

# Kostenlos aber effektiv: Performanceanalyse mit Statspack und SQL\*

**Benedikt Nahlovsky**  
**Performing Databases GmbH**  
**Mitterteich, Germany**

## **Schlüsselworte**

Performance, Statspack, Problemaufnahme (5 W-Fragen), Problemanalyse, Informationsbeschaffung, Blocking Locks, Tuning Maßnahmen, Live Demo, Fazit

## **Einleitung**

Viele Möglichkeiten, welche die Oracle Datenbank zur Analyse bereitstellt sind nicht umsonst zu bekommen und sprengen oft für kleine und mittelständische Unternehmen das Budget. Im Vortrag wird gezeigt, wie man auch mit kostenlos zur Verfügung stehenden Tools Probleme analysiert und diese löst.

## **Was ist Performance?**

Performance ist in der Datenbanktechnik gleichzusetzen mit der Leistungsfähigkeit. Die Bewertung von Performance hängt vom System oder der einzelnen Komponente ab. Performance ist mess- und vergleichbar.

## **Statspack**

Um die Effizienz bzw. Verfügbarkeit einer Datenbank zu überwachen, oder um bereits vorhandene Ressourcenengpässe zu lokalisieren, stellt Oracle ein kostenloses Tool zur Sammlung und Auswertung von Datenbankinformationen für die Performanceoptimierung zur Verfügung.

Mit Oracle Statspack können (Performance-)Informationen über viele Bereiche einer Oracle Datenbank gesammelt werden, um diese dann zu einem späteren Zeitpunkt auswerten zu können.

Seit der Oracle Database Version 10g gibt es zusätzlich zu Statspack auch noch ein weiteres Tool für diese Aufgaben, das Automatic Workload Repository (AWR). Der große Nachteil von AWR ist, dass es im Gegensatz zu Statspack kostenpflichtig ist und ausschließlich auf einer Oracle Enterprise Edition lizenzierbar ist.

## **Statspack Installation**

Bei der Installation von Oracle Statspack wird das neue Schema PERFSTAT in der Datenbank angelegt. Diesem Benutzer ist ein Default und ein Temporärer Tablespace zugewiesen. Hier empfehlen wir Ihnen, einen separaten Tablespace zu verwenden, da nach dem Löschen von Statspack und des PERFSTAT Schemas dieser Tablespace auch einfach gelöscht werden kann.

Nachdem ein eigener Tablespace angelegt wurde, kann mit der STATSPACK Installation als SYS begonnen werden:

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

Im Verlauf der Installation müssen Sie ein Passwort für den PERFSTAT Benutzer und die beiden zu verwendenden Tablespaces angeben.

Nach der erfolgreichen Installation sind Sie als PERFSTAT mit der Datenbank verbunden und so können Sie sofort einen ersten Snapshot durchführen.

```
SQL > EXECUTE statspack.snap;
```

Oder z.B. mit einem anderen Snapshot - Level (Default: 5)

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

### **Erstellen eines automatischen Snapshot Job**

Das unten aufgeführte Skript, erstellt einen Datenbank Scheduler Job, welcher im Intervall von 10 Minuten einen Snapshot mit Level 5 durchführt.

```
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 => 5, 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;
/
```

### **Statspack Report erzeugen**

Die nun gesammelten Statistiken können nun anhand des von Oracle zur Verfügung gestellten Skripts ausgewertet werden:

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

Nach dem Aufruf dieses Skripts werden alle zur Verfügung stehenden Snapshots aufgelistet. Dann wird zur Eingabe der Snap-ID's aufgefordert, über die ein Report erzeugt werden soll. Der Default Dateiname sp\_<begin\_snap\_id>\_<end\_snap\_id>.lst kann optional geändert werden.

### **Statspack Purge Job**

Um sicherzustellen, dass der neu angelegte PERFSTAT Tablespace nicht sinnlos wächst, sollten die erstellten Snapshots nur für eine bestimmte Zeit aufbewahrt werden. In unserem Beispiel werden die Snapshots im PERFSTAT Tablespace für 7 Tage aufbewahrt. Auch hierfür wird empfohlen, einen Datenbank Scheduler Job zu erstellen:

```
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;  
/
```

### **Problemaufnahme**

- Wann?
- Was?
- Welche Auswirkungen?
- Welches Projekt?
- Wer ist der Ansprechpartner für Rückfragen?

### **Informationsbeschaffung**

Die im STATSPACK gespeicherten Informationen helfen, das gemeldete Problem zu lokalisieren. Die Herangehensweise zur Analyse eines Statspack Reports wird gezeigt. Im Report ist zu finden, das ein ausgeführtes SQL Statement in der Datenbank viel Last verursacht, dieses Statement wird lokalisiert. Desweiteren zeigt der Report verschiedene Waits, darauf lässt sich schließen, das es möglicherweise eine Session in der Datenbank gibt welche nicht abgeschlossen (commit) wird.

