**Practices for Lesson 15**

**Global Resource Management Concepts**

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 1

**Practices for Lesson 15: Overview**

**Practices Overview**

In this practice, you will explore the dynamics of Cache Fusion, Sequences, Library Cache, Global Enqueues, and Results

Cache in an Oracle RAC environment.

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 2

# Practice 15-1: Buffer Cache Fusion

### Overview

In this lesson, you will examine the workings of cache fusion in an Oracle RAC database.

1. Open a terminal session on your desktop to host01 as the oracle user.

|  |
| --- |
| [vncuser@*classsroom\_pc* ~]$ **ssh oracle@host01**  oracle@host01's password:  Last login: Wed Dec 10 17:47:14 2014 from 192.0.2.1  [oracle@host01 ~]$ |

1. Change the directory to /stage/RAC/labs/less\_07. Set the environment and copy

/stage/RAC/labs/less\_07/tnsnames.ora to file to

$ORACLE\_HOME/network/admin on each node. Check which instance is running on

host01.

|  |
| --- |
| [oracle@host01 ~]$ **cd /stage/RAC/labs/less\_07**  [oracle@host01 less\_07]$ **. oraenv**  ORACLE\_SID = [oracle] ? **orcl**  The Oracle base has been set to /u01/app/oracle  [oracle@host01 less\_07]$ **cp tnsnames.ora**  **/u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora**  [oracle@host01 less\_07]$ **ssh host02 cp**  **/stage/RAC/labs/less\_07/tnsnames.ora**  **/u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora**  [oracle@host01 less\_07]$ **ssh host03 cp**  **/stage/RAC/labs/less\_07/tnsnames.ora**  **/u01/app/oracle/product/12.1.0/dbhome\_1/network/admin/tnsnames.ora**  [oracle@host01 less\_07]$ **ps -ef|grep ora\_smon**  oracle 12256 1 0 Dec09 ? 00:00:19 ora\_smon\_orcl\_3  oracle 29985 28381 0 17:50 pts/1 00:00:00 grep ora\_smon [oracle@host01 less\_07]$ **export ORACLE\_SID=orcl\_3**  [oracle@host01 less\_07]$ |

1. Start SQL\*Plus and verify the local instance name.

|  |
| --- |
| [oracle@host01 less\_07]$ **sqlplus / as sysdba** |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 3

|  |
| --- |
| SQL\*Plus:Release 12.1.0.2 Production on Wed Dec 10 19:22:12 2014 Copyright (c) 1982, 2014, Oracle. All rights reserved.  Connected to:  Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production  With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,  Advanced Analytics and Real Application Testing options  SYS@orcl\_3 SQL> **select instance\_name from v$instance;**  Hit Any Key to Continue  INSTANCE\_NAME  orcl\_3  1 row selected.  SYS@orcl\_3 SQL> |

1. Configure the scott database user as shown below.

|  |
| --- |
| SYS@orcl\_3 SQL> **alter user scott identified by tiger account unlock;**  User altered.  SYS@orcl\_3 SQL> |

1. Find the block ID and file number for the block containing the record for employee number 7521 from the SCOTT.EMP table.

|  |
| --- |
| SYS@orcl\_3 SQL> **select empno, ename, dbms\_rowid.rowid\_relative\_fno(rowid) fileno,**  **dbms\_rowid.rowid\_block\_number(rowid) block\_no from scott.emp where empno = 7521;**  Hit Any Key to Continue  EMPNO ENAME FILENO BLOCK\_NO  7521 WARD 6 197 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 4

|  |
| --- |
| 1 row selected.  SYS@orcl\_3 SQL> |

1. Using the file and block number identified, check the status of the block in GV$BH as shown below.

|  |
| --- |
| SYS@orcl\_3 SQL> **select inst\_id,file#,block#,status,dirty from gv$bh where file#=6 and block# = 197 and status !='free'**  **order by 1,2,3;**  no rows selected SYS@orcl\_3 SQL> |

1. Read the record for WARD from the SCOTT.EMP table and recheck the block status in

GV$BH. What do you see?

|  |
| --- |
| SYS@orcl\_3 SQL> **select \* from scott.emp where ename = 'WARD';**  Hit Any Key to Continue  EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO      7521 WARD SALESMAN 7698 22-feb-1981 00:00:00  1250 500 30  1 row selected.  SYS@orcl\_3 SQL> **select inst\_id,file#,block#,status,dirty from gv$bh where file#=6 and block# = 197 and status !='free'**  **order by 1,2,3;**  Hit Any Key to Continue  INST\_ID FILE# BLOCK# STATUS D  -  3 6 197 scur N  1 row selected.  SYS@orcl\_3 SQL> |

GV$BH shows that the local instance (orcl\_3) has cached the block image in SCUR (shared current) mode.

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 5

1. Open a terminal session on your desktop to host02 as the oracle user. Change directory to /stage/RAC/labs/less\_07, check which instance is running on host02, and set the environment.

|  |
| --- |
| [vncuser@*classsroom\_pc* ~]$ **ssh oracle@host02**  oracle@host02's password:  Last login: Wed Dec 10 17:47:14 2014 from 192.0.2.1 [oracle@host02 ~]$ **cd /stage/RAC/labs/less\_07**  [oracle@host02 less\_07]$ **ps -ef|grep ora\_smon**  oracle 12256 1 0 Dec09 ? 00:00:19 ora\_smon\_orcl\_1  oracle 29985 28381 0 17:50 pts/1 00:00:00 grep ora\_smon  [oracle@host02 less\_07]$ **. oraenv**  ORACLE\_SID = [oracle] ? **orcl**  The Oracle base has been set to /u01/app/oracle  [oracle@host02 less\_07]$ **export ORACLE\_SID=orcl\_1**  [oracle@host02 less\_07]$ |

1. On host02, start SQL\*Plus, read the record for WARD from the SCOTT.EMP table and recheck the block status in GV$BH. What do you see?

|  |
| --- |
| [oracle@host02 less\_07]$ **sqlplus / as sysdba**  SQL\*Plus: Release 12.1.0.2.0 Production on Mon Jan 5 14:01:26  2015  Copyright (c) 1982, 2014, Oracle. All rights reserved.  Connected to:  Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production  With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,  Advanced Analytics and Real Application Testing options SYS@orcl\_1 SQL> **select \* from scott.emp where ename = 'WARD';**  EMPNO ENAME JOB MGR HIREDATE SAL  COMM |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 6

|  |
| --- |
| DEPTNO  7521 WARD SALESMAN 7698 22-FEB-81 1250  500  30  SYS@orcl\_1 SQL> **select inst\_id,file#,block#,status,dirty from gv$bh where file#=6 and block# = 197 and status !='free'**  **order by 1,2,3;**  INST\_ID FILE# BLOCK# STATUS D  -  1 6 197 scur N  3 6 197 scur N  SYS@orcl\_1 SQL> |

You should see that both instance 1 (orcl\_1) and instance 3 (orcl\_3) have cached the block image in SCUR (shared current) mode.

1. Open a terminal session on your desktop to host03 as the oracle user. Change the directory to /stage/RAC/labs/less\_07, check which instance is running on host03, and set the environment.

|  |
| --- |
| [vncuser@*classsroom\_pc* ~]$ **ssh oracle@host03**  oracle@host03's password:  Last login: Wed Dec 10 17:57:14 2014 from 192.0.2.1 [oracle@host03 ~]$ **cd /stage/RAC/labs/less\_07**  [oracle@host03 less\_07]$ **ps -ef|grep ora\_smon**  oracle 6889 1 0 2014 ? 00:01:04 ora\_smon\_orcl\_2  oracle 21252 19534 0 14:25 pts/0 00:00:00 grep ora\_smon  [oracle@host03 less\_07]$ **. oraenv**  ORACLE\_SID = [oracle] ? **orcl**  The Oracle base has been set to /u01/app/oracle  [oracle@host03 less\_07]$ **export ORACLE\_SID=orcl\_2**  [oracle@host03 less\_07]$ |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 7

1. From host03, start SQL\*Plus, read the record for WARD from the SCOTT.EMP table, and recheck the block status in GV$BH. What do you see?

|  |
| --- |
| [oracle@host03 less\_07]$ **sqlplus / as sysdba**  SQL\*Plus: Release 12.1.0.2.0 Production on Mon Jan 5 14:27:08  2015  Copyright (c) 1982, 2014, Oracle. All rights reserved.  Connected to:  Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production  With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,  Advanced Analytics and Real Application Testing options  SYS@orcl\_2 SQL> **select \* from scott.emp where ename = 'WARD';**  Hit Any Key to Continue  EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO      7521 WARD SALESMAN 7698 22-feb-1981 00:00:00  1250 500 30  1 row selected.  SYS@orcl\_2 SQL> **select inst\_id,file#,block#,status,dirty from gv$bh where file#=6 and block# = 197 and status !='free'**  **order by 1,2,3;**  Hit Any Key to Continue  INST\_ID FILE# BLOCK# STATUS D  -   1. 6 197 scur N 2. 6 197 scur N 3. 6 197 scur N   3 rows selected. |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 8

|  |
| --- |
| SYS@orcl\_2 SQL> |

You should see that all three instances have cached the block image in SCUR (shared current) mode.

1. Next, let’s engineer a read/write for the block from the instance on host01. Use the

cf\_upward.sql script to do this. Recheck the block status in GV$BH. What do you see?

|  |
| --- |
| SYS@orcl\_3 SQL> **!cat cf\_upward.sql**  update scott.emp set sal = sal + 0 where ename = 'WARD'; SYS@orcl\_3 SQL> **@cf\_upward.sql**  1 row updated.  SYS@orcl\_3 SQL> **commit;**  Commit complete.  SYS@orcl\_3 SQL> **select inst\_id,file#,block#,status,dirty from gv$bh where file#=6 and block# = 197 and status !='free'**  **order by 1,2,3;**  Hit Any Key to Continue  INST\_ID FILE# BLOCK# STATUS D  -  1 6 197 cr N  2 6 197 cr N  3 6 197 xcur Y  6 197 cr N  4 rows selected.  SYS@orcl\_3 SQL> |

The updating instance now holds an XCUR (exclusive current) image for the block and all SCURs have been downgraded to CR. Note that the XCUR shows dirty. When you have a read/write condition, existing SCURs are downgraded to CR while the updating instance will indicate an XCUR. No other instance can have either a shared or exclusive image of the same block.

