

Wie misst Oracle die Verwendung von Database Features?

Martin Berger

Wien

Schlüsselworte

Oracle Features, DBA_FEATURE_USAGE_STATISTICS

Einleitung

Die wohl bekannteste Quelle zur Überprüfung der genutzten Features einer Datenbank ist die View DBA_FEATURE_USAGE_STATISTICS. Leider fällt die Dokumentation dazu reichlich bescheiden aus. Weder ist klar, was genau gemessen wird, noch wann oder wie.

Die aktuelle Dokumentation der Version 12.1 beschränkt sich auf den simplen Satz

DBA_FEATURE_USAGE_STATISTICS displays information about database feature usage statistics.¹

Auch zu beachten ist, dass die View DBA_FEATURE_USAGE_STATISTICS in der aktuellsten Oracle® Database Licensing Information 12c Release 1 (12.1) - E49208-11² keinerlei Erwähnung findet.

Trotzdem bietet die View DBA_FEATURE_USAGE_STATISTICS einiges mehr an Informationen, die für einen DBA von Nutzen sein können.

Auch muss man – mit dem entsprechenden Wissen ausgestattet – nicht warten, bis der Inhalt der View aktualisiert wird. Mit Wissen über die Hintergründe kann man sehr gezielt selbst die Nutzung von Features überprüfen.

Die View DBA_FEATURE_USAGE_STATISTICS und die Tabellen dahinter

Die View-definition von DBA_FEATURE_USAGE_STATISTICS ist :

```
select samp.dbid,
       fu.name,
       samp.version,
       detected_usages,
       total_samples,
       decode(to_char(last_usage_date, 'MM/DD/YYYY, HH:MI:SS'),
              NULL, 'FALSE',
              to_char(last_sample_date, 'MM/DD/YYYY, HH:MI:SS'),
              'TRUE', 'FALSE')
       currently_used,
```

¹<http://docs.oracle.com/database/121/REFRN/GUID-B8A38B9B-3AAA-4341-AA05-4309870CE73B.htm>

² <http://docs.oracle.com/database/121/DBLIC/toc.htm>

```

        first_usage_date,
        last_usage_date,
        aux_count,
        feature_info,
        last_sample_date,
        last_sample_period,
        sample_interval,
        mt.description
from wri$dbu_usage_sample samp,
     wri$dbu_feature_usage fu,
     wri$dbu_feature_metadata mt
where
  samp.dbid      = fu.dbid and
  samp.version   = fu.version and
  fu.name        = mt.name and
  fu.name not like '_DBFUS_TEST%' and /* filter out test features */
  bitand(mt.usg_det_method, 4) != 4 /* filter out disabled features */

```

Die Daten selbst werden also in den Tabellen `wri$dbu_usage_sample`, `wri$dbu_feature_usage` und `wri$dbu_feature_metadata` gespeichert. Die Filter beziehen sich nur auf test-features (`_DBFUS_TEST%`) bzw. Features, deren Test als „disabled“ markiert sind (`bitand(mt.usg_det_method, 4) != 4`).

Die Projektion der einzelnen Spalten ist größtenteils sehr einfach und leserlich gehalten. Hier die Liste der Spaltennamen der View, und von welcher Tabelle sie stammen:

```

DBA_FEATURE_USAGE_STATISTICS
DBID                samp.dbid
NAME                fu.name
VERSION             samp.version
DETECTED_USAGES     fu.detected_usages
TOTAL_SAMPLES       samp.total_samples
CURRENTLY_USED      decode(fu.last_usage_date, NULL, FALSE,
                          samp.last_sample_date, TRUE, FALSE)

FIRST_USAGE_DATE    fu.first_usage_date
LAST_USAGE_DATE     fu.last_usage_date
AUX_COUNT           fu.aux_count
FEATURE_INFO        fu.feature_info
LAST_SAMPLE_DATE    samp.last_sample_date
LAST_SAMPLE_PERIOD  samp.last_sample_period
SAMPLE_INTERVAL     samp.sample_interval
DESCRIPTION         mt.description

```

Viel interessanter als die sichtbaren Spalten sind natürlich immer diejenigen, welche verborgen geblieben sind. Das sind – mit den jeweiligen Tabellennamen und alias:

```

wri$dbu_usage_sample samp
LAST_SAMPLE_DATE_NUM

```

```

wri$dbu_feature_usage fu
DBID
VERSION
ERROR_COUNT

```

```

wri$dbu_feature_metadata mt
NAME

```

INST_CHK_METHOD
INST_CHK_LOGIC
USG_DET_METHOD
USG_DET_LOGIC

Die ersten beiden Tabellen enthalten recht wenig interessante zusätzliche Informationen.

