ORACLE 11G DATABASE STATISTICS – PRIVATE STATISTICS

Inderpal S. Johal, Data Softech Inc.

OVERVIEW

Prior to Oracle 11g, database statistics are gathered and stored directly in Data Dictionary. Oracle optimizer use this statistics also called Current Statistics as soon as it is stored in Data Dictionary. There are lots of serious flaws in this architecture like

1. Table Statistics can be published in some cases before the statistics of its indexes, partitions or sub-partitions and so make it inconsistent Statistics.
2. New Statistics can severely affect the Application and Database server due to statistics collection settings.

In Oracle 11g, Database statistics are not directly stored in Data dictionary and so you can check it before storing them permanently in Data Dictionary. This type of Statistics is also referred as Private or Deferred Statistics. This way we can take care of first issue mentioned above by publishing tables and their dependent objects at the same time. It will also take care of second issue as now you have the authority to check the new Statistics before submitting them as Current Statistics for Oracle Optimizer. You can validate the new Private Statistics using Oracle DB Replay feature or running Query in SQL prompt and instructing Oracle optimizer to use Private Statistics instead of Current Statistics. When you are done with the testing and satisfied with the new Statistics, then it can be publish as Current Statistics in Data dictionary.

DATABASE STATISTICS STORAGE

Current Statistics	Private Statistics
[DBA | USER | ALL]_TABLES	[DBA | USER | ALL]_TAB_PRIVATE_STATS
[DBA | USER | ALL]_INDEXES	[DBA | USER | ALL]_IND_PRIVATE_STATS
[DBA | USER | ALL]_TAB_COLUMNS	[DBA | USER | ALL]_COL_PRIVATE_STATS

Statistics are collected as Private or deferred Statistics when they are collected using PUBLISH gathering option to FALSE. This option can be set at Database or Schema or Table level using DBMS_STATS package as define earlier in Setting database preferences

SET_[Database | Schema | Table]_PREFS procedure

These statistics are collected in pending statistics tables instead of directly going to Data

[DBA|USER|ALL]_TAB_PENDING_STATS
[DBA|USER|ALL]_COL_PENDING_STATS
[DBA|USER|ALL]_IND_PENDING_STATS
[DBA|USER|ALL]_TAB_HISTGRM_PENDING_STATS

GATHERING PRIVATE STATISTICS

```
SQL> desc salgrade
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 GRADE                                              NUMBER
 LOSAL                                              NUMBER
 HISAL                                              NUMBER
```

```
SQL> select * from salgrade;
     GRADE      LOSAL      HISAL
---------- ---------- ----------
         1        700       1200
         2       1201       1400
         3       1401       2000
         4       2001       3000
         5       3001       9999
```

```
SQL> alter session set nls_date_format='dd/mm/yyyy hh24:mi:ss';
Session altered.
```

Delete the Statistics for SCOTT.SALGRADE table

```
SQL> exec dbms_stats.delete_table_stats('SCOTT','SALGRADE');
PL/SQL procedure successfully completed.
```

Verify that Statistics is deleted for Table/Index from the Data Dictionary

```
SQL> select table_name, last_analyzed from user_tables where table_name = 'SALGRADE';
TABLE_NAME                     LAST_ANALYZED
------------------------------ -------------------
SALGRADE

SQL> select index_name, last_analyzed from user_indexes where table_name =
'SALGRADE';
no rows selected

SQL> select column_name, last_analyzed from user_tab_columns where
table_name='SALGRADE';
COLUMN_NAME                    LAST_ANALYZED
------------------------------ -------------------
GRADE
LOSAL
HISAL
```

Verify if there is any Private or Deferred Statistics currently available for SCOTT.SALGRADE

```
SQL> select table_name, last_analyzed from user_tab_pending_stats where table_name =
'SALGRADE';
no rows selected

SQL> select table_name, last_analyzed from user_ind_pending_stats where table_name =
'SALGRADE';
no rows selected

SQL> select table_name, last_analyzed from user_col_pending_stats where table_name =
'SALGRADE';
no rows selected
```

So Currently Table SALGRADE has no Current as well as Private Statistics

Check the Global PUBLISH Preference setting.

```
SQL> select dbms_stats.get_prefs('PUBLISH') publish from dual;
PUBLISH
--------------------------------------------------------------------------------
TRUE
```

Check the Table Level PUBLISH Preference setting for SCOTT.SALGRADE.

```
SQL> select dbms_stats.get_prefs('PUBLISH', 'SCOTT', 'SALGRADE') publish from dual;
PUBLISH
--------------------------------------------------------------------------------
TRUE
```