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 9

1. Go to host02 and engineer a write/write for the block from the local instance. Use the

cf\_upward.sql script to do this. Recheck the block status in GV$BH. What do you see?

|  |
| --- |
| SYS@orcl\_1 SQL**>@cf\_upward.sql**  1 row updated.  SYS@orcl\_1 SQL> **commit;**  Commit complete.  SYS@orcl\_1 SQL> **select inst\_id,file#,block#,status,dirty from gv$bh where file#=6 and block# = 197 and status !='free'**  **order by 1,2,3;**  Hit Any Key to Continue  INST\_ID FILE# BLOCK# STATUS D  -  1 6 197 xcur Y  6 197 cr N  6 197 cr N  2 6 197 cr N  3 6 197 pi Y  6 197 cr N  6 rows selected.  SYS@orcl\_1 SQL> |

The updating instance (orcl\_1) now holds an XCUR (exclusive current) image for the block. The old XCUR held by instance 3 has been downgraded to a PI (past image). Note that the XCUR and PI both show as dirty. The past image is nothing more than a copy of a dirty block that is used by the Global Cache Service. Past images of blocks are maintained by GCS until writes covering those versions are recorded. Past images are used in failure recovery. Once a later image is written from any instance, the PI image is converted to CR.

1. From host01 (instance 3) issue a checkpoint. Recheck the block status in GV$BH. What do you see?

|  |
| --- |
| SYS@orcl\_3 SQL> **alter system checkpoint;**  System altered. |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 10

|  |
| --- |
| SYS@orcl\_3 SQL> **select inst\_id,file#,block#,status,dirty from gv$bh where file#=6 and block# = 197 and status !='free'**  **order by 1,2,3;** 2 3  Hit Any Key to Continue  INST\_ID FILE# BLOCK# STATUS D  -  1 6 197 xcur N  6 197 cr N  6 197 cr N  2 6 197 cr N  3 6 197 cr N  6 197 cr N  6 rows selected.  SYS@orcl\_3 SQL> |

The PI image has been converted to a CR image. No images are dirty after the checkpoint.

1. Close all terminals opened for this practice.

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 11

# Practice 15-2: Row Cache Coordination for Sequences

### Overview

In this practice you will examine the behavior of cached and non-cached sequences in a RAC database and the resource coordination provided by RAC for a row cache resource.

1. Establish an ssh connection option as the oracle user. Use oraenv to set the Oracle environment. Change the directory to /stage/RAC/labs/less\_07. Connect to instance 1 (orcl\_1).

[vncuser@*classsroom\_pc* ~]$ **ssh oracle@host01**

oracle@host01's password:

[oracle@host01 ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orcl**

The Oracle base has been set to /u01/app/oracle

[oracle@host01 ~]$ **cd /stage/RAC/labs/less\_07**

[oracle@host01 less\_07]$ **sqlplus sys/*sys\_password*@orcl\_1 as sysdba**

SQL\*Plus: Release 12.1.0.2.0 Production on Tue Jan 27 07:16:33 2015

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,

Advanced Analytics and Real Application Testing options

1. Run the seq\_lock\_typ.sql script to view sequence lock types.

SYS@orcl\_1 SQL> **!cat seq\_lock\_typ.sql**

col type for a2 col name for a20

col id1\_tag for a10 col id2\_tag for a10

col description for a40

select type,name,id1\_tag, id2\_tag,description from v$lock\_type where name like '%Sequence%';

SYS@orcl\_1 SQL> **@seq\_lock\_typ**

Hit Any Key to Continue

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 12

TY NAME

ID1\_TAG ID2\_TAG DESCRIPTION

SQ Sequence Cache

object # 0

Lock to ensure that only one process can

replenish the sequence cache

SV Sequence Ordering object # 0

Lock to ensure ordered sequence allocati

on in RAC mode

2 rows selected.

SYS@orcl\_1 SQL>

SQ lock is used both in RAC and single instance to single thread cache replenishment for a specific sequence. The SV lock is used to coordinate number ordering when using RAC and the ORDERED option for a sequence.

1. Next we will create five sequences. First, create a sequence with no order and no cache called **rac\_nocache\_noorder**.

SYS@orcl\_1 SQL> **create sequence rac\_nocache\_noorder**

1. **start with 1 increment by 1**
2. **nocache noorder;**

Sequence created.

SYS@orcl\_1 SQL>

1. Next, create a sequence using order but no cache called **rac\_nocache\_order**.

SYS@orcl\_1 SQL> **create sequence rac\_nocache\_order**

1. **start with 1 increment by 1**
2. **nocache order;**

Sequence created.

SYS@orcl\_1 SQL>

1. Create a sequence using cache but no order called **rac\_cache\_noorder**.

SYS@orcl\_1 SQL> **create sequence rac\_cache\_noorder**

1. **start with 1 increment by 1**
2. **cache 50000 noorder;**

Sequence created.

SYS@orcl\_1 SQL>

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 13

1. Create a sequence using cache and order called **rac\_cache\_order**.

SYS@orcl\_1 SQL> **create sequence rac\_cache\_order**

1. **start with 1 increment by 1**
2. **cache 50000 order;**

Sequence created.

SYS@orcl\_1 SQL>

1. Create another sequence using cache and order called **rac\_gap\_order**.

SYS@orcl\_1 SQL> **create sequence rac\_gap\_order**

1. **start with 1 increment by 1**
2. **cache 50000 order;**

Sequence created.

SYS@orcl\_1 SQL>

1. Let’s take a look at the newly created sequences in the data dictionary using the

seq\_dd.sql script.

SYS@orcl\_1 SQL> **!cat seq\_dd.sql**

col sequence\_owner for a14 col sequence\_name for a20

select SEQUENCE\_NAME, MIN\_VALUE, INCREMENT\_BY, CYCLE\_FLAG, ORDER\_FLAG,

CACHE\_SIZE, LAST\_NUMBER from dba\_sequences where sequence\_name like '%RAC\_%ORDER';

SYS@orcl\_1 SQL> **@seq\_dd.sql**

Hit Any Key to Continue

SEQUENCE\_NAME

MIN\_VALUE INCREMENT\_BY C O CACHE\_SIZE LAST\_NUMBER

- -

RAC\_CACHE\_NOORDER 1 1 N N 50000

RAC\_CACHE\_ORDER 1 1 N Y 50000

RAC\_GAP\_ORDER 1 1 N Y 50000

RAC\_NOCACHE\_NOORDER 1 1 N N 0

RAC\_NOCACHE\_ORDER 1 1 N Y 0

1

1

1

1

1

5 rows selected.

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 14

1. Now let’s look at the sequences in the row cache metadata. Use the seq\_meta.sql script.

SYS@orcl\_1 SQL> **!cat seq\_meta.sql**

select inst\_id,sequence\_name, active\_flag,replenish\_flag,wrap\_flag,cycle\_flag,order\_flag, min\_value,nextvalue,increment\_by,cache\_size

from gv$\_sequences

where sequence\_name like '%RAC%' order by inst\_id,sequence\_name;

SYS@orcl\_1 SQL> @**seq\_meta**

Hit Any Key to Continue

INST\_ID SEQUENCE\_NAME

1 RAC\_CACHE\_NOORDER

1 RAC\_CACHE\_ORDER

1 RAC\_GAP\_ORDER

1 RAC\_NOCACHE\_NOORDER

1 RAC\_NOCACHE\_ORDER

A R W C O MIN\_VALUE NEXTVALUE INCREMENT\_BY CACHE\_SIZE

- - - - -

N N N N N N N Y N Y N N Y N Y N N N N N

N N N N Y

1

1

1

1

1

1

1

1

1

1

50000

50000

50000

0

0

5 rows selected.

SYS@orcl\_1 SQL>

Only Instance 1 currently has metadata in the row cache, since the sequences were created there and not used in any other instance.

1. Let’s take a look at the SV resources currently owned by LCK0 using the sv\_res.sql

script.

SYS@orcl\_1 SQL> **!cat sv\_res.sql**

select l.inst\_id,l.pid,p.program,l.resource\_name2, l.owner\_node,l.state,l.blocked,l.blocker

from gv$ges\_enqueue l, gv$process p where resource\_name2 like '%SV%' and p.spid = l.pid

order by 1;

SYS@orcl\_1 SQL> **@sv\_res.sql**

Hit Any Key to Continue

INST\_ID PID PROGRAM

RESOURCE\_NAME2 OWNER\_NODE STATE BLOCKED BLOCKER

2 rows selected.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 1788 oracle@host02 | (LCK0) |  | 93295,0,SV |  | 0 |  | GRANTED |  | 0 |  | 0 |
| 1 1788 oracle@host02 | (LCK0) |  | 93294,0,SV |  | 0 |  | GRANTED |  | 0 |  | 0 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 15

1. Determine the names for resources 93295 and 93294 (in this example).

SYS@orcl\_1 SQL> col **OBJECT\_NAME format a20**

SYS@orcl\_1 SQL> **select object\_name, object\_type from dba\_objects where object\_id = 93295;**

Hit Any Key to Continue

OBJECT\_NAME

OBJECT\_TYPE

RAC\_GAP\_ORDER

SEQUENCE

1 row selected.

SYS@orcl\_1 SQL> **select object\_name, object\_type from dba\_objects where object\_id = 93294;**

Hit Any Key to Continue

OBJECT\_NAME

OBJECT\_TYPE

RAC\_CACHE\_ORDER

SEQUENCE

1 row selected.

SYS@orcl\_1 SQL>

Only sequences with cache and order get SV instance locks which are required to pass

NEXTVAL to requesting instances.

1. Now, let’s attempt to show gaps occurring in the RAC\_GAP\_ORDER sequence.

SYS**@orcl\_1 SQL> select rac\_gap\_order.nextval from dual;**

Hit Any Key to Continue NEXTVAL

1

1 row selected.

\*\*\* Do it again \*\*\*\*

SYS**@orcl\_1 SQL> select rac\_gap\_order.nextval from dual;**

Hit Any Key to Continue

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 16

NEXTVAL

2

1 row selected.

SYS**@orcl\_1 SQL>**

1. Force a loss of the cached sequence numbers by flushing the shared pool.

SYS@orcl\_1 SQL> **alter system flush shared\_pool;**

System altered.

SYS@orcl\_1 SQL>

1. Increment the RAC\_GAP\_ORDER sequence with cache and order again.

SYS@orcl\_1 SQL> **select rac\_gap\_order.nextval from dual;**

Hit Any Key to Continue NEXTVAL

50001

1 row selected.

