Become a Real Applications Clusters Administrator Certified Expert

By Harald van Breederode, Oracle 10g Certified Master
and Joel Goodman, Oracle 10g Certified Master

Concepts and Sample Questions on Oracle Expert Certification in RAC.

The new Oracle Certified Expert program provides opportunities for Oracle Professionals to earn certifications in advanced, niche or specialty technology areas. The Program's first advanced offering for DBAs is the Oracle Database 10g: Administering RAC exam. To earn the Real Application Clusters Administrator Certified Expert credential, one must pass this new exam, and also meet one of the additional certification requirements: one must already have the Oracle Database 10g DBA OCP credential, OR one must attend either the Oracle Database 10*g*: Real Application Clusters or Oracle Database 10*g*: RAC for Administrators Release 2 course.

In this article we will discuss concepts relating to Virtual IP Addresses (VIPs), Parallel Execution, Sequences and Clusterware File Management. We will also provide samples of the type of questions that appear in the test. The sample question format and SQL code have been adjusted for publication in the article.

Sequences and RAC

The use of sequences in an Oracle database is much the same in RAC as it is for single instance non-RAC databases, but there are certain RAC-specific issues relating to functionality and performance. With two or more database instances, sequences may require coordination amongst the instances to guarantee that the sequence numbers are allocated in the correct order.

> Which two are always true regarding the use of sequences in an Oracle 10g: RAC database?
>
> A. It is not possible to use the CACHE option, since each database has a row cache.
> B. Sequence numbers may be out of order if multiple instances run the application program that uses the sequence.
> C. Sequences add no extra overhead to traffic over the Interconnect.
> D. The best performance for a sequence is obtained using CACHE and NOORDER options together.

The correct answers are B and D.

Even though each instance has a row cache it is still possible to use the CACHE option with sequences. In this case each instance may cache numbers for that sequence in its row cache. If the CACHE option is used without the ORDER option then each instance caches a separate range of numbers, and sequence numbers may be assigned out of order by the different instances. If CACHE and ORDER options are used together then all instances must allocate numbers in order by coordinating the assignment of the next value using messages over the interconnect, thereby increasing interconnect traffic in proportion to the frequency of assigning new numbers.

To create a sequence with CACHE and NOORDER:

```
SQL> create sequence rac_cache_noorder
        start with 1 increment by 1
        cache 20 noorder;
```

To create a sequence with CACHE and ORDER:

```
SQL> create sequence rac_cache_order
        start with 1 increment by 1
        cache 20 order;
```

The definition of the sequences is obtained from the view DBA_SEQUENCES:

```
 SQL> select *
        from dba_sequences
        where sequence_name like '%RAC_%';
```

```
SEQUENCE_OWNER  SEQUENCE_NAME          MIN_VALUE  MAX_VALUE   INCREMENT_BY C O  CACHE_SIZE  LAST_NUMBER
--------------- ---------------------  ---------- ----------  ------------ - -  ----------  -----------
SYS             RAC_CACHE_NOORDER               1  1.0000E+27             1 N N          20            1
SYS             RAC_CACHE_ORDER                 1  1.0000E+27             1 N Y          20            1
```

Here both sequences have the same cache size. The last_number column indicates that the last time a change was made to the data dictionary for the sequences, 1 was assigned which means no sequence numbers have yet been allocated.

Application Developers and DBAs must both know how sequences work in a RAC environment so that the behavior of the sequences is understood. Otherwise when converting from single instance to RAC for example, the sequence may not behave in accordance with the application requirements.

The following SQL statements are issued from separate RAC instances on a two-node cluster for a sequence with CACHE and NOORDER.

Instance A:

```
SQL>  Select RAC_CACHE_NOORDER.NEXTVAL from dual;
```

Instance B:

```
SQL>  Select RAC_CACHE_NOORDER.NEXTVAL from dual;
```

What will be the value of LAST_NUMBER in DBA_SEQUENCES after both statements are executed?

A. 21
B. 41
C. 2
D. 20
E. 40

The correct answer is B.

When nextval is issued in instance A, the kernel sees that 1 is stored in the LAST_NUMBER column of DBA_SEQUENCES for this sequence meaning that no numbers for this sequence have yet been assigned and that no values are therefore cached in the row cache for instance A. It also sees that CACHE_SIZE is 20 and ORDER is “N”. The Kernel then returns 1 as the next value for the query and the first 20 numbers are cached in instance A’s row cache with 2 as the next value to be returned from the cache for statements issued from instance A. The LAST_NUMBER column in the Data Dictionary is set to 21 specifying that the next request made for this sequence from any instance should return 21.

