

Oracle Rdb *Journal*



Collecting and Deploying Oracle Rdb Workload Statistics

A Feature Article from the Rdb Journal

By Don Green

September 15, 2001

Copyright © 2001 Oracle Corporation. All Rights Reserved.

Collecting and Deploying Oracle Rdb Workload Statistics

Workload statistics improve the Rdb optimizer estimates when selecting optimal query strategies. Workload statistics can have a great performance benefit when executing complex queries.



Don Green

Technical Analyst, Oracle Rdb Support

September 15, 2001

Oracle Rdb7 introduced the functionality for collecting workload statistics. Workload statistics improve the Rdb optimizer estimates when selecting optimal query strategies. Workload statistics can have a great performance benefit when executing complex queries. Workload collection allows the optimizer to identify ICG's (interesting column groups) for specific queries and write them to the database in a special RDB\$WORKLOAD table.

Unlike the cardinality statistics, workload statistics are not automatically updated during insert/update/delete or truncate table operations. Once the workload is collected then RMU/COLLECT is used periodically to keep the statistics up-to-date.

Workload statistics were originally part of Oracle Rdb7. Now Rdb release 7.0.6 includes the ability to deploy statistics from one database to another. This functionality is ideal for developers designing new applications. These developers can test new applications using real production collected statistics, and/or design new queries in development and use the profile to better hone them before implementation.

In Oracle Rdb7 release 7.0.6, RMU/EXTRACT allows the workload and statistics stored in the RDB\$WORKLOAD table to be unloaded as a DCL script of RMU/INSERT OPTIMIZER_STATISTICS

statements. This unloaded information can be applied after a new database is created using the SQL EXPORT and IMPORT statements, or it can be applied to a similar database for use by the RMU/COLLECT OPTIMIZER_STATISTICS/STATISTIC=WORKLOAD command.

Note: Oracle Rdb 7.0.6.2 corrects a problem in RMU/EXTRACT that causes it to fail when more than one column is found in an ICG.

1. Create the sample database MF_PERSONNEL.

```
$ @rdm$demo:personnel sql m nocdd
```

2. Enable workload collecting on the database using the SQL ALTER DATABASE command:

```
SQL> ALTER DATABASE  
cont> FILENAME MF_PERSONNEL  
cont> WORKLOAD COLLECTION IS ENABLED;
```

Oracle Rdb creates the RDB\$WORKLOAD table as a result of this SQL ALTER DATABASE statement. This step is also called enabling the workload profile.

The Rdb Technical Corner is a regular feature of the Oracle Rdb Web Journal. (www.oracle.com/rdb/rdb_journal)

3. Execute the SQL query to gather statistics on. For this example the query is a simple SELECT from a view which does a join of five underlying tables.

```
SQL> select * from current_info where last_name='Toliver';
LAST_NAME          FIRST_NAME    ID          DEPARTMENT
JOB                JSTART       SSTART      SALARY
Toliver            Alvin        00164      Board Manufacturing North
Department Manager 21-Sep-1981  14-Jan-1983 $51,712.00
1 row selected
SQL> commit;
```

This one query caused 14 rows of data to be inserted into RDB\$WORKLOAD.

```
SQL> select count(*) from rdb$workload;
      14
1 row selected
```

4. Disable workload profiling, otherwise all future queries will be profiled.

```
SQL> ALTER DATABASE
cont> FILENAME MF_PERSONNEL
cont> WORKLOAD COLLECTION IS DISABLED;
```

Typically workload collection adds very little overhead during production use, however, as new ICG's are detected they are written to disk at commit time. This extra I/O is probably not beneficial over the longer time frame.

- Execute the `RMU/COLLECT OPTIMIZER_STATISTICS` command which forces RMU to update the system relation `RDB$WORKLOAD` with statistical data.

```
$ RMU/COLLECT OPTIMIZER_STATISTICS MF_PERSONNEL/STATISTICS=(WORKLOAD)
```

At this point you can use `RMU/SHOW OPTIMIZER` and display the data collected for the five underlying tables. Note the "Last collected time" column. This timestamp depicts when the statistics were last updated.