SYS@orcl\_1 SQL>

Note that NEXTVAL is now 50001. This gap would occur in both single instance and RAC databases.

1. You must manually keep the sequence to prevent ageouts. This is done using the dbms\_shared\_pool.keep procedure. Keep the RAC\_GAP\_ORDER sequence to prevent this condition.

SYS@orcl\_1 SQL> **exec dbms\_shared\_pool.keep('RAC\_GAP\_ORDER','Q');**

PL/SQL procedure successfully completed.

SYS@orcl\_1 SQL>

1. Increment the RAC\_GAP\_ORDER sequence and then flush the shared pool again. Increment the sequence again. What do you observe?

SYS@orcl\_1 SQL> **select rac\_gap\_order.nextval from dual;**

Hit Any Key to Continue NEXTVAL

50002

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 17

1 row selected.

SYS@orcl\_1 SQL> **alter system flush shared\_pool;**

System altered.

SYS@orcl\_1 SQL> **select rac\_gap\_order.nextval from dual;**

Hit Any Key to Continue NEXTVAL

50003

1 row selected.

SYS@orcl\_1 SQL>

Because the sequence has been kept (for this instance), the cached values are not lost. If you are using cached sequences, you will need to initiate a keep for them when instances are restarted. In a RAC environment, the keep must be performed on all instances.

1. You can query gv$db\_object\_cache to view the keep status of the RAC\_GAP\_ORDER sequence. Drop the RAC\_GAP\_ORDER sequence when finished. Note that it shows as kept only in one instance.

SYS@orcl\_1 SQL> **col name format a20**

SYS@orcl\_1 SQL> **col type format a10**

SYS@orcl\_1 SQL> **select name, type, kept, inst\_id from gv$db\_object\_cache where kept = 'YES' and name like '%RAC\_%';**

SYS@orcl\_1 SQL> **drop sequence rac\_gap\_order;**

Sequence dropped.

SYS@orcl\_1 SQL>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| NAME |  | TYPE |  | KEP |  | INST\_ID |
| RAC\_GAP\_ORDER  1 row selected. |  | SEQUENCE |  | YES |  | 1 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 18

1. Keep the remaining cached sequences. Starting with instance 1 (orcl\_1) keep

RAC\_CACHE\_NOORDER and RAC\_CACHE\_ORDER.

SYS@orcl\_1 SQL> **exec dbms\_shared\_pool.keep ('RAC\_CACHE\_NOORDER','Q');**

PL/SQL procedure successfully completed.

SYS@orcl\_1 SQL> **exec dbms\_shared\_pool.keep ('RAC\_CACHE\_ORDER','Q');**

PL/SQL procedure successfully completed.

SYS@orcl\_1 SQL>

1. Check the keep status of the sequences using the view\_obj\_cache.sql script .

SYS@orcl\_1 SQL> **!cat view\_obj\_cache.sql**

select inst\_id, name, type, kept from gv$db\_object\_cache where name like '%RAC\_CACHE%'

and type = 'SEQUENCE' order by 1,2;

SYS@orcl\_1 SQL> **@view\_obj\_cache**

Hit Any Key to Continue

2 rows selected.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| INST\_ID |  | NAME |  | TYPE |  | KEP |
| 1 |  | RAC\_CACHE\_NOORDER |  | SEQUENCE |  | YES |
| 1 |  | RAC\_CACHE\_ORDER |  | SEQUENCE |  | YES |

1. Now get the first value from the RAC\_NOCACHE\_NOORDER sequence in the first instance. Check SQ enqueues on the first instance using the sq\_enq.sql script.

SYS@orcl\_1 SQL> **select rac\_nocache\_noorder.nextval from dual;**

Hit Any Key to Continue NEXTVAL

1

1 row selected.

SYS@orcl\_1 SQL> **!cat sq\_enq.sql**

select \* from gv$enqueue\_stat where eq\_type = 'SQ' order by inst\_id;

SYS@orcl\_1 SQL> **@sq\_enq**

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 19

Hit Any Key to Continue

INST\_ID EQ TOTAL\_REQ# TOTAL\_WAIT# SUCC\_REQ# FAILED\_REQ# CUM\_WAIT\_TIME CON\_ID

-

***1 SQ 17***

2 SQ 15

3 SQ 20

***0 17 0 0 0***

0 15 0 0 0

0 20 0 0 0

3 rows selected.

SYS@orcl\_1 SQL>

1. Get the first value from the RAC\_NOCACHE\_ORDER sequence in the first instance. Check SQ enqueues on the first instance using the sq\_enq.sql script.

SYS@orcl\_1 SQL> **select rac\_nocache\_order.nextval from dual;**

Hit Any Key to Continue NEXTVAL

1

1 row selected.

SYS@orcl\_1 SQL> **@sq\_enq.sql**

Hit Any Key to Continue

INST\_ID EQ TOTAL\_REQ# TOTAL\_WAIT# SUCC\_REQ# FAILED\_REQ# CUM\_WAIT\_TIME CON\_ID

-

***1 SQ 19***

2 SQ 15

3 SQ 20

***0 19 0 0 0***

0 15 0 0 0

0 20 0 0 0

3 rows selected.

SYS@orcl\_1 SQL>

1. Get the first value from the RAC\_CACHE\_NOORDER sequence in the first instance. Check SQ enqueues on the first instance using the sq\_enq.sql script.

SYS@orcl\_1 SQL> **select rac\_cache\_noorder.nextval from dual;**

Hit Any Key to Continue NEXTVAL

1

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 20

1 row selected.

SYS@orcl\_1 SQL> **@sq\_enq**

Hit Any Key to Continue

INST\_ID EQ TOTAL\_REQ# TOTAL\_WAIT# SUCC\_REQ# FAILED\_REQ# CUM\_WAIT\_TIME CON\_ID

-

***1 SQ 24***

2 SQ 16

3 SQ 21

***0 24 0 0 0***

0 16 0 0 0

0 21 0 0 0

3 rows selected.

SYS@orcl\_1 SQL>

1. Get the first value from the RAC\_CACHE\_ORDER sequence in the first instance. Check SQ enqueues on the first instance using the sq\_enq.sql script.

SYS@orcl\_1 SQL> **select rac\_cache\_order.nextval from dual;**

Hit Any Key to Continue NEXTVAL

1

1 row selected.

SYS@orcl\_1 SQL> **@sq\_enq.sql**

Hit Any Key to Continue

INST\_ID EQ TOTAL\_REQ# TOTAL\_WAIT# SUCC\_REQ# FAILED\_REQ# CUM\_WAIT\_TIME CON\_ID

-

***1 SQ 25***

2 SQ 16

3 SQ 21

***0 25 0 0 0***

0 16 0 0 0

0 21 0 0 0

3 rows selected.

SYS@orcl\_1 SQL>

1. View the sequence data contained in the data dictionary using the seq\_dd.sql script. Compare that with the sequence metadata displayed in GV$\_SEQUENCES using the seq\_meta\_val.sql script.

SYS@orcl\_1 SQL> **@seq\_dd**

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 21

Hit Any Key to Continue

SEQUENCE\_NAME

MIN\_VALUE INCREMENT\_BY C O CACHE\_SIZE LAST\_NUMBER

- -

RAC\_CACHE\_NOORDER 1 1 N N 50000

RAC\_CACHE\_ORDER 1 1 N Y 50000

RAC\_NOCACHE\_NOORDER 1 1 N N 0

RAC\_NOCACHE\_ORDER 1 1 N Y 0

50001

50001

2

2

4 rows selected.

SYS@orcl\_1 SQL> **!cat seq\_meta\_val.sql**

select inst\_id,sequence\_name,order\_flag, nextvalue,cache\_size from gv$\_sequences where sequence\_name like '%RAC\_%ORDER'

order by inst\_id,sequence\_name;

SYS@orcl\_1 SQL> **@seq\_meta\_val**

Hit Any Key to Continue

INST\_ID SEQUENCE\_NAME

1 RAC\_CACHE\_NOORDER

1 RAC\_CACHE\_ORDER

1 RAC\_NOCACHE\_NOORDER

1 RAC\_NOCACHE\_ORDER

O

NEXTVALUE CACHE\_SIZE

-

N Y N

Y

2

2

50000

50000

0

0

4 rows selected.

SYS@orcl\_1 SQL>

In the data dictionary, the two non-cached sequences show a zero cache size as expected, and the LAST\_NUMBER column contains 2 for both of them. This represents the number written to the data dictionary the last time it was updated for those sequences. For non- cached sequences, each time a NEXTVAL is requested, the dictionary will be updated so the LAST\_NUMBER column shows the next value that will be allocated. In the row cache metadata the NEXTVALUE column contains nulls for the non-cached sequences, because no values are cached. For the cached sequences, the cache size is 50000. In the data dictionary, the LAST\_NUMBER column shows 50001 as that number is the value written to the data dictionary for these two sequences when caching the first 50000 numbers for each sequence in the row cache of instance 1. Notice that the NEXTVALUE column contains 2 in the row cache metadata, as that represents the next cached number that will be allocated for each of the cached sequences. The data dictionary will only need to be updated for either of these sequences when cache replenishment occurs, either due to ageouts, or due to exhausting the cached range of numbers.

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 22

1. Connect to the second instance and keep both cached sequences.

SYS@orcl\_1 SQL> **connect sys/*sys\_password*@orcl\_2 as sysdba**

Connected.

SYS@orcl\_2 SQL> **exec dbms\_shared\_pool.keep ('RAC\_CACHE\_NOORDER','Q');**

PL/SQL procedure successfully completed.

SYS@orcl\_2 SQL> **exec dbms\_shared\_pool.keep ('RAC\_CACHE\_ORDER','Q');**

PL/SQL procedure successfully completed. SYS@orcl\_2 SQL>

1. Use the view\_obj\_cache.sql script to view the cached sequences in the second instance.

SYS@orcl\_2 SQL> **@view\_obj\_cache.sql**

Hit Any Key to Continue

4 rows selected.

SYS@orcl\_2 SQL>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| INST\_ID |  | NAME |  | TYPE |  | KEP |
| 1 |  | RAC\_CACHE\_NOORDER |  | SEQUENCE |  | YES |
|  |  | RAC\_CACHE\_ORDER |  | SEQUENCE |  | YES |
| 2 |  | RAC\_CACHE\_NOORDER |  | SEQUENCE |  | YES |
|  |  | RAC\_CACHE\_ORDER |  | SEQUENCE |  | YES |

1. Get the next value for the RAC\_NOCACHE\_NOORDER, RAC\_NOCACHE\_ORDER, RAC\_CACHE\_NOORDER, and RAC\_CACHE\_ORDER sequences.

