for

| Database | DB Id      | Instance | Inst Num | Startup Time    | Release    | RAC |
|----------|------------|----------|----------|-----------------|------------|-----|
| ~~~~~    | -----      | -----    | -----    | -----           | -----      | --- |
|          | 3342991817 | PERFDB   | 1        | 28-Oct-14 15:21 | 11.2.0.3.0 | NO  |

| Host Name | Platform         | CPUs  | Cores | Sockets | Memory (G) |
|-----------|------------------|-------|-------|---------|------------|
| ~~~~~     | -----            | ----- | ----- | -----   | -----      |
| perfdbvm  | Linux x86 64-bit | 12    | 6     | 1       | 31.5       |



# Report

| Load Profile      | Per Second | Per Transaction | Per Exec | Per Call |
|-------------------|------------|-----------------|----------|----------|
| ~~~~~             | -----      | -----           | -----    | -----    |
| DB time(s):       | 0.2        | 0.0             | 0.00     | 0.00     |
| DB CPU(s):        | 0.2        | 0.0             | 0.00     | 0.00     |
| Redo size:        | 152,213.9  | 9,602.6         |          |          |
| Logical reads:    | 15,600.0   | 984.1           |          |          |
| Block changes:    | 914.2      | 57.7            |          |          |
| Physical reads:   | 0.7        | 0.0             |          |          |
| Physical writes:  | 10.6       | 0.7             |          |          |
| User calls:       | 549.4      | 34.7            |          |          |
| Parses:           | 1.4        | 0.1             |          |          |
| Hard parses:      | 0.0        | 0.0             |          |          |
| W/A MB processed: | 29.8       | 1.9             |          |          |
| Logons:           | 0.1        | 0.0             |          |          |
| Executes:         | 219.0      | 13.8            |          |          |
| Rollbacks:        | 8.2        | 0.5             |          |          |
| Transactions:     | 15.9       |                 |          |          |



# Report

## Instance Efficiency Indicators

~~~~~

Buffer Nowait %:	100.00	Redo NoWait %:	100.00
Buffer Hit %:	100.00	Optimal W/A Exec %:	100.00
Library Hit %:	100.00	Soft Parse %:	100.00
Execute to Parse %:	99.35	Latch Hit %:	99.74
Parse CPU to Parse Elapsed %:	25.74	% Non-Parse CPU:	99.39

Report

Top 5 Timed Events

~~~~~

| Event                        | Waits  | Time (s) | Avg wait (ms) | %Total Call Time |
|------------------------------|--------|----------|---------------|------------------|
| CPU time                     |        | 457      |               | 75.4             |
| log file sync                | 27,728 | 53       | 2             | 8.7              |
| log file parallel write      | 28,104 | 51       | 2             | 8.5              |
| db file parallel write       | 4,240  | 13       | 3             | 2.2              |
| control file sequential read | 10,512 | 10       | 1             | 1.7              |

## Host CPU (CPUs: 12 Cores: 6 Sockets: 1)

~~~~~

Load Average

Begin	End	User	System	Idle	WIO	WCPU
0.14	0.09	12.36	2.85	84.20	0.00	

SQL ordered by CPU

SQL ordered by CPU DB/Inst: PERFDB/PERFDB Snaps: 166632-166638

-> Total DB CPU (s): 678

-> Captured SQL accounts for 63.0% of Total DB CPU

-> SQL reported below exceeded 1.0% of Total DB CPU

CPU Time (s)	Executions	CPU per Exec (s)	%Total	Elapsd Time (s)	Buffer Gets	Old Hash Value
312.91	25,894	0.01	46.2	325.03	47,815,839	2208385654

Module: oracle@perfdbvm (TNS V1-V3)

```
SELECT "A1"."DESCRIPTION", "A1"."ONBUFFER", "A1"."ONTHEWAY" FROM
(SELECT MIN("A2"."DESCRIPTION") "DESCRIPTION", SUM("A2"."ONBUFFER
") "ONBUFFER", SUM("A2"."ONTHEWAY") "ONTHEWAY" FROM ( (SELECT MI
N("A6"."DESCRIPTION") "DESCRIPTION", COUNT(*) "ONBUFFER", 0 "ONTHE
```

SQL ordered by

- SQL ordered by CPU
- SQL ordered by Elapsed Time
- SQL ordered by Gets
- SQL ordered by Reads
- SQL ordered by Executions
- SQL ordered by Parse Calls
- SQL ordered by Sharable Memory

Purge Job erstellen

- Datenbank Scheduler Job für Purgen der alten Snapshots

```
SQL> BEGIN
sys.dbms_scheduler.create_job(
job_name => '"PERFSTAT"."PERFSTAT_CLEANUP"',

job_type => 'PLSQL_BLOCK', job_action => 'begin
execute immediate 'delete from stats$snapshot where snap_time < sysdate
-7''; commit; end;',

repeat_interval => 'FREQ=DAILY;BYHOUR=22;BYMINUTE=0;BYSECOND=0', start_date
=> systimestamp at time zone 'Europe/Berlin',
job_class => 'DEFAULT_JOB_CLASS',
comments => 'Perfstat cleanup 5 days history',
auto_drop => FALSE,
enabled => TRUE); END;
/
```

0800 - DBA



5 W - Fragen

- Welches Projekt?
- Welche Beschwerden?
- Welcher Zeitraum?
- Welche Auswirkungen?
- Wer ist der Ansprechpartner für Rückfragen?

Analyse

Host Name	Platform	CPUs	Cores	Sockets	Memory (G)
~~~~~	-----	-----	-----	-----	-----
oravm11gR2	Linux x86 64-bit	2	2	1	3.9

