dba_feature_usage_statistics: Lizenzverletzung im Griff

Paolo Kreth
Schweizerische Mobiliar Versicherungsgesellschaft AG
Bern

Schlüsselworte

Dba_feature_usage_statistics, Lizenzen, Monitoring, Audit, LMS, Licensing

Einleitung
Seit circa drei Jahren habe ich die Leitung des DBA Teams in der Mobi übernehmen dürfen. Gleich am Anfang durfte ich mit einem Hersteller einer Datenbank die wir in der Mobi betreiben einen Lizenz Audit mitmachen. Zum Glück hatten wir nur geringe Datenbanken auf wenigen Server verteilet, aber das Sammeln aller Informationen, und das Know How diesbezüglich war sehr zeitintensiv. Nach dem Audit waren wir dann wirklich im Bild, wussten das A und O des Lizenzmanagement und fühlten uns sicher. Schon nach drei Monate, als ein Neueinkauf bevorstand merkten wir, dass wir die Sicherheit doch nicht mehr hatten, einzelne Details mussten wir wieder hervorrufen, wir konnten nicht ganz so einfach alles nachvollziehen. Daher reagierten wir schnell. Wir trafen die Entscheidung ein Lizenz Cockpit aufzubauen. Ziel war es die Lizenzdaten automatisch, mindestens monatlich zu erheben. Durch die Automatisierung und klare Regeln wollten wir das Know How verewigen, und gleichzeitig merken „wenn etwas aus dem Ruder läuft“. Und, da in der Mobi verschieden DB Systeme benützt werden doch gleich für alle, inklusiv Oracle. Dieser Vortrag erzählt die Geschichte der Lizenzdaten Erhebung auf Oracle, welche Entscheidungen getroffen wurden, und welche Stolpersteine wir aus dem Weg räumen mussten.

Oracle Lizenzierung: welche Option wird wo genützt, welche Lizenzen stehen zur Verfügung

Der erste Schritt zu einer transparenten Lizenzüberprüfung war das Aufnehmen der bestehenden Lizenzen. Diese sind zwar bekannt, denn jeder Vertrag ist gespeichert, und jährlich erhalten wir ja eine schöne Rechnung von Oracle. Leider ist das Vertragssystem nicht wirklich Benützer freundlich, daher entschieden wir uns die Vertragsdaten in unserem Performance Warehouse zu speichern. Jeder Vertrag wurde ein gekippt, und doppelt verifiziert.

Nächster Schritt: wie viele Server sind lizenziert zu welchen Zeitpunkt?

In der Mobi haben wir eine ziemlich einfache Datenbank Landschaft. Wir fahren Oracle auf 8 physische Server, mit einer fixen Anzahl an CPU Cores. Informationen über welche Datenbank sich wo befindet werden über Cloud Control täglich im Mobiliar Performance Warehouse geladen. Daher einfache Sache. Der Mehrwert nachdem wir die Daten geladen und erhoben hatten war, dass wir täglich, auch während einer Server Migration die wir gerade ausführten die genaue Anzahl der benützten Lizenzen im Griff hatten. Parallelbetrieb war keine Variable mehr. Tja Datenbanken bewegen sich halt mal von Zeit zurzeit. Erster Quick Win.

Was mit den Optionen ?

Die Probleme fingen leider bei den benützten Optionen an. Wir haben nicht alle Oracle Optionen in der Mobiliar lizenziert. Oracle benützt kein Key um die Optionen zu aktivieren, in der Software ist schon alles vorhanden. Und die Nutzung wird nach dem Prinzip „Installed and Running“ bewertet.

Ausschalten, z.B. mit einem „Underscore“ Parameter geht nicht immer, zum Beispiel bei der Advanced Compression Option. Genau zum Zeitpunkt als wir die Lizenzen in unserem Warehouse gesammelt hatten und wir erste Auswertungen über die gesamte Landschaft testeten, erstellte ein Lehrling eine Tabelle auf einem Testsystem mit „Compression Advanced“. Zum Glück merkten wir es sofort (wir waren ja am Auswerten), und konnten die Tabelle löschen und neu erstellen. Aber seitdem fingen bei mir die schlaflosen Nächte an.