SYS@orcl\_2 SQL> **select rac\_nocache\_noorder.nextval from dual;**

Hit Any Key to Continue NEXTVAL

2

1 row selected.

SYS@orcl\_2 SQL> **select rac\_nocache\_order.nextval from dual;**

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 23

Hit Any Key to Continue NEXTVAL

2

1 row selected.

SYS@orcl\_2 SQL> **select rac\_cache\_noorder.nextval from dual;**

Hit Any Key to Continue NEXTVAL

50001

1 row selected.

SYS@orcl\_2 SQL> **select rac\_cache\_order.nextval from dual;**

Hit Any Key to Continue NEXTVAL

2

1 row selected.

SYS@orcl\_2 SQL>

1. From the second instance, view the sequence data contained in the data dictionary using the seq\_dd.sql script. Compare that with the sequence metadata displayed in GV$\_SEQUENCES using the seq\_meta\_val.sql script.

SYS@orcl\_2 SQL> **@seq\_dd**

Hit Any Key to Continue

SEQUENCE\_NAME

MIN\_VALUE INCREMENT\_BY C O CACHE\_SIZE LAST\_NUMBER

- -

RAC\_CACHE\_NOORDER 1 1 N N 50000

RAC\_CACHE\_ORDER 1 1 N Y 50000

RAC\_NOCACHE\_NOORDER 1 1 N N 0

RAC\_NOCACHE\_ORDER 1 1 N Y 0

100001

50001

3

3

4 rows selected.

SYS@orcl\_2 SQL> **@seq\_meta\_val**

Hit Any Key to Continue

INST\_ID SEQUENCE\_NAME O NEXTVALUE CACHE\_SIZE

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 24

1 RAC\_CACHE\_NOORDER RAC\_CACHE\_ORDER RAC\_NOCACHE\_NOORDER

RAC\_NOCACHE\_ORDER

-

N Y N

Y

2

2

50000

50000

0

0

2 RAC\_CACHE\_NOORDER RAC\_CACHE\_ORDER RAC\_NOCACHE\_NOORDER

RAC\_NOCACHE\_ORDER

N Y N

Y

50002

3

50000

50000

0

0

8 rows selected.

SYS@orcl\_2 SQL>

By now you should have noticed that the non-cached sequences hit the data dictionary for each NEXTVAL requested while the cached sequences are not incremented in the data dictionary because NEXTVAL is supplied from memory.

Note too that the sequence that is not ordered causes another range of 50000 numbers to be cached in the second row cache. This is why the LAST\_NUMBER column in the data dictionary contains 100001 for RAC\_CACHE\_NOORDER. The value for RAC\_CACHE\_ORDER remains at 50001 since all instances are in effect caching the same range of numbers. This can also be seen in the row cache metadata. RAC\_CACHE\_NOORDER has a NEXTVALUE of 2 in one instance but a NEXTVALUE of 50002 in another and no coordination is done to allocate the numbers in order. RAC\_CACHE\_ORDER also has two different values for NEXTVALUE having a 2 in one instance and a 3 in the other but the instance with the highest value in the row cache NEXTVALUE column owns the “true” next value and if another instance increments that sequence, the “true” next value will be passed by transferring the value in the SV lock to the requesting instance.

1. Connect back to the first instance and get the next value for the RAC\_NOCACHE\_NOORDER, RAC\_NOCACHE\_ORDER, .RAC\_CACHE\_NOORDER, and RAC\_CACHE\_ORDER sequences.

SYS@orcl\_2 SQL> **connect sys/*sys\_password*@orcl\_1 as sysdba**

Connected.

SYS@orcl\_1 SQL> **select rac\_nocache\_noorder.nextval from dual;**

Hit Any Key to Continue NEXTVAL

3

1 row selected.

SYS@orcl\_1 SQL> **select rac\_nocache\_order.nextval from dual;**

Hit Any Key to Continue NEXTVAL

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 25

3

1 row selected.

SYS@orcl\_1 SQL> **select rac\_cache\_noorder.nextval from dual;**

Hit Any Key to Continue NEXTVAL

2

1 row selected.

SYS@orcl\_1 SQL> **select rac\_cache\_order.nextval from dual;**

Hit Any Key to Continue NEXTVAL

3

1 row selected.

SYS@orcl\_1 SQL>

1. View the sequence data contained in the data dictionary using the seq\_dd.sql script. Compare that with the sequence metadata displayed in GV$\_SEQUENCES using the seq\_meta\_val.sql script. What are your observations?

SYS@orcl\_1 SQL> **@seq\_dd.sql**

Hit Any Key to Continue

SEQUENCE\_NAME

MIN\_VALUE INCREMENT\_BY C O CACHE\_SIZE LAST\_NUMBER

- -

RAC\_CACHE\_NOORDER 1 1 N N 50000

RAC\_CACHE\_ORDER 1 1 N Y 50000

RAC\_NOCACHE\_NOORDER 1 1 N N 0

RAC\_NOCACHE\_ORDER 1 1 N Y 0

100001

50001

4

4

4 rows selected.

SYS@orcl\_1 SQL> **@seq\_meta\_val**

Hit Any Key to Continue INST\_ID SEQUENCE\_NAME

1 RAC\_CACHE\_NOORDER

O

NEXTVALUE CACHE\_SIZE

-

N

3

50000

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 26

|  |  |  |  |
| --- | --- | --- | --- |
| RAC\_CACHE\_ORDER | Y | 4 | 50000 |
| RAC\_NOCACHE\_NOORDER | N |  | 0 |
| RAC\_NOCACHE\_ORDER | Y |  | 0 |
| 2 RAC\_CACHE\_NOORDER | N | 50002 | 50000 |
| RAC\_CACHE\_ORDER | Y | 3 | 50000 |
| RAC\_NOCACHE\_NOORDER | N |  | 0 |
| RAC\_NOCACHE\_ORDER | Y |  | 0 |

The non-cached sequences update the data dictionary with each NEXTVAL, potentially causing performance degradation**.** The best performance is usually obtained using cached and ordered sequences.

8 rows selected.

SYS@orcl\_1 SQL>

The RAC\_CACHE\_ORDER sequence got the correct value of 3, due to the resource coordination provided by the SV lock. The RAC\_CACHE\_NOORDER sequence got the number 2. So the order of allocation for RAC\_CACHE\_ORDER was 1, 2, 3 as required. The order of allocation for RAC\_CACHE\_NOORDER was 1, 50001, 2.

1. Drop all sequences created in this practice.

SYS@orcl\_1 SQL> **drop sequence rac\_nocache\_noorder;**

Sequence dropped.

SYS@orcl\_1 SQL> **drop sequence rac\_nocache\_order;**

Sequence dropped.

SYS@orcl\_1 SQL> **drop sequence rac\_cache\_noorder;**

Sequence dropped.

SYS@orcl\_1 SQL> **drop sequence rac\_cache\_order;**

Sequence dropped.

SYS@orcl\_1 SQL>

1. Close all terminal windows opened for this practice.

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 27

# Practice 15-3: Library Cache Coordination for Procedure and Packages

### Overview

In this lesson you explore the behavior of the library cache in a RAC environment.

1. Establish an ssh connection as the oracle user. Use oraenv to set the Oracle environment.

[vncuser@*classsroom\_pc* ~]$ **ssh oracle@host01**

oracle@host01's password:

[oracle@host01 ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orcl**

The Oracle base has been set to /u01/app/oracle [oracle@host01 ~]$

1. Change the directory to /stage/RAC/labs/less\_07, check which instance is running on

host01, and set ORACLE\_SID.

[oracle@host01 ~]$ **cd /stage/RAC/labs/less\_07**

[oracle@host01 less\_07]$ **ps -ef|grep ora\_smon**

[oracle@host01 less\_07]$ **export ORACLE\_SID=orcl\_3**

[oracle@host01 less\_07]$

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| oracle | 12256 | 1 | 0 Dec09 ? | 00:00:19 ora\_smon\_orcl\_3 |
| oracle | 29985 | 28381 | 0 17:50 pts/1 | 00:00:00 grep ora\_smon |

1. Start SQL\*Plus and verify the SID using the show\_my\_sid.sql script.

[oracle@host01 less\_07]$ **sqlplus / as sysdba**

SQL\*Plus:Release 12.1.0.2 Production on Wed Dec 10 19:22:12 2014 Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,

Advanced Analytics and Real Application Testing options

SYS@orcl\_3 SQL> **@show\_my\_sid**

Hit Any Key to Continue

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 28

SID 73

1 row selected.

SYS@orcl\_3 SQL>

1. Next, create a stored procedure using the lc\_demo.sql script. Take a moment and look at the script to see what it does.

SYS@orcl\_3 SQL> !**cat lc\_demo.sql**

create or replace procedure lcdemo is begin

for i in 1 ..100000 loop

dbms\_lock.sleep (1); end loop;

end;

/

SYS@orcl\_3 SQL>

The script creates a long running procedure called lcdemo.

1. Run the lc\_demo.sql script to compile the lcdemo procedure, and then execute it.

SYS@orcl\_3 SQL> **@lc\_demo.sql**

Procedure created. SYS@orcl\_3 SQL> **exec lcdemo**

█

The session essentially hangs while it runs the lcdemo procedure.

1. Open another terminal window to host01, set the environment, change the directory to

/stage/RAC/labs/less\_07 and connect to the same instance. Use the

show\_my\_sid.sql script to display the SID.

[vncuser@*classsroom\_pc* ~]$ **ssh oracle@host01**

oracle@host01's password:

[oracle@host01 ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orcl**

The Oracle base has been set to /u01/app/oracle

[oracle@host01 ~]$ **export ORACLE\_SID=orcl\_3**

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 29

[oracle@host01 ~]$ **cd /stage/RAC/labs/less\_07**

[oracle@host01 less\_07]$ **sqlplus / as sysdba**

SQL\*Plus: Release 12.1.0.2.0 Production on Wed Jan 28 12:51:28 2015 Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,

Advanced Analytics and Real Application Testing options

SYS@orcl\_3 SQL> **@show\_my\_sid**

Hit Any Key to Continue

SID 47

1 row selected.

SYS@orcl\_3 SQL>

1. From the second terminal session, query V$SESSION and see what is held by the first SID.

SYS@orcl\_3 SQL> **select sid, event from v$session where sid = 73;**

Hit Any Key to Continue SID EVENT

73 PL/SQL lock timer

1 row selected.