This value comes from the `RDB$LAST ALTERED` column in `RDB$WORKLOAD`. If it is `NULL` then they have never been collected. For this example some of the output has been truncated.

```
$ RMU/SHOW OPTIMIZER MF_PERSONNEL -
_$ /TABLE=(JOBS, DEPARTMENTS, JOB_HISTORY, EMPLOYEES, SALARY_HISTORY)
```

Optimizer Statistics for table : JOBS

```
Cardinality          : 15
Row clustering factor : 0.0000000

Workload Column group : JOB_CODE
Duplicity factor      : 1.0000000
Null factor           : 0.0000000
First created time    : 19-MAR-2001 13:34:28.61
Last collected time   : 20-MAR-2001 15:58:43.37
```

Optimizer Statistics for table : DEPARTMENTS

```
Cardinality          : 26
Row clustering factor : 0.0000000

Workload Column group : DEPARTMENT_CODE
Duplicity factor      : 1.0000000
Null factor           : 0.0000000
First created time    : 19-MAR-2001 13:34:28.61
Last collected time   : 20-MAR-2001 15:58:43.36
```

Index name : DEPARTMENTS_INDEX

```
Index Cardinality    : 0
Average Depth        : 0.0000000
Key clustering factor : 0.0000000
Data clustering factor : 0.0000000
Segment Column       Prefix cardinality
DEPARTMENT_CODE     0
```

6. Execute the query previously profiled again. The optimizer uses the workload statistics gathered by Oracle RMU/COLLECT. To see if the optimizer has

referenced the workload statistics define the following logical:

```
$ define rdms$debug_flags "0"
```

Then run the query and look for the line:

```
~0: Workload statistics used
```

7. Create the SQL script which will be used to input the statistics into a second database.

is not included in the ALL item.

In Rdb 7.0.6 a new WORKLOAD keyword has been added to the /ITEM qualifier. The default is /ITEM=NOWORKLOAD. The WORKLOAD item

This item generates a DCL command language script for OpenVMS as shown in the example below. The script includes references to Rdb system tables as well as user tables.

```
$ RMU/EXTRACT/ITEM=WORKLOAD/OUT=WL_EXT.COM -  
MF_PERSONNEL /LOG/OPTION=(FILENAME,AUDIT)  
  
%RMU-I-LOGIDENT, RMU EXTRACT for Oracle Rdb V7.0-61, Log date  
19-MAR-2001 14:34:08.20  
%RMU-I-LOGITEM, selected item - WORKLOAD  
%RMU-I-LOGOPTION, selected option - FILENAME_ONLY  
%RMU-I-LOGOPTION, selected option - AUDIT_COMMENT  
%RMU-I-LOGOUTPUT, output file is DISK:[DIR.WORKLOAD]WL_EXT.COM;1  
%RMU-I-LOGDBNAME, database root file is  
DISK:[DIR.WORKLOAD]MF_PERSONNEL.RDB;1  
%RMU-I-LOGVERSION, this database is at version V7.0  
%RMU-I-ENDEXTRACT, elapsed time for metadata extract : 0 00:00:02.89
```

The following options can be used to modify the output of the WORKLOAD item.

```
- /OPTION=AUDIT_COMMENT
```

Each RMU/INSERT OPTIMIZER_STATISTICS statement is preceded by the created and altered date for the workload entry. The default is /OPTION=NOAUDIT_COMMENT.

```
- /OPTION=FILENAME_ONLY
```

The database file specification output for the RMU/INSERT OPTIMIZER_STATISTICS statement is abbreviated to just the filename portion.