Das Erstellen von Tabellen auf Entwicklungssysteme ist bei uns in der Obhut des Entwicklers. Nicht all unsere Entwickler sind wirklich mit Lizenz Themen vertraut. Aus meiner Sicht ein grosses Risiko Potential für unser Unternehmen. Dadurch, dass die Performance und Data Dictionary Daten jeder Datenbank täglich in unserem Performance Warehouse geladen werden, war der erste Lösungsansatz sehr einfach. Tägliches „Select distinct "COMPRESS_FOR" from DBA_TABLES“, auf der entsprechende Tabelle in unserem Performance Warehouse. Das Resultat frisch jeden Morgen auf dem DBA Tisch und gegessen.

Wirklich ? Und was mit Datapump (siehe Abbildung 1) ?

```
cat exptest.par
USERID=/
DUMPFILE="exp_for_test.dmp"
LOGFILE="exp_for_test.log"
DIRECTORY=DATA_PUMP_DIR
COMPRESSION=ALL
CONTENT=ALL
SCHEMAS='TEST1'
```

Abb. 1: Parfile Datapump

Auch hier kann man schnell ein Fehler machen. Vor allem weil einige Administration und Entwickler Tools Compression als Default Parameter benützen. ☹

Eine interessante Tech Note

Ich fing an im Oracle Support zu suchen, und merkte schnell es würde zu einer Tech Note Schlacht werden. Für jede Option ein anderes Rezept. Zum Beispiel sagt uns Tech Note 1066837.1 ob Oracle Spatial (Lizenzpflichtig) oder Oracle Locator (in der Standard und der Enterprise Edition enthalten) benützt wird. Tech Note 1926052.1 dagegen erklärt wie man abfragen kann ob irgendwo In Memory aktiviert ist. Als ich dann nach Real Application Testing suchte, und auf Tech Note 1981563.1 stoss, fand ich eine interessante View die ich nicht kannte: DBA_FEATURE_USAGE_STATISTICS. Ich suchte dann Tech Note 1317265.1 und konnte sogar ein Skript finden, welches mir nun Informationen versprach über die Optionen die in einer Datenbank lizenziert sind. Zwar steht in der Tech Note *„Kindly note the report generated is to be used for informational purposes only and this does not represent your license entitlement or requirement”.* Hätte uns diese View helfen können wenigstens ein Bauchgefühl über unseren Lizenzstatus zu gewinnen?

Zur Umsetzung

Wir entschieden, also, die Auswertung zentralisiert in unserem Performance Warehouse über alle Datenbanken auszuführen. Diese Daten werden täglich erhoben und ausgewertet. Bis wir merkten, dass auf einige Datenbanken die Daten gar nicht erhoben wurden. Tja, wir durften uns mit einige ORA-19382 Fehler, die auf einen Timeout verwiesen, konfrontieren. Hier mussten wir ein paar Service Requests öffnen bis wir endlich eine Workaround hatten. Dieser ist auch heute noch auf 12.2 notwendig.

Workaround:

Disable Dynamic Sampling related directives in SQL Plan Directives in system level
alter system set "_optimizer_dsdir_usage_control"=0;

Manually run the Feature Usage Tracking (kick the nmon process)
alter session set events 'immediate trace name mmon_test level 6';

Reenable Dynamic Sampling related directives in SQL Plan Directives in system level
alter system set "_optimizer_dsdir_usage_control"=126;

Ein sehr mächtiges Tool

Das Skript liefert einen gut lesbaren und verständlichen Output. Zuerst werden ein paar Informationen über die Datenbank, Instanz, deren Version und Software Edition angezeigt. Dieses Beispiel beruht auf 12.1.

```
HOST_NAME                      |INSTANCE_NAME  |DATABASE_NAME |OPEN_MODE      |DATABASE_ROLE  |CREATED           |      DBID|VERSION    |BANNER
-------------------------------|---------------|--------------|---------------|---------------|------------------|----------|-----------|-----------------------------------------------------------------------------------------------------------------------------
Myserver1                      |ipaolo1        |IPAOLO1       |READ WRITE     |PRIMARY        |2017.05.04_13.00.02|1131994258|12.1.0.2.0 |Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production
```