Snapshot	Snap Id	Snap Time	Sessions	Curs/Sess	Comment
~~~~~	-----	-----	-----	-----	-----
Begin Snap:	42	05-Nov-14 19:16:12	34	1.2	
End Snap:	43	05-Nov-14 19:26:12	35	1.2	

Elapsed:	10.00 (mins)	Av Act Sess:	0.9
DB time:	9.08 (mins)	DB CPU:	0.02 (mins)

Analyse

Top 5 Timed Events

~~~~~

| Event                         | Waits | Time (s) | Avg wait (ms) | %Total Call Time |
|-------------------------------|-------|----------|---------------|------------------|
| enq: TX - row lock contention | 1     | 543      | #####         | 99.3             |
| CPU time                      |       | 2        |               | .3               |
| control file sequential read  | 1,018 | 1        | 1             | .1               |
| db file parallel write        | 414   | 0        | 1             | .1               |
| os thread startup             | 8     | 0        | 42            | .1               |

# Lock Analyse

```
SQL> select s1.username || '@' || s1.machine || ' ( SID=' || s1.sid || ' ) is  
blocking  
2   ' || s2.username || '@' || s2.machine || ' ( SID=' || s2.sid || ' ) ' AS  
blocking_status  
3   from v$lock l1, v$session s1,  
4   v$lock l2, v$session s2  
5   where s1.sid=l1.sid and s2.sid=l2.sid  
6   and l1.BLOCK=1 and l2.request > 0  
7   and l1.id1 = l2.id1  
8   and l2.id2 = l2.id2  
9   order by s1.sid, s2.sid  
10  /
```

BLOCKING\_STATUS

-----  
HR@oravm11gR2 ( SID=10 ) is blocking HR@oravm11gR2 ( SID=20 )

**alter system kill session (sid, serial#) immediate;**

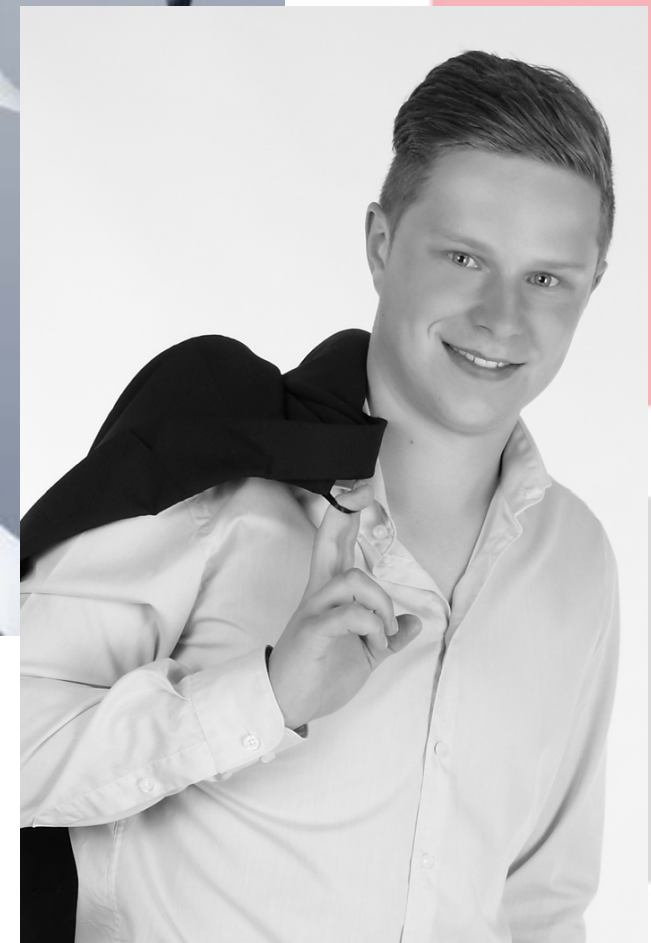
# Locks

- Locks (Sperren) verhindern, dass mehrere Benutzer zur gleichen Zeit die gleiche Zeile einer Tabelle ändern
- Innerhalb einer Transaktion können X Zeilen, sowie ganze Tabellen gesperrt werden
- Jeder Benutzer kann auch manuell eine Sperre auf Tabellen Ebene setzen

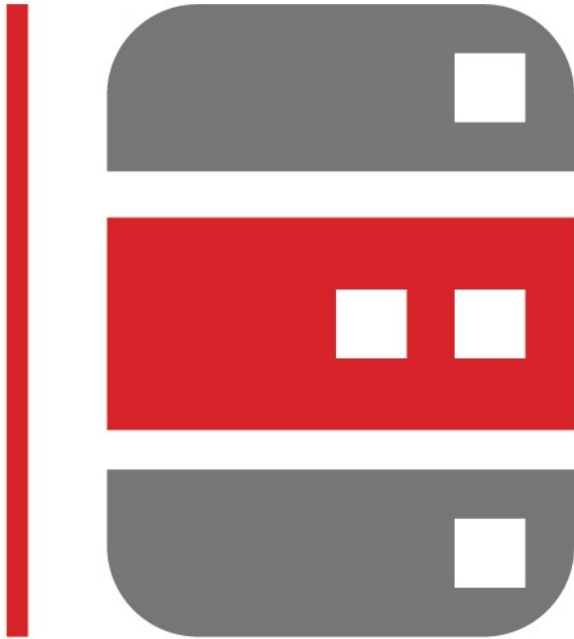
# ***Live Demo***



[info@performing-db.com](mailto:info@performing-db.com)



# performing databases



Your reliability. Our concern.