When the nextval is issued in instance B, the kernel sees that 21 is stored in the LAST_NUMBER column of DBA_SEQUENCES for this sequence indicating that 20 numbers for this sequence have already been assigned. It also sees that CACHE_SIZE is 20 and ORDER is “N” but that no caching has yet been done for this sequence in instance B’s row cache. The kernel then returns 21 as the next value for the query and the numbers from 21 to 40 are cached in instance B’s row cache with 22 the next value to be returned from the cache for statements issued from instance B.. The LAST_NUMBER column in the Data Dictionary is set to 41 specifying that the next request made for this sequence from any instance should return 41.

Examining the Data Dictionary after the two statements we see the following:

```
SEQUENCE_OWNER SEQUENCE_NAME          MIN_VALUE  MAX_VALUE  INCREMENT_BY C O CACHE_SIZE LAST_NUMBER
-------------- ---------------------- ---------- ---------- ------------ - - ---------- -----------
SYS            RAC_CACHE_NOORDER               1 1.0000E+27            1 N N         20          41
SYS            RAC_CACHE_ORDER                 1 1.0000E+27            1 N Y         20          21
```

To see the values of the row caches use a query like the following;

```
SQL> select inst_id,sequence_name,order_flag,
     nextvalue, cache_size
     from gv$_sequences
     where sequence_name like '%RAC_%'
     order by inst_id,sequence_name;
```

```
INST_ID SEQUENCE_NAME              O  NEXTVALUE  CACHE _SIZE
```

```
---------- ------------------------       -    ---------------- ----------------

    1     RAC_CACHE_NOORDER      N                 2                 20
          RAC_CACHE_ORDER        Y                 2                 20


    2     RAC_CACHE_NOORDER      N                2 2                20
          RAC_CACHE_ORDER        Y                  3                20
```

The gv$_sequences view shows the metadata from the row cache for both instances.

If instance A issues a request for the next value for RAC_CACHE_NOORDER it will get 2. If instance B then issues the same request it will get 22. Here the order of the values returned to the application would be 1,21,2,22. So caching without ordering provides all the benefits of caching for good performance but does not guarantee that the sequence numbers are issued in order.

Notice that the sequence RAC_CACHE_ORDER does not behave the same way and that the LAST_NUMBER column in DBA_SEQUENCES shows 21 rather than 41. Also notice that gv$_sequences indicates the NEXTVALUE column is 3 and not 22. Adding the ORDER option changes the behavior of the Oracle kernel requiring that sequence numbers be returned in order.

> How does Oracle coordinate sequences with CACHE and ORDER options so that numbers are cached in each instance's row cache but are still allocated in the correct order?
>
> A. One instance acts as the mastering instance for the cached values.
> B. Instances regularly send the NEXTVALUE data for all CACHE and ORDER sequences to other instances to guarantee ordering.
> C. The cache information is written and read from the controlfile.
> D. When an instance allocates a new number from a CACHE and ORDER sequence it requests all other instances to pass the NEXTVALUE over the interconnect. The highest value for all instances including the requesting instance is used.

The correct answer is D.

All instances know their own NEXTVALUE based on the last cached value used in that instance. So the NEXTVALUE used for a request from any instance must be the highest NEXTVALUE from any instance. The method used has the lowest possible overhead while guaranteeing sequence numbers are allocated in order.

Parallel Execution and RAC

RAC databases support parallel execution of Queries, DML and DDL much the same way they are supported in a single instance Oracle database, but there are some special considerations regarding performance and administration issues unique to RAC.

> A two-instance RAC database has PARALLEL_MAX_SERVERS = 100 and PARALLEL_MIN_PERCENT = 0 on each instance. The DBA has also set PARALLEL_ADAPTIVE_MULTIUSER to false on both instances. The DBA then logs

in to instance A and attempts to create a large index in parallel using the following statement:

```
SQL> Create index sh.sales_prod_cust on SH.sales (prod_id, cust_id) parallel
10;
```

How are the parallel execution slave processes allocated to build this index?

A. 5 Slaves are allocated from each instance and if either instance has fewer than 5 slaves available then an error is returned.
B. 10 slaves are allocated from instance A and if fewer than 10 are available in instance A then an error is returned.
C. 10 slaves are allocated from instance A and if fewer are available then the creation of the index proceeds with fewer slaves, all on instance A.
D. 10 slaves are allocated from instance A if possible. If fewer are available then slaves are requested from instance B. If instance A and B together can not provide 10 slaves then the statement executes with fewer slaves.
E. 10 slaves are allocated from instance A if possible. If fewer are available then slaves are requested from instance B. If instance A and B together can not provide 10 slaves then the statement returns an error.

The correct answer is D.

Oracle attempts to allocate all slaves on the instance where the coordinator process is running, in this case on instance A. If all slaves required are available then they are allocated from this instance which reduces interconnect overheads. If the coordinator instance is unable to provide enough slaves because some are already allocated or because the parallel request exceeded PARALLEL_MAX_SERVER for the requesting instance, then slaves are requested from other instances. If all the instances together are unable to provide enough slaves then the statement will execute with a reduced set of slaves as PARALLEL_MIN_PERCENT=0, just as would be the case in single instance Oracle.