Set the CUSTOMERS_OBE tables publish value to false. This means that any statistics gathered from now on will not be automatically published. From your terminal window, execute the following SQL script:

```
SQL> exec dbms_stats.set_table_prefs('SCOTT', 'SALGRADE', 'PUBLISH', 'false');
PL/SQL procedure successfully completed.

SQL> select dbms_stats.get_prefs('PUBLISH', 'SCOTT', 'SALGRADE') publish from dual;
PUBLISH
--------------------------------------------------------------------------------
FALSE
```

Now you can gather statistics on the SALGRADE table

```
SQL> execute dbms_stats.gather_table_stats('SCOTT', 'SALGRADE');
PL/SQL procedure successfully completed.
```

Verify if the Stats is collected in Private Area or in Data Dictionary

```
SQL> select table_name, last_analyzed from user_tables where table_name = 'SALGRADE';
TABLE_NAME                     LAST_ANAL
------------------------------ ---------
SALGRADE

SQL> select index_name, last_analyzed from user_indexes where table_name =
'SALGRADE';
no rows selected
```

```
SQL> select column_name, last_analyzed from user_tab_columns where
table  name='SALGRADE';
COLUMN_NAME                    LAST_ANAL
------------------------------ ---------
GRADE
LOSAL
HISAL
```

We find that Stats is only collected in Pending are as shown below

```
SQL> select table_name, last_analyzed from user_tab_pending_stats where table_name =
'SALGRADE';
TABLE_NAME                     LAST_ANAL
------------------------------ ---------
SALGRADE                       06-AUG-07

SQL> select table_name, last_analyzed from user_ind_pending_stats where table_name =
'SALGRADE';
no rows selected

SQL> select table_name, last_analyzed from user_col_pending_stats where table_name =
'SALGRADE';
TABLE_NAME                     LAST_ANAL
------------------------------ ---------
SALGRADE                       06-AUG-07
SALGRADE                       06-AUG-07
SALGRADE                       06-AUG-07
```

TESTING PRIVATE STATISTICS

There are two methods for testing the Private or Pending Statistics

1. Test on QA server
2. Test on Production Server itself

TEST ON QA SERVER

In order to test the Private Statistics collected on QA server, you need to take these statistics to QA server . This can be achieved by using the following Steps

1. **Create the STATS table on Production server to copy the Private Statistics to Database Table**
   SQL> exec dbms_stats.create_stat_table('SCOTT','STATS');
   PL/SQL procedure successfully completed.

2. **Take the Export Dump of Private Statistics in newly created table STATS using DBMS_STATS.export_pending_stats procedure.**
   SQL> exec dbms_stats.export_pending_stats(tabname=>'SALGRADE',stattab=>'STATS');
   PL/SQL procedure successfully completed.

   We found that 4 records are exported to STATS table
   SQL> select count(*) from stats;
   ```
    COUNT(*)
   ----------
           4
   ```

3. Export the STATS table using **expdp** utility
4. Import the STATS table on QA server using **impdp** utility
5. **Import the Statistics to Database Data Dictionary on QA server using DBMS_STATS.import_table_stats procedure**
   SQL> exec
   dbms_stats.import_table_stats(ownname=>'SCOTT',tabname=>'SALGRADE',stattab=>'STATS');
   PL/SQL procedure successfully completed.

   Verify that Stats is transferred in the Data Dictionary of Test Instance

```
SQL> select table_name, last_analyzed from user_tables where table_name =
'SALGRADE';
TABLE_NAME                     LAST_ANAL
------------------------------ ---------
SALGRADE                       06-AUG-07

SQL> select index_name, last_analyzed from user_indexes where table_name =
'SALGRADE';
no rows selected

SQL> select column_name, last_analyzed from user_tab_columns where
table_name='SALGRADE';
COLUMN_NAME                    LAST_ANAL
------------------------------ ---------
GRADE                          06-AUG-07
LOSAL                          06-AUG-07
HISAL                          06-AUG-07
```

6. Copy the Production Data on the QA server using **Expdp** or **RMAN** or other procedure.
7. Copy the Workload on the Production server and replay on QA server using **DBReplay** or execute your scripts or Application to test the new Statistics