Wir%_dbu_feature_metadata allerdings hat 2 zusätzliche Gruppen: INST_CHK_% und USG_DET_%
Mit der in dem LOB INST_CHK_METHOD beschriebenen Logik wird überprüft, ob das jeweilige Feature überhaupt installiert ist, und deshalb die Verwendung überprüft werden soll.

INST_CHK_LOGIC gibt an, welche Art der Methode bei dieser Überprüfung verwendet werden soll.

Die derzeit gültigen Werte sind

DBU_INST_ALWAYS_INSTALLED	1
DBU_INST_OBJECT	2
DBU_INST_TEST	4

Wobei der Wert 1 signalisiert, dass der Test immer durchgeführt werden soll. Beim Wert 2 wird die Existenz eines in INST_CHK_LOGIC angegebenen Objekts überprüft. Der Wert 4 scheint nur für interne Tests in Verwendung zu sein.

Die Werte für die Überprüfungsmethode selbst sind in USG_DET_METHOD angegeben:

DBU_DETECT_BY_SQL	1
DBU_DETECT_BY_PROCEDURE	2
DBU_DETECT_NULL	4

Hier stehen 1 und 2 recht offensichtlich für ein konkretes SQL-Statement oder eine Prozedur, die in USG_DET_LOGIC hinterlegt sind. 4 ist auch hier ein Wert, der nur in internen Test-Prozeduren verwendet wird (in diesem Fall nur _DBFUS_TEST_SQL_7).

Andere Objekte mit Beziehungen zu wir\$_dbu% und DBA_FEATURE_USAGE_STATISTICS

War der Weg von der View DBA_FEATURE_USAGE_STATISTICS zu den darunter liegenden Tabellen geradezu trivial, ist auch der Rückschluss auf andere Objekte, die mit den wir\$_dbu% Tabellen in Kontakt stehen, einfach machbar: über DBA_DEPENDENCIES ergeben sich diese abhängigen Objekte:

DBA_FEATURE_USAGE_STATISTICS

DBA_FEATURE_USAGE_STATISTICS	SYNONYM
DBMS_SWRP_REPORT_INTERNAL	PACKAGE BODY
DBMS_FEATURE_%	PROCEDURE

wri\$_dbu_usage_sample

DBA_FEATURE_USAGE_STATISTICS	VIEW
DBMS_FEATURE_AWR	PROCEDURE
DBMS_FEATURE_DMU	PROCEDURE
DBMS_FEATURE_USAGE_INTERNAL	PACKAGE BODY

wri\$_dbu_feature_usage

DBA_FEATURE_USAGE_STATISTICS	VIEW
DBMS_FEATURE_USAGE	PACKAGE BODY
DBMS_FEATURE_USAGE_INTERNAL	PACKAGE BODY

wri\$_dbu_feature_metadata

DBA_FEATURE_USAGE_STATISTICS	VIEW
DBMS_FEATURE_USAGE	PACKAGE BODY
DBMS_FEATURE_USAGE_INTERNAL	PACKAGE BODY

Leider sind die – wohl sehr interessanten – Packages `DBMS_FEATURE_USAGE` und `DBMS_FEATURE_USAGE_INTERNAL` von Oracle nicht dokumentiert und durch Wrappen auch nicht ohne weiteres lesbar.

Allerdings sind wenigstens einige Methoden der Packages von aussen aufrufbar:

DBMS_FEATURE_USAGE

```
REGISTER_DB_FEATURE  
REGISTER_HIGH_WATER_MARK
```

DBMS_FEATURE_USAGE_INTERNAL

```
CLEANUP_DATABASE  
EXEC_DB_USAGE_SAMPLING  
SAMPLE_ONE_FEATURE  
SAMPLE_ONE_HWM
```

Deren genaue Funktion bleibt leider unbekannt, doch geben sie einen groben Einblick in die Möglichkeiten dieser Packages.

Weitere Informationen

Durch umfangreiche Recherche zu den sich bis hierher ergebenden Schlüsselwörtern kann doch einiges an Information zusammengetragen werden.

Metalink Note

How to Manually Refresh Dbafeature_usage_statistics (Doc ID 1629485.1)³

Zeigt, dass man mit den `mmon` zu einem aussergewöhnlichen Check zwingen kann:

```
alter session set events 'immediate trace name mmon_test level 6';
```

Viel interessanter ist aber ein

```
alter session set "_SWRF_TEST_ACTION"=53;
```

mit dem die aktuelle Session die Tests ausführt. Neugierige können dabei zum Beispiel mit einem `SQL_TRACE` der Session selbst den genauen Ablauf nachverfolgen. Interessant ist dabei, dass die Spalte `SAMPLE_INTERVAL` in `wri$_dbu_usage_sample` sich dabei nie ändert.