A related concept in monitoring RAC is the "geometry" of the slave allocation. The view V$PQ_SESSTAT has statistics for parallel operations performed by a session amongst which are the following:

```
SQL> select * from v$pq_sesstat;

STATISTIC              LAST_QUERY SESSION_TOTAL
-------------          ------------------ ----------------------
Queries Parallelized            0              2
DML Parallelized                0              0
DDL Parallelized                1              1
DFO Trees                       1              3
Server Threads                 20              0
Allocation Height              10              0
Allocation Width                2              0
Local Msgs Sent              9216           9507
Distr Msgs Sent              1733           2008
Local Msgs Recv'd            9221           9521
Distr Msgs Recv'd            1748           2036
```

Allocation Height shows the number of slaves used in this instance
Allocation Width shows the number of instances used to execute the statement.
Distr Msgs Sent shows the number of parallel messages passed over the interconnect.
Distr Msgs Recv'd shows the number of parallel messages received over the interconnect.

In addition V$PQ_SYSSTAT has statistics for all instances displaying the total traffic for the instances for messages sent and received.

```
SQL> select * from v$pq_sysstat;

STATISTIC                   VALUE
------------------------    -------------
Servers Busy                            0
Servers Idle                            2
Servers Highwater                      12
Server Sessions                     77595
Servers Started                        59
Servers Shutdown                       57
Servers Cleaned Up                      0
Queries Initiated                   50280
DML Initiated                           0
DDL Initiated                           1
DFO Trees                           73325
Sessions Active                         0
Local Msgs Sent                    356192
Distr Msgs Sent                    244514
Local Msgs Recv'd                  553038
Distr Msgs Recv'd                  247507
```

Another important performance issue related to parallel execution is instance recovery.

A four-node RAC cluster has an instance on each node. Instance C fails due to node failure on node C. All instances have PARALLEL_MAX_SERVER =300. How does the DBA assure recovery for instance C is done in parallel thereby speeding up the recovery?

A. No action is required because PARALLEL_MAX_SERVER is already set on all instances.
B. Issue the recover command for instance C using the parallel option.

C. Make sure that the RECOVERY_PARALLELISM parameter is greater than or equal to 2 to guarantee parallel recovery.
D. No action is required because PARALLEL_MAX_SERVER is set and RECOVERY_PARALLELISM defaults in to 10;

The correct answer is C.

In Oracle 10g RECOVERY_PARALLELISM defaults to CPU_COUNT –1 but the DBA must assure that it a non-zero value and that the degree of parallelism used allows instance recovery to complete within the required service level agreement for recovery. Note that other factors affect recovery speed which are not RAC specific nor related to parallelism such as default buffer cache size, which is outside the scope of this article.

Virtual IP Addresses

High Availability (HA) databases are less than satisfactory if database clients are unable to connect or reconnect quickly when planned or unplanned downtime causes one or more instances to become unavailable. Database clients use TNS descriptors to contact a TNS listener on one of the cluster nodes from a list of listeners on all cluster nodes. TNS descriptors for RAC contain hostnames or IP addresses of the public network interfaces on all cluster nodes.

When database clients attempt connections to cluster databases one of the available hostnames in the TNS descriptor is selected and a connection request is made. If the selected instance or listener on that host is unavailable, clients select another hostname and try again until they succeed in connecting. Although this technique improves availability when instances or listeners are down, a network timeout is required for clients to detect unavailable nodes. Clients are therefore delayed by the TCP timeout period before attempting to connect to another hostname resulting in slower network connection establishment and lower availability of connections.

To circumvent these network timeouts Oracle Database 10g clusters use Virtual IP addresses or VIPs, which respond to connection requests made over the public network interfaces in one of two ways. While a cluster node is available its associated VIP is active on that node, and inbound connection requests are accepted by the listener. If a node becomes unavailable its associated VIP is activated on one of the remaining cluster nodes by the clusterware thereby enabling this other node to reject connection requests originally sent to the failed node. This rejection of connection requests on foreign VIPs results in immediate notification to requesting clients, which immediately select another hostname from the TNS descriptor. This results in faster network connection establishment and higher availability of connections.

Which statements are true about VIPs?

A. VIPs always accept connection requests.
B. VIPs must be manually moved from one node to another.
C. Clients should connect to the VIP instead of the public hostname.
D. VIPs should be resolvable through DNS.
E. VIPs are used to circumvent network timeouts on the cluster interconnect.

The correct answers are C and D.

A is wrong because VIPs only accept connections when the VIP is used on its own node. B is wrong because the Oracle Clusterware will automatically relocate VIPs as required. C is correct as discussed previously. D is correct because clients must resolve IP addresses of VIPs. E is wrong because VIPs are used for client connections made over public network interfaces, not over the interconnect which is a private network for the cluster.