Danach folgen Informationen zur Datenbank (Single Instance, CDB, PDBs), und darauf Informationen wann zum letzten Mal die Daten erhoben wurden.

```
LAST_DBA_FUS_DBID|LAST_DBA_FUS_VERS|LAST_DBA_FUS_SAMPLE|SYSDATE            |REMARKS
-----------------|-----------------|-------------------|-------------------|-----------------
       1131994258|12.1.0.2.0       |2017.10.21_03.31.34|2017.10.25_11.10.38|
```

Und schon folgen die ersten Informationen über die angewendeten Optionen. Leider ist der Output in diesem Fall nicht korrekt, da auf meiner Datenbank In Memory aktiviert ist und es leider nicht angezeigt wird.

```
PRODUCT                                            |USAGE                |LAST_SAMPLE_DATE   |FIRST_USAGE_DATE   |LAST_USAGE_DATE
---------------------------------------------------|---------------------|-------------------|-------------------|-------------------
Active Data Guard                                  |NO_USAGE             |2017.10.21_03.31.34|                   |
Advanced Analytics                                 |NO_USAGE             |2017.10.21_03.31.34|                   |
Advanced Compression                               |NO_USAGE             |2017.10.21_03.31.34|                   |
Advanced Security                                  |NO_USAGE             |2017.10.21_03.31.34|                   |
Database In-Memory                                 |NO_USAGE             |2017.10.21_03.31.34|                   |
Database Vault                                     |NO_USAGE             |2017.10.21_03.31.34|                   |
Diagnostics Pack                                   |PAST_USAGE           |2017.10.21_03.31.34|2014.10.10_15.16.32|2017.07.01_02.40.25
Label Security                                     |NO_USAGE             |2017.10.21_03.31.34|                   |
OLAP                                               |NO_USAGE             |2017.10.21_03.31.34|                   |
Partitioning                                       |CURRENT_USAGE        |2017.10.21_03.31.34|2014.02.21_17.18.29|2017.10.21_03.31.34
RAC or RAC One Node                                |NO_USAGE             |2017.10.21_03.31.34|                   |
Real Application Clusters                          |NO_USAGE             |2017.10.21_03.31.34|                   |
Real Application Clusters One Node                 |NO_USAGE             |2017.10.21_03.31.34|                   |
Real Application Testing                           |NO_USAGE             |2017.10.21_03.31.34|                   |
Spatial and Graph                                  |NO_USAGE             |2017.10.21_03.31.34|                   |
Tuning Pack                                        |CURRENT_USAGE        |2017.10.21_03.31.34|2014.02.21_17.18.29|2017.10.21_03.31.34
.Database Gateway                                  |NO_USAGE             |2017.10.21_03.31.34|                   |
```

Abb. 2: Output Script

Wenn man dann weiter nach unten scrollt, dann merkt man schnell, dass der In Memory Teil in der 12.1 wegen einen Bug nicht angezeigt wird. Also mit Vorbehalt geniessen, wie aber auch die Tech Note klar sagt.

```
Database In-Memory                    |In-Memory Aggregation                  |NO_CURRENT_USAGE
Database In-Memory                    |In-Memory Column Store                 |SUPPRESSED_DUE_TO_BUG
Database In-Memory                    |In-Memory Column Store                 |SUPPRESSED_DUE_TO_BUG
```

Abb. 3: Suppressed due to bug

Wie erwähnt In Memory ist wirklich aktiviert, denn gemäss Tech Note (Doc ID 1926052.1) gibt es Tabellen die im Column Store enthalten sind

```
SQL> SELECT OWNER, TABLE_NAME, INMEMORY FROM DBA_TABLES WHERE INMEMORY='ENABLED';

OWNER                          TABLE_NAME                     INMEMORY
------------------------------ ------------------------------ --------------------------------
PAOLO                           IMTEST                        ENABLED
```