### **Execution Plan from Cache generieren**

Im Statspack Report lässt sich das Problem SQL Statement finden. Diese Statements sind mit einem eindeutigen OLD\_HASH\_VALUE identifizierbar. Anhand eines Skripts lässt sich der Ausführungsplan des Statements anzeigen. Nach der Analyse wird gezeigt, wie das SQL Statement in im Ressourcenverbrauch und der Laufzeit verbessert werden kann.

```
ALTER SESSION SET "_cursor_plan_unparse_enabled"=TRUE;
```

```
define OHV=  
undefine CHILD_NUMBER
```

```
select SQL_ID from v$SQLAREA where old_hash_value=&&OHV;
```

```
select inst_id, child_number, plan_hash_value  
from gv$sql_plan  
where SQL_ID=(select SQL_ID from v$SQLAREA where old_hash_value=&&OHV)  
group by inst_id, child_number, plan_hash_value  
order by inst_id, child_number  
;
```

```
select * from gv$sqlarea where sql_id=(select SQL_ID from v$SQLAREA where  
old_hash_value=&&OHV);
```

```
select inst_id, plan_hash_value, executions, buffer_gets, round(buffer_gets/nvl(executions,0),0) as  
BG_average from gv$sql  
where sql_id=(select SQL_ID from v$SQLAREA where old_hash_value=&&OHV) order by  
inst_id,plan_hash_value;
```

```
select 'exec sys.dbms_shared_pool.purge ("||address||','||hash_value||', "C");' from gv$sql where  
sql_id=(select SQL_ID from v$SQLAREA where old_hash_value=&&OHV);
```

```
SELECT * FROM table(DBMS_XPLAN.DISPLAY_CURSOR((select SQL_ID from v$SQLAREA where  
old_hash_value=&&OHV),&&CHILD_NUMBER,'COST,IOSTATS,LAST,ADVANCED'));
```

## Blocking Session identifizieren

Mit Hilfe eines Selects auf die Views V\$LOCK und V\$SESSION lässt sich die blockierende Session sowie die wartende Session identifizieren:

```
SQL> select s1.username || '@' || s1.machine || ' (SID=' || s1.sid || ') is blocking ' || s2.username || '@' || s2.machine || ' (SID=' || s2.sid || ') ' AS blocking_status
from v$lock l1, v$session s1, v$lock l2, v$session s2
where s1.sid=l1.sid and s2.sid=l2.sid
and l1.BLOCK=1 and l2.request > 0
and l1.id1 = l2.id1 and l2.id2 = l2.id2
order by s1.sid, s2.sid
;
```

## Fazit

Mit Hilfe von kostenlos zur Verfügung stehenden Tools wie beispielsweise STATSPACK werden Datenbankinformationen gespeichert und können später über einen Report ausgewertet werden. Sie dienen somit als Stütze zur Problemlösung für Datenbankadministratoren. Auch ohne Oracle Enterprise Edition mit Diagnostic und Tuning Pack lassen sich Probleme in der Datenbank lösen. Gerade für kleine und mittelständische Unternehmen ist dies oftmals der bevorzugte Weg um zu Hohe Lizenzkosten aus dem Weg zu gehen.

## Autor

Nach einer zweijährigen Beschäftigung als Java-Entwickler ist Benedikt Nahlovsky seit nun mehr als 5 Jahren als Administrator in der Welt der Datenbanktechnik tätig.

Probleme zu analysieren, neue technische Funktionen zu testen und sein Wissen jeden Tag zu erweitern begeistert ihn, was auch die Gründe für die berufliche Herausforderung Datenbankadministrator waren.

Seine Expertise konzentriert sich auf die Bereiche Systemsizing, Lizenzberatung, Installation / Konfiguration von Single-Instance Datenbanken mit Oracle Restart, sowie die Hochverfügbarkeitslösungen RAC und Data Guard. Benedikt konnte seine fachlichen Kenntnisse bei verschiedenen Migrationen unter Beweis stellen. Besonders engagiert er sich gerne für das proaktive und reaktive Performancetuning.

Durch die Begeisterung für das Beseitigen von Problemen berichtet Benedikt Nahlovsky die Lösungswege immer wieder auf seinem Blog <http://oradbn.wordpress.com>. Folgen Sie @bnahlovsky auf Twitter um mehr über sein Daily Life als Datenbankadministrator zu erfahren. Desweiteren ist er in verschiedenen sozialen Netzwerken wie beispielsweise Xing vertreten.

## Kontaktdaten:

Benedikt Nahlovsky  
managing partner / database technology  
Performing Databases GmbH  
Wiesauer Straße, 27  
D-95666 Mitterteich



Your reliability. Our concern.

Telefon: +49 (0) 9633 631  
Mobil: +49 (0) 170 7373326  
Fax: +49 (0) 9633 4199  
E-Mail: [benedikt.nahlovsky@performing-db.com](mailto:benedikt.nahlovsky@performing-db.com)  
Internet: <http://www.performing-databases.com>

Blog: <http://oradbn.wordpress.com>