8. From the example in step 7 the file WL_EXT.COM is created with the necessary SQL statements to deploy the workload statistics into a similar database. The view CURRENT_INFO does a five table join

and you can see the ICG's for the underlying user tables in WL_EXT.COM as well as the Rdb system relations used. For this example the output has been truncated.

```
$! RMU/EXTRACT for Oracle Rdb V7.0-61      19-MAR-2001 14:34:08.20
$!
$!                                     WORKLOAD Procedure
$!
$!-----
$ SET VERIFY
$ SET NOON
$
$! Created on 19-MAR-2001 13:34:28.61
$! Last collected on 19-MAR-2001 14:01:53.25
$!
$ RMU/INSERT OPTIMIZER_STATISTICS -
MF_PERSONNEL -
/TABLE=(DEPARTMENTS) -
/COLUMN_GROUP=(DEPARTMENT_CODE) -
/DUPLICITY_FACTOR=(1.0000000) -
/NULL_FACTOR=(0.0000000) /LOG
:
:
$! Created on 19-MAR-2001 13:34:28.61
$! Last collected on 19-MAR-2001 14:01:53.36
$!
$ RMU/INSERT OPTIMIZER_STATISTICS -
MF_PERSONNEL -
/TABLE=(JOB_HISTORY) -
/COLUMN_GROUP=(JOB_CODE) -
/DUPLICITY_FACTOR=(18.2666664) -
/NULL_FACTOR=(0.0000000) /LOG
$
$! Created on 19-MAR-2001 13:34:28.61
$! Last collected on 19-MAR-2001 14:01:53.36
$!
$ RMU/INSERT OPTIMIZER_STATISTICS -
MF_PERSONNEL -
/TABLE=(JOB_HISTORY) -
/COLUMN_GROUP=(JOB_END) -
/DUPLICITY_FACTOR=(1.7018634) -
/NULL_FACTOR=(0.3649635) /LOG
```

Note the different NULL_FACTOR values in the above example. JOB_CODE is never NULL but JOB_END is NULL 36% of the time.

```
$
$! Created on 19-MAR-2001 13:34:28.61
$! Never collected
$!
$ RMU/INSERT OPTIMIZER_STATISTICS -
  MF_PERSONNEL -
  /TABLE=(RDB$FIELDS) -
  /COLUMN_GROUP=(RDB$FIELD_NAME) -
  /NULL_FACTOR=(0.0000000) /LOG
$
```

Note that the system queries performed by SQL are also cataloged. If these are not required for the workload then they can be deleted using the RMU/DELETE OPTIMIZER command.

9. Execute the WL_EXT.COM file in the VMS directory where the recipient database root file is located. Note, that it is critical that the recipient database has already been created from either a SQL EXPORT and IMPORT from the original database,

or that it has been created previously from a RMU/BACKUP and RMU/RESTORE of the originating database. In this example the secondary database was already created using RMU/RESTORE.

```
$ SET DEFAULT DKA300:[SQLUSER70.WORKLOAD.TEST]

$ SQL$ ALTER DATABASE FILENAME MF_PERSONNEL -
  WORKLOAD COLLECTION IS ENABLED;

$ @WL_EXT.COM

$ SQL$ ALTER DATABASE FILENAME MF_PERSONNEL -
  WORKLOAD COLLECTION IS DISABLED;
```

DBA's who are deploying statistics on a regular basis may want to consider extracting the data from the workload profile before adding a new query (on the original database) and then doing this again after executing the query for the purpose of editing the newer file so that it only contains the information just

added. Otherwise, executing the RMU script will result in informational messages that the rows already exist in the secondary database. You can always use the following query to see what rows in the RDB\$WORKLOAD table are for what table:

```
SELECT * FROM RDB$WORKLOAD
WHERE RDB$RELATION_ID = (SELECT RDB$RELATION_ID
                        FROM RDB$RELATIONS
                        WHERE RDB$RELATION_NAME = 'yourtablename')
ORDER BY RDB$CREATED;
```

10. RDB\$WORKLOAD specific rows can be deleted from the table by using the RMU/DELETE OPTIMIZER command and the /TABLE parameter.

Consult the Oracle Rdb RMU Reference Manual for more information regarding the RMU/COLLECT and RMU/DELETE commands. ↻

Related documentation:

- [Oracle Rdb SQL Reference Manual](#)
- [Oracle Rdb RMU Reference Manual](#)
- [Oracle Rdb V7.0.6 Release Notes](#)

Don Green has been supporting Rdb and its peripheral products since 1992. He lives in Massachusetts and has been a past contributor to the Rdb Web Journal.