Und es ist Memory für den Column Store alloziert.

```
SQL> show parameter inmemory_size

NAME                                 TYPE                              VALUE
------------------------------------ --------------------------------- ------------------------
inmemory_size                        big integer                       250G
```

Interessanter Weise ergibt ein direktes Query auf der View folgenden Output:

```
SQL> select name,currently_used from dba_feature_usage_statistics where name like '%In-
Memory%' ;

NAME                                                             CURRE
---------------------------------------------------------------- -----
In-Memory Aggregation                                            FALSE
In-Memory Aggregation                                            FALSE
In-Memory Column Store                                           TRUE
In-Memory Column Store                                           TRUE
```

Also entschieden wir uns in unserem Cockpit auf eine Hybride Lösung zurückzugreifen. Wo die View gute Resultate liefert wird Sie benütz, dagegen wo Sie „buggy“ ist, werden die Auswertungen gemäss Tech Note ausgeführt. Dies Bedeutet bei In Memory, dass wir sowohl die DBA_SEGMENTS, als auch die V$PARAMETER abfragen.

Besser auf 12.2 ?

Wenigstens in unserem Test Szenario sieht es mit 12.2 besser aus. Wir haben keine „Bug“ Meldungen mehr. Sowohl wenn die Datenbank kein InMemory Store alloziert hat, als auch wenn der InMemory Store alloziert ist aber keine Objekte enthalten sind, wird richtig NO_CURRENT_USAGE angezeigt.

```
PRODUCT                                          |USAGE                |LAST_SAMPLE_DATE   |FIRST_USAGE_DATE   |LAST_USAGE_DATE
-------------------------------------------------|---------------------|-------------------|-------------------|-------------------
Database In-Memory                               |NO_USAGE             |2017.10.26_14.16.19|                   |
```

Sobald man dann eine Tabelle im Column Store setzt, ändert sich der Status zu:

```
PRODUCT                                          |USAGE                |LAST_SAMPLE_DATE   |FIRST_USAGE_DATE   |LAST_USAGE_DATE
-------------------------------------------------|---------------------|-------------------|-------------------|-------------------
Database In-Memory                               |CURRENT_USAGE        |2017.10.26_14.38.57|2017.10.26_14.35.58|2017.10.26_14.38.57
```

Und wenn man dann die Tabelle löscht, ändert sich der Status noch einmal, diesmal zu PAST_USAGE, mit richtigen SAMPLE_DATE und FIRST_USAGE_DATE

```
PRODUCT                                          |USAGE                |LAST_SAMPLE_DATE   |FIRST_USAGE_DATE   |LAST_USAGE_DATE
-------------------------------------------------|---------------------|-------------------|-------------------|-------------------
Database In-Memory                               |PAST_USAGE           |2017.10.26_15.48.24|2017.10.26_14.35.58|2017.10.26_14.38.57
```

Wenn man mehr in den Dettails schaut sieht man auch, dass das Feature zwei Mal benützt wurde und wann zum Letzen Mal.

```
PRODUCT                                                |FEATURE_BEING_USED
-------------------------------------------------------|-----------------------------------------
Database In-Memory                                     |In-Memory Column Store

|USAGE          |LAST_SAMPLE_DATE    |       DBID    |VERSION
----------------|--------------------|---------------|--------
|PAST_USAGE     |2017.10.27_07.58.00 |3698943815     |12.2.0.1.0

|DETECTED_USAGES|TOTAL_SAMPLES|CURRENTLY_USED|FIRST_USAGE_DATE    |LAST_USAGE_DATE
----------------|-------------|--------------|--------------------|-------------------|-------
   2            |9            |FALSE         |2017.10.26_14.35.58|2017.10.26_14.38.57|

|EXTRA_FEATURE_INFO
-----------------------------------------------------------
```

Was kam danach?

Wir bauten ein Cockpit mit Apex, wo jeden Tag der Status aller Lizenzen ersichtlich ist.