SYS@orcl\_3 SQL>

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 30

1. From the second terminal, attempt to drop the lcdemo procedure.

SYS@orcl\_3 SQL> **drop procedure lcdemo;**

█

The session now hangs trying to drop the lcdemo preocedure.

1. Now, open a third terminal window to host01. Set the environment, change the directory to

/stage/RAC/labs/less\_07 and connect to the same instance. Query V$SESSION to see what the session attempting to drop the lcdemo preocedure is waiting on.

[vncuser@*classsroom\_pc* ~]$ **ssh oracle@host01**

oracle@host01's password:

[oracle@host01 ~]$ **. oraenv**

ORACLE\_SID = [oracle] ? **orcl**

The Oracle base has been set to /u01/app/oracle [oracle@host01 ~]$ **export ORACLE\_SID=orcl\_3** [oracle@host01 ~]$ **cd /stage/RAC/labs/less\_07** [oracle@host01 less\_07]$ **sqlplus / as sysdba**

SQL\*Plus: Release 12.1.0.2.0 Production on Wed Jan 28 12:51:28 2015 Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production

With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,

Advanced Analytics and Real Application Testing options

SYS@orcl\_3 SQL> **select sid, event from v$session where sid=47;**

Hit Any Key to Continue SID EVENT

47 library cache pin

1 row selected.

SYS@orcl\_3 SQL>

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 31

The dropping session is waiting on the library cache pin held by the first session. This behavior is expected in a single-instance scenario as shown here.

1. Return to the second terminal session. Cancel the drop procedure lcdemo statement by typing <Control-c>. Connect to a different instance. In this example, we will connect to orcl\_1. Use the show\_my\_sid.sql script to determine the session identifier.

SYS@orcl\_3 SQL> **drop procedure lcdemo;**

* **<Control-C>**

SYS@orcl\_3 SQL> **connect sys/*sys\_password*@orcl\_1 as sysdba**

Connected.

SYS@orcl\_1 SQL> **@show\_my\_sid**

Hit Any Key to Continue

SID 18

1 row selected.

SYS@orcl\_1 SQL>

1. Attempt to drop the lcdemo procedure from the session connected to orcl\_1. What happens?

SYS@orcl\_1 SQL> **drop procedure lcdemo;**

█

Again, the statement hangs.

1. Go to the third terminal session and connect to orcl\_1. Query V$SESSION to check what the session attempting to drop the lcdemo preocedure is waiting on.

SYS@orcl\_3 SQL> **connect sys/*sys\_password*@orcl\_1 as sysdba**

Connected.

SYS@orcl\_1 SQL> **select sid, event from v$session where sid = 18;**

Hit Any Key to Continue SID EVENT

18 library cache pin

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 32

1 row selected.

SYS@orcl\_1 SQL>

The session connected to orcl\_1 attempting to drop the lcdemo is waiting on a library cache pin held by the session connected to orcl\_3 currently executing that procedure. This indicates that acquisition of library cache objects is coordinated in a RAC environment among all instances in the cluster.

1. Close all terminals opened for this practice.

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 33

# Practice 15-4: Global Enqueue Coordination

### Overview

This practice will demonstrate how Global Enqueues are coordinated in an Oracle RAC environment.

1. Open a terminal session on your desktop to host01 as the oracle user.

|  |
| --- |
| [vncuser@*classsroom\_pc* ~]$ **ssh oracle@host01**  oracle@host01's password:  Last login: Wed Dec 10 17:47:14 2014 from 192.0.2.1  [oracle@host01 ~]$ |

1. Change the directory to /stage/RAC/labs/less\_07, check which instance is running on

host01, and set the environment.

|  |
| --- |
| [oracle@host01 ~]$ **cd /stage/RAC/labs/less\_07**  [oracle@host01 less\_07]$ **ps -ef|grep ora\_smon**  oracle 12256 1 0 Dec09 ? 00:00:19 ora\_smon\_orcl\_3  oracle 29985 28381 0 17:50 pts/1 00:00:00 grep ora\_smon  [oracle@host01 less\_07]$ **. oraenv**  ORACLE\_SID = [oracle] ? **orcl**  The Oracle base has been set to /u01/app/oracle  [oracle@host01 less\_07]$ **export ORACLE\_SID=orcl\_3**  [oracle@host01 less\_07]$ |

1. Start SQL\*Plus and determine the SID. Run the query shown below against GV$LOCK to show lock specifics. The query will only return rows when TYPE is TM (DML enqueue). The query should return no rows at this stage.

|  |
| --- |
| [oracle@host01 less\_07]$ **sqlplus / as sysdba**  SQL\*Plus:Release 12.1.0.2 Production on Wed Dec 10 19:22:12 2014 Copyright (c) 1982, 2014, Oracle. All rights reserved.  Connected to:  Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production  With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,  Advanced Analytics and Real Application Testing options |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 34

|  |
| --- |
| SYS@orcl\_3 SQL> **select distinct sid from v$mystat;**  Hit Any Key to Continue  SID 61  1 row selected.  SYS@orcl\_3 SQL> **select inst\_id, sid, type, id1, id2, lmode, request, block from gv$lock where type = 'TM' order by 1,2;**  no rows selected SYS@orcl\_3 SQL> |

1. Lock the HR.EMPLOYEES table in exclusive mode. Rerun the GV$LOCK query. The instance ID (INST\_ID) should be 3, and lock mode (LMODE) should be 6 (EXCLUSIVE). BLOCK should be 2, meaning that instances on other nodes are blocked from acquiring a lock on the object. Query DBA\_OBJECTS by using the value of ID1 to verify that the results pertain to the HR.EMPLOYEES table.

|  |
| --- |
| SYS@orcl\_3 SQL> **lock table hr.employees in exclusive mode;**  Table(s) Locked.  SYS@orcl\_3 SQL> **select inst\_id, sid, type, id1, id2, lmode, request, block from gv$lock where type = 'TM' order by 1,2;**  Hit Any Key to Continue  INST\_ID SID TY ID1 ID2 LMODE REQUEST BLOCK 3 61 TM 92593 0 6 0 2  1 row selected.  SYS@orcl\_3 SQL> **col owner format a10**  SYS@orcl\_3 SQL> **col object\_name format a15**  SYS@orcl\_3 SQL> **select owner, object\_name from dba\_objects where object\_id = 92593;**  Hit Any Key to Continue  OWNER OBJECT\_NAME    HR EMPLOYEES |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 35

|  |
| --- |
| 1 row selected.  SYS@orcl\_3 SQL> |

1. From your desktop open a second terminal to host01 as the oracle user. Change the directory to /stage/RAC/labs/less\_07 and set the environment.

|  |
| --- |
| [vncuser@*classsroom\_pc* ~]$ **ssh oracle@host01**  oracle@host01's password:  Last login: Wed Dec 10 17:47:27 2014 from 192.0.2.1 [oracle@host01 ~]$ **cd /stage/RAC/labs/less\_07**  [oracle@host01 less\_07]$ **. oraenv**  ORACLE\_SID = [oracle] ? **orcl**  The Oracle base has been set to /u01/app/oracle [oracle@host01 less\_07]$ **export ORACLE\_SID=orcl\_3** [oracle@host01 less\_07]$ **sqlplus / as sysdba**  SQL\*Plus:Release 12.1.0.2 Production on Thu Dec 11 11:20:49 2014 Copyright (c) 1982, 2014, Oracle. All rights reserved.  Connected to:  Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production  With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,  Advanced Analytics and Real Application Testing options  SYS@orcl\_3 SQL> **select distinct sid from v$mystat;**  Hit Any Key to Continue  SID 67  1 row selected.  SYS@orcl\_3 SQL> |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 36

1. From the second terminal window, attempt to lock the HR.EMPLOYEES table in exclusive mode. What do you observe?

|  |
| --- |
| SYS@orcl\_3 SQL> **lock table hr.employees in exclusive mode;** |

1. Return to the first terminal window and rerun the query from step 5. What do you observe?

|  |
| --- |
| SYS@orcl\_3 SQL> **select inst\_id, sid, type, id1, id2, lmode, request, block from gv$lock where type = 'TM' order by 1,2;**  Hit Any Key to Continue  INST\_ID SID TY ID1 ID2 LMODE REQUEST BLOCK 3 61 TM 92593 0 6 0 1  67 TM 92593 0 0 6 0  2 rows selected.  SYS@orcl\_3 SQL> |

1. Open another terminal window. This time connect to host02 as the oracle user. Change the directory to /stage/RAC/labs/less\_07 and check which instance is running on host02. Set the environment, start SQL\*Plus, and determine the SID.

|  |
| --- |
| [vncuser@*classsroom\_pc* ~]$ **ssh oracle@host02**  oracle@host02's password:  [oracle@host02 ~]$ **cd /stage/RAC/labs/less\_07**  [oracle@host02 less\_07]$ **ps -ef|grep ora\_smon**  oracle 12308 1 0 Dec09 ? 00:00:32 ora\_smon\_orcl\_1  oracle 26462 26298 0 11:49 pts/0 00:00:00 grep ora\_smon  [oracle@host02 less\_07]$ **. oraenv**  ORACLE\_SID = [oracle] ? **orcl**  The Oracle base has been set to /u01/app/oracle [oracle@host02 less\_07]$ export ORACLE\_SID=**orcl\_1** [oracle@host02 less\_07]$ **sqlplus / as sysdba**  SQL\*Plus:Release 12.1.0.2 Production on Thu Dec 11 11:50:14 2014  Copyright (c) 1982, 2014, Oracle. All rights reserved. |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 37

|  |
| --- |
| Connected to:  Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production  With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,  Advanced Analytics and Real Application Testing options  SYS@orcl\_1 SQL> **select distinct sid from v$mystat;**  Hit Any Key to Continue  SID 75  1 row selected.  SYS@orcl\_1 SQL> |

1. From the third terminal window (host02) attempt to lock the HR.EMPLOYEES table in exclusive mode. What do you observe?

|  |
| --- |
| SYS@orcl\_1 SQL> **lock table hr.employees in exclusive mode;** |

1. Return to the first terminal session and rerun the query from step 7 by entering / and pressing Enter. What do you observe?

|  |
| --- |
| SYS@orcl\_3 SQL> **/**  Hit Any Key to Continue  INST\_ID SID TY ID1 ID2 LMODE REQUEST BLOCK 1 75 TM 92593 0 0 6 0  3 61 TM 92593 0 6 0 1  67 TM 92593 0 0 6 0  3 rows selected.  SYS@orcl\_3 SQL> |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 38