TEST ON PRODUCTION SERVER

This can be done using the following approach

1. Session level setting to enable Optimizer to use Private Statistics
2. Set up the initialization parameter to enable Optimizer to use Private Statistics instead of Current Statistics. It can be done at System or Session level based in your environment and application. By default this parameter is FALSE.

```
SQL>  select name,value,isses_modifiable,issys_modifiable
        from v$parameter
       where name='optimizer_use_pending_statistics';

NAME                                VALUE      ISSES ISSYS_MOD
----------------------------------- ---------- ----- ---------
optimizer_use_pending_statistics    FALSE      TRUE  IMMEDIATE

SQL>ALTER SESSION SET optimizer_use_pending_statistics  =TRUE;
```

PUBLISHING PRIVATE OR PENDING STATISTICS

Once testing is done and you are satisfied with the new statistics, you can now publish or convert the Private Statistics to Public or Current Statistics

Check if there is any current Statistcis available in the Data Dictionary

```
SQL> select table_name, last_analyzed from user_tables where table_name = 'SALGRADE';
TABLE_NAME                     LAST_ANAL
------------------------------ ---------
SALGRADE

SQL> select index_name, last_analyzed from user_indexes where table_name =
'SALGRADE';
no rows selected

SQL> select column_name, last_analyzed from user_tab_columns where
table_name='SALGRADE';
COLUMN_NAME                    LAST_ANAL
------------------------------ ---------
GRADE
LOSAL
HISAL
```

Check the Statistics is available in the Pending area to be moved to Data Dictionary for Optimizer usage

```
SQL> select table_name, last_analyzed from user_tab_pending_stats where table_name =
'SALGRADE';
TABLE_NAME                     LAST_ANAL
------------------------------ ---------
SALGRADE                       06-AUG-07

SQL> select table_name, last_analyzed from user_ind_pending_stats where table_name =
'SALGRADE';
no rows selected

SQL> select table_name, last_analyzed from user_col_pending_stats where table_name =
'SALGRADE';
TABLE_NAME                     LAST_ANAL
------------------------------ ---------
SALGRADE                       06-AUG-07
SALGRADE                       06-AUG-07
SALGRADE                       06-AUG-07
```

Publish the Stats to Data Dictionary for Optimizer Usage

```
SQL>  exec dbms_stats.publish_pending_stats(tabname=>'SALGRADE');
PL/SQL procedure successfully completed.
```

Verify if the Stats is moved to Data Dictionary

```
SQL> select table_name, last_analyzed from user_tables where table_name = 'SALGRADE';
TABLE_NAME                     LAST_ANAL
------------------------------ ---------
SALGRADE                       06-AUG-07

SQL> select index_name, last_analyzed from user_indexes where table_name =
'SALGRADE';
no rows selected

SQL> select column_name, last_analyzed from user_tab_columns where
table_name='SALGRADE';
COLUMN_NAME                    LAST_ANAL
------------------------------ ---------
GRADE                          06-AUG-07
LOSAL                          06-AUG-07
HISAL                          06-AUG-07
```

Check what happen to Stat earlier available in Pending Area

```
SQL> select table_name, last_analyzed from user_tab_pending_stats where table_name =
'SALGRADE';
no rows selected

SQL> select table_name, last_analyzed from user_ind_pending_stats where table_name =
'SALGRADE';
no rows selected

SQL> select table_name, last_analyzed from user_col_pending_stats where table_name =
'SALGRADE';
no rows selected
```

Once the Statistics are published you have to make sure that Table or Schema level preference for PUBLISH option are changed back to TRUE so that new Statistics collected by the Auto Maintenance Job is published immediately.

```
SQL> select name,value from v$parameter where
name='optimizer_use_pending_statistics';
NAME                                 VALUE
------------------------------------ ----------
optimizer_use_pending_statistics     TRUE

SQL> select name,value from v$parameter where
name='optimizer_use_pending_statistics';
NAME                                 VALUE
------------------------------------ ----------
optimizer_use_pending_statistics     FALSE
```

```
SQL> Select dbms_stats.get_prefs('PUBLISH','SCOTT','SALGRADE') from dual;
DBMS_STATS.GET_PREFS('PUBLISH','SCOTT','SALGRADE')
--------------------------------------------------------------------------------
FALSE

SQL> Exec dbms_stats.delete_table_prefs('SCOTT','SALGRADE','PUBLISH');
PL/SQL procedure successfully completed.

SQL> Select dbms_stats.get_prefs('PUBLISH','SCOTT','SALGRADE') from dual;
DBMS_STATS.GET_PREFS('PUBLISH','SCOTT','SALGRADE')
--------------------------------------------------------------------------------
TRUE
```