Und es wird sogar richtig angezeigt. Z.B. als wir ein POC für den Database Gateway gefahren haben (mit offiziellen Temporäre Lizenzen von Oracle ☺) war der Status auf einmal gelb.

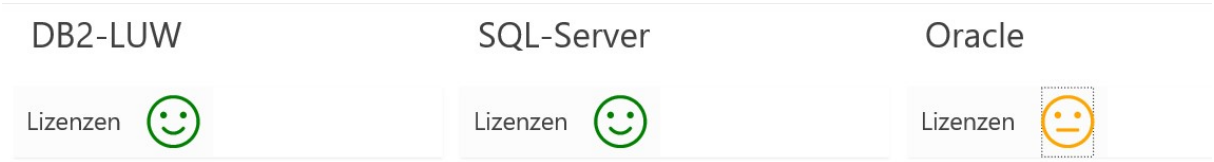

Abb. 4: Einstieg im Lizenz Cockpit

Mit einem Drill Down konnte man dann sehen, dass die „Verletzung“ angezeigt wurde.

DB-System	Produkt	Benutzt	Vorhanden	
Oracle	Oracle Advanced Compression - Processor Perpetual	-		
Oracle	Oracle Real Application Testing - Processor Perpetual	-		
Oracle	Database In-Memory			
Oracle	Multitenant			
Oracle	Oracle Database Lifecycle Management Pack - Processor Perpetual	-		
Oracle	Tuning Pack			
Oracle	Diagnostics Pack			
Oracle	Oracle DB Enterprise Edition			
Oracle	Partitioning			
Oracle	.Database Gateway			☹

Abb. 5: Vorhanden und ausgewiesene Lizenzen (Daten wurden ausgeblendet)

In der Abbildung sieht man die zwei Spalten (deren Zahlen hier ausgeblendet wurden). Die erste zeigt die Anzahl der benützen Lizenzen an. Diese Zahl wird entweder von der DBA_FEATURE_USAGE_STATISTICS entnommen, oder, wo die View noch „buggy“ ist, direkt gemäss Tech Notes aus dem zentralisierten Oracle Catalog. Die zweite Spalte zeigt die Anzahl der vorhandenen Lizenzen, die manuell eingefügt wurden.

Die Zeilen sind dynamisch. Z.B. ist der Database Gateway nicht lizenziert. Dieser Screenshot wurde während ein POC den wir gefahren haben entnommen (wir hatten hier offizielle temnporäre Lizenzen, die wir aber im System nicht eingepflegt hatten). Das System konnte aber sofort erkennen, dass hier Lizenzen benutzt wurden die nicht im Inventar vorhanden waren.

Produkt	Feature	Usage	DB ID	Version	Last usage	Detected usages	Instance	Host
Partitioning	Partitioning (user)	CURRENT_USAGE	3771860575	12.1.0.2.0	21-OCT-17	36		
Partitioning	Partitioning (user)	CURRENT_USAGE	3771860575	12.1.0.2.0	21-OCT-17	36		
Partitioning	Partitioning (user)	CURRENT_USAGE	3261676470	12.1.0.2.0	21-OCT-17	87		

Abb. 6: Usage Details

Fazit

Die DBA_FEATURE_USAGE_STATISTIKS hilft uns sehr ein gutes Bauchgefühl zu haben. Wir fahren aber trotzdem immer noch mit den 4 Augen Prinzip. Neben dem Resultat der View führen wir auch Report aus auf andere Views im Warehouse… ein Bug könnte sich ja jederzeit einschleichen.

Das Beste ist aber, dass alle DBAs in meinem Team eine grössere „Lizenz Awareness“ erlangt haben, und Sie auch viel mehr aufpassen… ein Fehler würde sofort bemerkt werden ☺

Kontaktadresse:

Paolo Kreth

Schweizerische Mobiliar Versicherungsgesellschaft AG

Direktion Bern, Monbijoustrasse 68,

3001 Bern

Telefon: +41 (0) 31 389 75 23

E-Mail paolo.kreth@mobi.ch

Internet: www.mobiliar.ch