1. Open another terminal window. This time connect to host03 as the oracle user. Change the directory to /stage/RAC/labs/less\_07 and check which instance is running on host03. Set the environment, start SQL\*Plus, and determine the SID.

|  |
| --- |
| [vncuser@*classsroom\_pc* ~]$ **ssh oracle@host03**  The authenticity of host 'host03 (192.0.2.103)' can't be established.  RSA key fingerprint is e8:aa:00:2c:2e:5c:e4:d8:fe:fd:9b:3f:8c:8b:d4:0b.  Are you sure you want to continue connecting (yes/no)? yes  Warning: Permanently added 'host03,192.0.2.103' (RSA) to the list of known hosts.  oracle@host03's password:  [oracle@host03 ~]$ **cd /stage/RAC/labs/less\_07**  [oracle@host03 less\_07]$ ps -ef|grep ora\_smon  oracle 18647 1 0 Dec09 ? 00:00:36 ora\_smon\_orcl\_2  oracle 28904 28703 0 14:29 pts/0 00:00:00 grep ora\_smon  [oracle@host03 less\_07]$ **. oraenv**  ORACLE\_SID = [oracle] ? **orcl**  The Oracle base has been set to /u01/app/oracle [oracle@host03 less\_07]$ **export ORACLE\_SID=orcl\_2** [oracle@host03 less\_07]$ **sqlplus / as sysdba**  SQL\*Plus: Release 12.1.0.2.0 Production on Thu Dec 11 14:30:14  2014  Copyright (c) 1982, 2014, Oracle. All rights reserved.  Connected to:  Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production  With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,  Advanced Analytics and Real Application Testing options  SYS@orcl\_2 SQL> **select distinct sid from v$mystat;**  Hit Any Key to Continue  SID |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 39

|  |
| --- |
| 38  1 row selected.  SYS@orcl\_2 SQL> |

1. From the new terminal, attempt to lock the HR.EMPLOYERES table in exclusive mode.

|  |
| --- |
| SYS@orcl\_2 SQL> **lock table hr.employees in exclusive mode;** |

1. Return to the first terminal window and rerun the query to view the lock status. What do you observe?

|  |
| --- |
| SYS@orcl\_3 SQL> **/**  Hit Any Key to Continue  INST\_ID SID TY ID1 ID2 LMODE REQUEST BLOCK 1 75 TM 92593 0 0 6 0  2 38 TM 92593 0 0 6 0  3 61 TM 92593 0 6 0 1  67 TM 92593 0 0 6 0  4 rows selected.  SYS@orcl\_3 SQL> |

Instance 3, SID 61 is holding the lock in exclusive mode (LMODE 6), blocking SID 67 (inst 3), 38 (inst 2), and SID 75 (inst 1) all requesting the lock in exclusive mode (REQUEST 6).

1. Go to the first terminal session, (INST\_ID 3, SID 61 in this example) and issue a rollback. Query GV$LOCK when finished. What do you observe?

|  |
| --- |
| SYS@orcl\_3 SQL> **rollback;**  Rollback complete.  SYS@orcl\_3 SQL> **select inst\_id, sid, type, id1, id2, lmode, request, block from gv$lock where type = 'TM' order by 1,2;**  Hit Any Key to Continue  INST\_ID SID TY ID1 ID2 LMODE REQUEST BLOCK |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 40

|  |
| --- |
| 1 75 TM 92593 0 6 0 2  2 38 TM 92593 0 0 6 0  3 67 TM 92593 0 0 6 0  3 rows selected.  SYS@orcl\_3 SQL> |

SID 75 (inst 1) holds the lock, the BLOCK value is 2, meaning the lock is not blocking any blocked processes on the local node, but it may or may not be blocking processes on remote nodes. SID 38 (inst 2) and SID 67 (inst 3) are waiting for the lock in exclusive mode (LMODE 6).

1. Go to the terminal window for instance 1 (host02 in this example). Issue a rollback. When complete, go back to the first terminal window (host01) and query GV$LOCK again. What do you see now?

|  |
| --- |
| SYS@orcl\_1 SQL> **rollback;**  Rollback complete.  SYS@orcl\_1 SQL> SYS@orcl\_3 SQL> **/**  Hit Any Key to Continue  INST\_ID SID TY ID1 ID2 LMODE REQUEST BLOCK 2 38 TM 92593 0 6 0 2  3 67 TM 92593 0 0 6 0  2 rows selected.  SYS@orcl\_3 SQL> |

SID 38 (inst 1) holds the lock, the BLOCK value is 2, meaning the lock is not blocking any blocked processes on the local node, but it may or may not be blocking processes on remote nodes. SID 67 (inst 3) is waiting for the lock in exclusive mode (LMODE 6).

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 41

1. Go to the terminal window for instance 2 (host03 in this example). Issue a rollback. When complete, go back to the first terminal window (host01) and query GV$LOCK again. What do you see now?

|  |
| --- |
| SYS@orcl\_2 SQL> **rollback;**  Rollback complete. SYS@orcl\_2 SQL> SYS@orcl\_3 SQL> /  Hit Any Key to Continue  INST\_ID SID TY ID1 ID2 LMODE REQUEST BLOCK 3 67 TM 92593 0 6 0 2  1 row selected.  SYS@orcl\_3 SQL> |

SID 67 (inst 3) now holds the lock, the BLOCK value is mode 2.

1. Go to the terminal window for instance 3, (host01 and SID 67 in this example). Issue a rollback. When complete, go back to the first terminal window (host01) and query GV$LOCK again. What do you see now?

|  |
| --- |
| SYS@orcl\_3 SQL> rollback; Rollback complete.  SYS@orcl\_3 SQL>  ### Switch terminal windows ###  SYS@orcl\_3 SQL> / no rows selected  SYS@orcl\_3 SQL> |

1. Close all terminals opened for this practice.

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 42

# Practice 15-5: Results Cache Coordination

### Overview

In this practice you will enable Results Cache for a table and examine Results Cache coordination in an Oracle RAC database.

1. Open a terminal session on your desktop to host01 as the oracle user.

|  |
| --- |
| [vncuser@*classsroom\_pc* ~]$ **ssh oracle@host01**  oracle@host01's password:  Last login: Wed Dec 10 17:47:14 2014 from 192.0.2.1  [oracle@host01 ~]$ |

1. Change directory to /stage/RAC/labs/less\_07, check which instance is running on

host01, and set the environment.

|  |
| --- |
| [oracle@host01 ~]$ **cd /stage/RAC/labs/less\_07**  [oracle@host01 less\_07]$ **ps -ef|grep ora\_smon**  oracle 12256 1 0 Dec09 ? 00:00:19 ora\_smon\_orcl\_3  oracle 29985 28381 0 17:50 pts/1 00:00:00 grep ora\_smon  [oracle@host01 less\_07]$ **. oraenv**  ORACLE\_SID = [oracle] ? **orcl**  The Oracle base has been set to /u01/app/oracle  [oracle@host01 less\_07]$ **export ORACLE\_SID=orcl\_3**  [oracle@host01 less\_07]$ |

1. Start SQL\*Plus and determine the SID. Issue the SQL statement below to force result cache creation for the HR.EMPLOYEES table. Find the object ID for the table by querying DBA\_OBJECTS.

|  |
| --- |
| [oracle@host01 less\_07]$ **sqlplus / as sysdba**  SQL\*Plus:Release 12.1.0.2 Production on Wed Dec 10 19:22:12 2014 Copyright (c) 1982, 2014, Oracle. All rights reserved.  Connected to:  Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production  With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,  Advanced Analytics and Real Application Testing options |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 43

|  |
| --- |
| SYS@orcl\_3 SQL> **alter table hr.employees result\_cache (mode force);**  Table altered.  SYS@orcl\_3 SQL> **select object\_id from dba\_objects where owner = 'HR' and object\_name = 'EMPLOYEES';**  Hit Any Key to Continue  OBJECT\_ID  92593  1 row selected.  SYS@orcl\_3 SQL> |

1. Query DBA\_TABLES to confirm that a result cache has been enabled for the EMPLOYEES

table.

|  |
| --- |
| SYS@orcl\_3 SQL> **col owner format a10** SYS@orcl\_3 SQL> **col table\_name format a15** SYS@orcl\_3 SQL> **col result\_cache format a15**  SYS@orcl\_3 SQL> **select owner, table\_name, result\_cache from dba\_tables where table\_name = 'EMPLOYEES';**  Hit Any Key to Continue  OWNER TABLE\_NAME RESULT\_CACHE HR EMPLOYEES FORCE  1 row selected.  SYS@orcl\_3 SQL> |

1. Let’s flush the results cache for all the three instances. Start with the instance running on host01, then open terminal windows to host02 and host03, open a SQL\*Plus connection to the local instance, and flush the results cache for them also.

|  |
| --- |
| SYS@orcl\_3 SQL> **exec dbms\_result\_cache.flush;**  PL/SQL procedure successfully completed. SYS@orcl\_3 SQL>  [vncuser@*classsroom\_pc* ~]$ **ssh oracle@host02**  oracle@host02's password: |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 44

|  |
| --- |
| Last login: Wed Dec 10 17:47:14 2014 from 192.0.2.1  [oracle@host02 ~]$ **ps -ef|grep ora\_smon**  oracle 5033 4984 0 07:12 pts/0 00:00:00 grep ora\_smon  oracle 18647 1 0 Dec09 ? 00:03:03 ora\_smon\_orcl\_1 [oracle@host02 ~]$ **cd /stage/RAC/labs/less\_07**  [oracle@host02 less\_07]$ **. oraenv**  ORACLE\_SID = [oracle] ? **orcl**  The Oracle base has been set to /u01/app/oracle [oracle@host02 less\_07]$ **export ORACLE\_SID=orcl\_1** [oracle@host02 less\_07]$ **sqlplus / as sysdba**  SQL\*Plus:Release 12.1.0.2 Production on Tue Dec 16 07:23:31 2014  ...  SYS@orcl\_1 SQL> **exec dbms\_result\_cache.flush;**  PL/SQL procedure successfully completed. SYS@orcl\_1 SQL>  [vncuser@*classsroom\_pc* ~]$ **ssh oracle@host03**  oracle@host03's password:  Last login: Wed Dec 10 17:47:14 2014 from 192.0.2.1  [oracle@host03 ~]$ **ps -ef|grep ora\_smon**  oracle 5033 4984 0 07:12 pts/0 00:00:00 grep ora\_smon  oracle 18647 1 0 Dec09 ? 00:03:03 ora\_smon\_orcl\_2 [oracle@host03 ~]$ **cd /stage/RAC/labs/less\_07**  [oracle@host03 less\_07]$ **. oraenv**  ORACLE\_SID = [oracle] ? **orcl**  The Oracle base has been set to /u01/app/oracle  [oracle@host03 less\_07]$ **export ORACLE\_SID=orcl\_2** |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 45

|  |
| --- |
| [oracle@host03 less\_07]$ **sqlplus / as sysdba**  SQL\*Plus:Release 12.1.0.2 Production on Tue Dec 16 07:23:31 2014  ...  SYS@orcl\_2 SQL> **exec dbms\_result\_cache.flush;**  PL/SQL procedure successfully completed. SYS@orcl\_2 SQL> |