> Your company plans to switch from one Internet Service Provider or ISP to another ISP resulting in new IP addresses for your RAC VIPs. What must you do to implement these new VIP addresses for your RAC cluster?
>
> A. You don't need to do anything. Oracle Clusterware will discover the new VIPS automatically.
> B. You make changes in DNS and Oracle Clusterware will determine the new addresses automatically.
> C. You must plan some scheduled downtime for the whole cluster and make the changes while all cluster software is inactive.
> D. You must stop all VIP dependent cluster components on one node, change the VIP address using SRVCTL and restart all VIP dependent cluster components. Then you repeat the same steps on all cluster nodes one at a time.
> E. You need to re-install your clusterware from scratch.

The correct answer is D.

If A were true we wouldn't need a DBA. B is wrong because the Clusterware stores VIPs in its own metadata to avoid dependency on DNS. C is wrong because planned downtime would compromise the High Availability architecture, which is a major feature of RAC. D is correct. A crucial element of RAC High Availability architecture is support for most maintenance activities in a rolling fashion. This allows software on one node to be shut down and maintenance performed while other nodes continue to operate normally. Upon completion of the maintenance action the node and software are restarted and the same steps are then carried out serially on the remaining nodes. E is wrong for the same reason as answer C.

Clusterware File Management

The two important Oracle Clusterware file types are the Oracle Cluster Registry or OCR and the voting disk. The OCR contains cluster configuration data such as public and private node names, database and instance names, IP and VIP addresses, node applications and voting disk locations. The voting disk is a disk device or file, which plays an important role during cluster reconfiguration activities such as nodes joining or leaving the cluster and public or private network failures. Both the OCR and voting disks can be protected against media failures by multiplexing them, but making backups for disaster recovery is still an important activity for the DBA.

> While evaluating your company's backup and recovery strategy your IT manager asks you to recommend a backup strategy for the Oracle Clusterware voting disk. When would you recommend that backups of the voting disk be done? (Choose all that apply.)
>
> A. Never because there is nothing inside the voting disk.
> B. After adding a node to the cluster.
> C. After removing a node from the cluster.

D. Never because voting disk backups are taken automatically by Oracle Clusterware whenever database backups are taken.

The correct answers are B and C. A is wrong because each node has a heartbeat area in the voting disk. B is correct because a new heartbeat area is created on the voting disk when a new node is added to the cluster. C is correct because a node heartbeat area is removed from the voting disk whenever a node is removed from the cluster. D is wrong because Oracle Clusterware creates OCR backups automatically but does not do so for the voting disk.

Your system administrator informs you of a disk failure in your Oracle RAC cluster, necessitating a replacement disk. An OCR mirror was located on the failed disk and requires recovery, thereby preventing a cluster outage if the remaining OCR were to fail. How would you resolve this issue?

A. Copy the remaining OCR to the mirror location while the cluster is still running.
B. Restore the OCR from a backup location.
C. Copy the remaining OCR to the mirror location while the cluster is shut down.
D. Replace the OCR mirror by issuing a 'ocrconfig –replace' command.
E. Repair the OCR mirror by issuing a 'ocrconfig –repair' command.

The correct answer is D.

A is wrong because the Clusterware prevent mirrors being copied whilst in use. B is wrong because the backup OCR will be out of sync with the remaining OCR and no logs exist to facilitate a roll forward. C is possible but conflicts with RAC High-Availability objectives. D is correct. OCRCONFIG is the command to manage your OCR devices. E is wrong because the –repair argument is used to repair a broken /etc/oracle/ocr.loc file which points to the OCR device location(s).

For Further Information

For further information on the Oracle Certified Expert Program and the Oracle Database 10g RAC Administrator Certified Expert certification, please visit the Oracle Certification Program website at http://www.oracle.com/education/certification. Those considering the RAC Expert exam are advised to consult the exam objectives on the certification website, attend the recommended training from Oracle University, and get hands-on practice with the product before taking the exam.

Questions about Oracle certification can be directed to ocpexam_ww@oracle.com.

Harald van Breederode (ocpexam_ww@oracle.com) is the Linux technical advisor to the Oracle Certification Exam Development team and has worked for Oracle University NL since 1999 after spending 6 years in the Dutch Data Centre in the Netherlands as a Unix Systems Administrator.

Joel Goodman (ocpexam_ww@oracle.com) is the Database technical advisor to the Oracle Certification Exam Development team, and has worked for Oracle University UK since 1997 after spending 2 years in Oracle Support UK.

They are both trainers in the advanced Oracle DBA Curriculum, Oracle Certified Master DBAs, are on the review team for Oracle DBA courseware and deliver courses and seminars in the EMEA region.