In Note

ORA-12751 "cpu time or run time policy violation" in the MMON Slave Process File During Feature Usage Statistics [FUS] Collection by SYS.DBMS_FEATURE_USAGE_INTERNAL (Doc ID 1291296.1)⁴

Wird sehr schön gezeigt, wie auch die Aktivitäten des `mmon` für das Reporten der Feature Usage getraced werden können:

```
begin  
    dbms_monitor.serv_mod_act_trace_enable(service_name=>'SYS$BACKGROUND',  
        module_name=>'MMON_SLAVE',  
        action_name=>'Auto-DBFUS Action');  
end;  
/
```

Sogar wie die Überprüfung einzelner Features disabled werden kann ist dort beschrieben. Das sollte aber mithilfe dieses Dokuments auch leicht möglich sein.

³ <https://support-lite.oracle.com/kmlite/faces/DocumentDisplay?id=1629485.1>

⁴ <https://support-lite.oracle.com/kmlite/faces/DocumentDisplay?id=1291296.1>

Es gibt auch zwei hidden spfile Parameter, die auf DBFUS hinweisen:⁵

Parameter	SessionValue	InstanceValue	Description
__swrf_mmon_dbfus	TRUE	TRUE	Enable/disable SWRF MMON DB Feature Usage
__swrf_test_dbfus	FALSE	FALSE	Enable/disable DB Feature Usage Testing

Diese zu verändern ist aber nur auf ausgewiesenen Sandbox Systemen oder nach Rücksprache mit Oracle Support zu empfehlen!

An einem Beispiel zeigt sich recht schön, welche Informationen sich hier eröffnen:

Das Feature „Real Application Clusters (RAC)“ wird durch die Prozedur `DBMS_FEATURE_RAC` überprüft. Dabei wird in `v$system_parameter` auf `cluster_database=TRUE` überprüft. „Real Application Cluster One Node“ allerdings wird in der Prozedur `DBMS_FEATURE_ROND` ausgewertet. Dabei wird auf `x$kjidt` zugegriffen. Leider auch hier undokumentiert.

Dieses kleine Beispiel zeigt, dass das Lesen der jeweiligen Prüfmethode einige zusätzliche Informationen zutage fördern und mitunter auch zu weiteren Nachforschungen anregen kann.

Und die Lizenzen?

Wie am Anfang erwähnt gibt es in der Dokumentation keine direkte Verbindung zwischen Der View `DBA_FEATURE_USAGE_STATISTICS` und dem License Guide.

Allerdings gibt es die Support Note

Database Options/Management Packs Usage Reporting for Oracle Databases 11gR2 and 12c (Doc ID 1317265.1)⁶

Mit dem Sehr ambitionierten Ziel

This note describes how to check the usage of separately licensed Oracle Database Options/Management Packs.

Allerdings wird einige Zeilen später schon relativiert:

Please review the Description & Disclaimer section in the script before executing the script.

Kindly note the report generated is to be used for informational purposes only and this **does not represent your license entitlement or requirement**. for known issues with this check MOS DOC ID 1309070.1

Dies zeigt leider wieder, dass die in der Datenbank gesammelten Informationen zu den Features aus DBA Sicht sehr interessant sein können, um beteiligte Objekte kennenzulernen, Wirkkreise besser zu verstehen oder auch Schlagwörter für eine weitere Recherche aufzutun.

```
5 select a.ksppinm "Parameter",
       b.ksppstvl "SessionValue",
       c.ksppstvl "InstanceValue",
       a.KSPDESC "Description"
from x$ksppi a, x$ksppcv b, x$ksppsv c
where a.indx = b.indx and a.indx = c.indx
      and substr(ksppinm,1,1)='_'
      and (upper(ksppinm) like '%DBFUS%')
order by a.ksppinm;
```

⁶ <https://support-lite.oracle.com/kmlite/faces/DocumentDisplay?id=1317265.1>

Als Mittel des Lizenzmanagements ist es leider nicht zulässig. Dafür sei auf die Unterstützung von Oracle oder zertifizierter Partner verwiesen.

Kontaktadresse:

Martin Berger
Lederergasse 27/2/14
A-1080 Wien

Telefon: +43 660 660 83306
E-Mail martin.a.berger@gmail.com
Internet: berxblog.blogspot.com