1. Return to the host01 terminal and query GV$\_RESULT\_CACHE\_STATISTICS and check initial results cache statistics for all three instances. Take note of the values for BLOCK COUNT CURRENT, CREATE COUNT SUCCESS, and GLOBAL HIT COUNT.

|  |
| --- |
| SYS@orcl\_3 SQL> **select inst\_id, name, value from gv$result\_cache\_statistics order by 1,2;**  Hit Any Key to Continue  INST\_ID NAME VALUE     1. Block Count Current 0   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 0   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 46

|  |
| --- |
| Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193  3 Block Count Current 0  Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  3 Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193 |

1. From the host01 terminal, populate the results cache.

|  |
| --- |
| SYS@orcl\_3 SQL> **set autotrace traceonly**  SYS@orcl\_3 SQL> **select \* from hr.employees;**  Hit Any Key to Continue  107 rows selected.  Execution Plan  Plan hash value: 1445457117  | Id | Operation | Name | Rows | Bytes| Cost (%CPU)| Time |  | 0 | SELECT STATEMENT | | 107| 7383 | 3 (0)| 00:00:01 |  **| 1 | RESULT CACHE |afz18h8m8pm02td181yj8 | | | | |** |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 47

|  |
| --- |
| | 2 | TABLE ACCESS FULL| EMPLOYEES | 107 | 7383 | 3 (0)| 00:00:01 |  Result Cache Information (identified by operation id):  1 - column-count=11; dependencies=(HR.EMPLOYEES); name="select \* from hr.employees"  Statistics  75 recursive calls  0 db block gets  189 consistent gets  **19 physical reads**  0 redo size  10121 bytes sent via SQL\*Net to client  628 bytes received via SQL\*Net from client  9 SQL\*Net roundtrips to/from client  5 sorts (memory)  0 sorts (disk)  107 rows processed  **SYS@orcl\_3 SQL> set autotrace off SYS@orcl\_3 SQL>** |

Note the value for physical reads and the step for the result cache in the explain plan.

1. Again, query GV$\_RESULT\_CACHE\_STATISTICS and re-check results cache statistics for all the three instances.

|  |
| --- |
| SYS@orcl\_3 SQL> **select inst\_id, name, value from gv$result\_cache\_statistics order by 1,2;**  Hit Any Key to Continue  INST\_ID NAME VALUE    1 Block Count Current 0  Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 48

|  |
| --- |
| Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 0   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 32   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 1  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193  SYS@orcl\_3 SQL> |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 49

Note that Create Count Success for the local instance (orcl\_3) has been incremented by one, indicating results cache creation.

1. Rerun the query from the previous step.

|  |
| --- |
| SYS@orcl\_3 SQL> **set autotrace traceonly**  SYS@orcl\_3 SQL> **select \* from hr.employees;**  Hit Any Key to Continue  107 rows selected.  Execution Plan  Plan hash value: 1445457117  |Id | Operation | Name |Rows |Bytes|Cost (%CPU)| Time |  | 0 | SELECT STATEMENT | | 107 | 7383| 3 (0)| 00:00:01 |  | 1 | RESULT CACHE | **afz18h8m8pm02td181yj8** | | | | |  | 2 | TABLE ACCESS FULL| EMPLOYEES | 107| 7383| 3 (0)| 00:00:01 |  Result Cache Information (identified by operation id):  1 - column-count=11; dependencies=(HR.EMPLOYEES); name="select \* from hr.employees"  Statistics  0 recursive calls  0 db block gets  0 consistent gets  0 physical reads  0 redo size  10121 bytes sent via SQL\*Net to client  628 bytes received via SQL\*Net from client  9 SQL\*Net roundtrips to/from client  0 sorts (memory)  0 sorts (disk)  107 rows processed  **SYS@orcl\_3 SQL> set autotrace off SYS@orcl\_3 SQL>** |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 50

Note that the explain plan is the same as in the previous step (including result cache identifier) but that there are no db block gets, consistent gets, or physical reads, indicating the results cache was used.

1. Again, query GV$\_RESULT\_CACHE\_STATISTICS from the host01 (instance 3) terminal session and re-check results cache statistics for all three instances. What do you see?

|  |
| --- |
| SYS@orcl\_3 SQL> **select inst\_id, name, value from gv$result\_cache\_statistics order by 1,2;**  Hit Any Key to Continue  INST\_ID NAME VALUE     1. Block Count Current 0   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 0   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 51

|  |
| --- |
| 3 Block Count Current 32  Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 1  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 1  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193  SYS@orcl\_3 SQL> |

Note that the find count for instance 3 has been incremented by 1 and that block counts have not been incremented indicating a result cache hit.

1. Next, let’s invalidate the results cache by updating the EMPLOYEES table.

|  |
| --- |
| SYS@orcl\_3 SQL> **update hr.employees set salary = salary + 0;**  107 rows updated.  SYS@orcl\_3 SQL> **commit;**  Commit complete.  SYS@orcl\_3 SQL> |

1. Set autotrace as shown below and repeat the EMPLOYEES query. What do you observe?

|  |
| --- |
| SYS@orcl\_3 SQL> **set autotrace traceonly**  SYS@orcl\_3 SQL> **select \* from hr.employees;**  Hit Any Key to Continue  107 rows selected.  Execution Plan |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 52

|  |
| --- |
| Plan hash value: 1445457117  |Id | Operation | Name | Rows | Bytes| Cost (%CPU)| Time |  | 0 | SELECT STATEMENT | | 107 | 7383 | 3 (0 | 00:00:01 |  | 1 | **RESULT CACHE | afz18h8m8pm02td181yj8** | | | | |  | 2 | TABLE ACCESS FULL| EMPLOYEES | 107 | 7383 | 3 (0)| 00:00:01 |  Result Cache Information (identified by operation id):  1 - column-count=11; dependencies=(HR.EMPLOYEES); name="select \* from hr.employees"  Statistics  0 recursive calls  0 db block gets  15 consistent gets  0 physical reads  0 redo size  10121 bytes sent via SQL\*Net to client  628 bytes received via SQL\*Net from client  9 SQL\*Net roundtrips to/from client  0 sorts (memory)  0 sorts (disk)  107 rows processed  SYS@orcl\_3 SQL> |

The result cache entry for this statement has been repopulated. This is apparent from the execution plan as consistent gets has been incremented and is now 15.

1. Again, query GV$\_RESULT\_CACHE\_STATISTICS from the host01 (instance 3) terminal session and re-check results cache statistics for all three instances. What do you see?

|  |
| --- |
| SYS@orcl\_3 SQL> **set autotrace off**  SYS@orcl\_3 SQL> **select inst\_id, name, value from gv$result\_cache\_statistics order by 1,2;**  Hit Any Key to Continue  INST\_ID NAME VALUE    1 Block Count Current 0  Block Count Maximum 3872 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 53

|  |
| --- |
| Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 0   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 32   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  **Create Count Success 2**  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  3 Find Count 1  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 1  **Invalidation Count 1**  Latch (Share) 0  Result Size Maximum (Blocks) 193  45 rows selected. |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 54

|  |
| --- |
| SYS@orcl\_3 SQL> |

Scanning down to instance 3, we can see that Create Count Success has been incremented by 1 and is now 2 as a new results cache has been created, and Invalidation Count is now 1 caused by the committed update to the table.

1. Rerun the EMPLOYEES query and see if the newly cached results will be used.

|  |
| --- |
| SYS@orcl\_3 SQL> **set autotrace traceonly**  SYS@orcl\_3 SQL> **select \* from hr.employees;**  Hit Any Key to Continue  107 rows selected.  Execution Plan  Plan hash value: 1445457117  |Id | Operation | Name | Rows | Bytes| Cost (%CPU)| Time |  | 0 | SELECT STATEMENT| | 107 | 7383 | 3 (0)| 00:00:01 |  | 1 | RESULT CACHE | afz18h8bm8pm02td181yj8n4 | | | | |  | 2 |TABLE ACCESS FULL| EMPLOYEES | 107 | 7383 | 3 (0)| 00:00:01 |  Result Cache Information (identified by operation id):  1 - column-count=11; dependencies=(HR.EMPLOYEES); name="select \* from hr.employees"  Statistics  0 recursive calls  0 db block gets  0 consistent gets  0 physical reads  0 redo size  10121 bytes sent via SQL\*Net to client  628 bytes received via SQL\*Net from client  9 SQL\*Net roundtrips to/from client  0 sorts (memory)  0 sorts (disk) |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 55

|  |
| --- |
| 107 rows processed  SYS@orcl\_3 SQL> |

From the statistics, we can see that no reads or block gets are performed. Looking at the explain plan shows us that the newly cached results were used instead.

1. Again, query GV$\_RESULT\_CACHE\_STATISTICS from the host01 (instance 3) terminal session and re-check results cache statistics for all three instances. What do you see?

|  |
| --- |
| SYS@orcl\_3 SQL> **set autotrace off**  SYS@orcl\_3 SQL> **select inst\_id, name, value from gv$result\_cache\_statistics order by 1,2;**  Hit Any Key to Continue  INST\_ID NAME VALUE     1. Block Count Current 0   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 0   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 56

|  |
| --- |
| Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193  3 Block Count Current 32  Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 2  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 2  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 1  Latch (Share) 0  Result Size Maximum (Blocks) 193  45 rows selected.  SYS@orcl\_3 SQL> |

Looking at Find Count under instance 3, we can see that it has been incremented from 1 to 2, indicating a result cache hit.

1. Next, go to the terminal window for instance 1 and query the EMPLOYEES table. Look at the explain plan. What do you see?

|  |
| --- |
| SYS@orcl\_1 SQL> **set autotrace traceonly**  SYS@orcl\_1 SQL> select **\* from hr.employees;**  107 rows selected.  Execution Plan  Plan hash value: 1445457117  |Id| Operation | Name |Rows |Bytes|Cost (%CPU)| Time |  |0| SELECT STATEMENT | | 107 |7383 | 3 (0)| 00:00:01 |  |1| RESULT CACHE | afz18h8bm8pm02td181yj8udn4 | | | | |  |2| TABLE ACCESS FULL| EMPLOYEES | 107 |7383 | 3 (0)| 00:00:01 | |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 57

|  |
| --- |
| Result Cache Information (identified by operation id):  1 - column-count=11; dependencies=(HR.EMPLOYEES); name="select \* from hr.employees"  Statistics  189 recursive calls  0 db block gets  290 consistent gets  12 physical reads  0 redo size  10121 bytes sent via SQL\*Net to client  628 bytes received via SQL\*Net from client  9 SQL\*Net roundtrips to/from client  9 sorts (memory)  0 sorts (disk)  107 rows processed  SYS@orcl\_1 SQL> |

From the statistics we can see that I/O is being performed and the local result cache entry is built.

1. Again, query GV$\_RESULT\_CACHE\_STATISTICS from the host01 (instance 3) terminal session and re-check results cache statistics for all three instances. You can do this by entering “/” and then pressing Enter. What do you see?

|  |
| --- |
| SYS@orcl\_3 SQL> **/**  Hit Any Key to Continue  INST\_ID NAME VALUE  1 Block Count Current 32  Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 1  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 58

|  |
| --- |
| Find Count 0  Global Hit Count 1  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 0   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 32   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 2  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 2  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 1  Latch (Share) 0  Result Size Maximum (Blocks) 193  45 rows selected.  SYS@orcl\_3 SQL> |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 59

Looking at the statistics for instance 1, we can see that the Global Hit Count has been incremented by 1.

1. Now, update the EMPLOYEES table on instance 1 and commit. When the commit is complete, rerun the EMPLOYEES query. What do you see?

|  |
| --- |
| SYS@orcl\_1 SQL> **set autotrace off**  SYS@orcl\_1 SQL> **update hr.employees set salary = salary + 0;**  107 rows updated.  SYS@orcl\_1 SQL> **commit;**  Commit complete.  SYS@orcl\_1 SQL> **set autotrace traceonly**  SYS@orcl\_1 SQL> **select \* from hr.employees;**  107 rows selected.  Execution Plan  Plan hash value: 1445457117  |Id| Operation | Name |Rows|Bytes |Cost (%CPU)| Time |  |0 | SELECT STATEMENT | |107 |7383 | 3 (0)| 00:00:01 |  |1 | RESULT CACHE | afz18h8bm8pm02td181yj8udn4 | | | | |  |2 | TABLE ACCESS FULL| EMPLOYEES |107 |7383 | 3 (0)| 00:00:01 |  Result Cache Information (identified by operation id):  1 - column-count=11; dependencies=(HR.EMPLOYEES); name="select \* from hr.employees"  Statistics  4 recursive calls  0 db block gets |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 60

|  |
| --- |
| 15 consistent gets  0 physical reads  0 redo size  10121 bytes sent via SQL\*Net to client  628 bytes received via SQL\*Net from client  9 SQL\*Net roundtrips to/from client  0 sorts (memory) |

Looking at the statistics, you can see that there are no db block gets or physical reads and the explain plan shows the local results cache entry for instance 1.

1. Again, query GV$\_RESULT\_CACHE\_STATISTICS from the host01 (instance 3) terminal session and re-check results cache statistics. You can do this by entering “/” and then pressing Enter. What do you see?

|  |
| --- |
| SYS@orcl\_3 SQL> **/**  Hit Any Key to Continue  INST\_ID NAME VALUE   1. Block Count Current 32   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 2  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 1  Global Miss Count 1  Hash Chain Length 1  Invalidation Count 1  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 0   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 0  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 61

|  |
| --- |
| Global Hit Count 0  Global Miss Count 0  Hash Chain Length 0  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193  3 Block Count Current 32  Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 2  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 2  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 2  Latch (Share) 0  Result Size Maximum (Blocks) 193  45 rows selected.  SYS@orcl\_3 SQL> |

Invalidation Count and Create Count Success for instance 1 has been incremented. Invalidation Count for instance 3 has also been incremented. There are two instances caching EMPLOYEES query results.

1. Go to the third instance (orcl\_2 in this example) and query the EMPLOYEES table.

|  |
| --- |
| SYS@orcl\_2 SQL> **set autotrace traceonly**  SYS@orcl\_2 SQL> **select \* from hr.employees;**  107 rows selected.  Execution Plan  Plan hash value: 1445457117  |Id |Operation | Name |Rows |Bytes |Cost (%CPU)| Time | |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 62

|  |
| --- |
| | 0 | SELECT STATEMENT | | 107 | 7383 | 3 (0)| 00:00:01 |  | 1 | RESULT CACHE |afz18h8bm8pm02td181yj8udn4 | | | | |  | 2 | TABLE ACCESS FULL| EMPLOYEES | 107 | 7383 | 3 (0)| 00:00:01 |  Result Cache Information (identified by operation id):  1 - column-count=11; dependencies=(HR.EMPLOYEES); name="select \* from hr.employees"  Statistics  212 recursive calls  0 db block gets  315 consistent gets  17 physical reads  0 redo size  10121 bytes sent via SQL\*Net to client  629 bytes received via SQL\*Net from client  9 SQL\*Net roundtrips to/from client  9 sorts (memory)  0 sorts (disk)  107 rows processed  SYS@orcl\_2 SQL> |

From the explain plan we can see that the results cache entry has been created for instance 2.

1. Query GV$\_RESULT\_CACHE\_STATISTICS from the host01 (instance 3) terminal session and re-check results cache statistics. What do you see?

|  |
| --- |
| SYS@orcl\_3 SQL> **/**  Hit Any Key to Continue  INST\_ID NAME VALUE  1 Block Count Current 32  Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 2  Delete Count Invalid 0 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 63

|  |
| --- |
| Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 1  Global Miss Count 1  Hash Chain Length 1  Invalidation Count 1  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 32   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 1  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 1  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 32   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 2  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 2  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 2  Latch (Share) 0  Result Size Maximum (Blocks) 193  SYS@orcl\_3 SQL> |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 64

Looking at instance 2 statistics we can see that Create Count Success and Global Hit Count have been incremented from 0 to 1.

1. Rerun the EMPLOYEES query on instance 2. What do you observe?

|  |
| --- |
| SYS@orcl\_2 SQL> **/**  107 rows selected.  Execution Plan  Plan hash value: 1445457117  | Id | Operation | Name | Rows | Bytes | Cost (%CPU)| Time |  |0 |SELECT STATEMENT | | 107 | 7383 | 3 (0)| 00:00:01 |  |1 | RESULT CACHE | afz18h8bm8pm02td181yj8udn4 | | | | |  |2 | TABLE ACCESS FULL| EMPLOYEES | 107 | 7383 | 3 (0)| 00:00:01 |  Result Cache Information (identified by operation id):  1 - column-count=11; dependencies=(HR.EMPLOYEES); name="select \* from hr.employees"  Statistics  0 recursive calls  0 db block gets  0 consistent gets  0 physical reads  0 redo size  10121 bytes sent via SQL\*Net to client  629 bytes received via SQL\*Net from client  9 SQL\*Net roundtrips to/from client  0 sorts (memory)  0 sorts (disk)  107 rows processed  SYS@orcl\_2 SQL> |

The explain plan shows that the instance 2 local result cache entry was used. The statistics section confirms this.

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 65

1. Query GV$\_RESULT\_CACHE\_STATISTICS from the host01 (instance 3) terminal session and re-check results cache statistics. What do you see?

|  |
| --- |
| SYS@orcl\_3 SQL> **/**  Hit Any Key to Continue  INST\_ID NAME VALUE   1. Block Count Current 32   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 2  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 0  Global Hit Count 1  Global Miss Count 1  Hash Chain Length 1  Invalidation Count 1  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 32   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 1  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 1  Global Hit Count 1  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 0  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 32   Block Count Maximum 3872  Block Size (Bytes) 1024 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 66

|  |
| --- |
| Create Count Failure 0  Create Count Success 2  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 2  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 2  Latch (Share) 0  Result Size Maximum (Blocks) 193  45 rows selected.  SYS@orcl\_3 SQL> |

Looking at the instance 2 statistics, you can see that the Find Count is now 1. Now, all three instances have results cached for the same table.

1. Next, update the EMPLOYEES table from instance 2 and commit.

|  |
| --- |
| SYS@orcl\_2 SQL> update hr.employees set salary = salary + 0;  107 rows updated.  SYS@orcl\_2 SQL> **commit;**  Commit complete.  SYS@orcl\_2 SQL> |

1. Next, query the result cache stats again. What do you observe?

|  |
| --- |
| SYS@orcl\_3 SQL> **/**  Hit Any Key to Continue  INST\_ID NAME VALUE  1 Block Count Current 32  Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 2  Delete Count Invalid 0  Delete Count Valid 0 |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 67

|  |
| --- |
| Find Copy Count 0  Find Count 0  Global Hit Count 1  Global Miss Count 1  Hash Chain Length 1  Invalidation Count 2  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 32   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 1  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 1  Global Hit Count 1  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 1  Latch (Share) 0  Result Size Maximum (Blocks) 193   1. Block Count Current 32   Block Count Maximum 3872  Block Size (Bytes) 1024  Create Count Failure 0  Create Count Success 2  Delete Count Invalid 0  Delete Count Valid 0  Find Copy Count 0  Find Count 2  Global Hit Count 0  Global Miss Count 0  Hash Chain Length 1  Invalidation Count 2  Latch (Share) 0  Result Size Maximum (Blocks) 193  SYS@orcl\_3 SQL> |

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 68

Note that Invalidation Count has been incremented by 1 for all instances, demonstrating global results cache invalidation coordination handled by the result cache background process, RCBG.

1. To finish, return the EMPLOYEES table result\_cache mode back to DEFAULT.

|  |
| --- |
| SYS@orcl\_3 SQL> **alter table hr.employees result\_cache (mode default);**  SYS@orcl\_3 SQL> |

1. Close all terminals opened for this practice.

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 69

Practices for Lesson 15: Global Resource Management Concepts

Lab 15 - Page 70