Servicio Andaluz de Salud
CONSEJERÍA DE SALUD

Oficina Técnica para la Gestión y Supervisión de Servicios TIC
Subdirección de Tecnologías de la Información

BEST PRACTICES DE SISTEMAS EN ENTORNOS ORACLE REAL APPLICATION CLUSTER

Referencia documento:
InfV5_JASAS_RAC_System_BestPractices_V920.doc
Fecha: 16 de noviembre de 2018
Versión: 9.2.0

Registro de Cambios

Fecha	Autor	Versión	Notas
13 de Enero de 2011	Jonathan Ortiz	2.1.0.	Versión inicial
14 de Abril de 2011	Jonathan Ortiz	2.2.0.	Versión actualizada
14 de Julio de 2011	Jonathan Ortiz	2.3.0.	Versión actualizada
13 de Octubre de 2011	Jonathan Ortiz	2.4.0.	Versión actualizada
17 de Enero de 2012	Jonathan Ortiz	3.1.0.	Versión actualizada
14 de Marzo de 2013	Jonathan Ortiz	4.1.0.	Versión actualizada
14 de Junio de 2013	Jonathan Ortiz	4.2.0.	Versión actualizada
17 de Octubre de 2013	Jonathan Ortiz	4.3.0.	Versión actualizada
16 de Julio de 2015	Enrique Ramiro	6.1.0	Revisión de Julio de 2015
16 de Diciembre de 2015	Enrique Ramiro	6.2.0	Revisión de Diciembre de 2015
16 de Junio de 2016	Enrique Ramiro	7.1.0	Revisión de Junio de 2016
16 de Noviembre de 2016	Enrique Ramiro	7.2.0	Revisión de Noviembre de 2016
16 de Junio de 2017	Enrique Ramiro	8.1.0	Revisión de Junio de 2017
16 de Noviembre de 2017	Enrique Ramiro	8.2.0	Revisión de Noviembre de 2017
16 de Junio de 2.018	Enrique Ramiro	9.1.0	Revisión de Junio de 2018, contrato 2016-2018
16 de Noviembre de 2018	Enrique Ramiro	9.2.0	Revisión de Noviembre de 2018

Revisiones

Nombre	Role
Jonathan Ortiz	Advanced Support Engineer
Gregorio Adame	Advanced Support Engineer
José María Gómez	Technical Account Manager

Distribución

Copia	Nombre	Empresa
1	Subdirección de Tecnologías de la Información	Servicio Andaluz de Salud, Junta de Andalucía
2	Dirección General de Política Digital	Consejería de Hacienda y Administración Pública, Junta de Andalucía

Índice de Contenidos

Control de cambios

Cambio	Descripción	Página
1	No se realizan cambios en esta versión	N/A

Introducción

Este documento recoge una serie de recomendaciones de Oracle Soporte planteadas como buenas prácticas de sistemas para administradores que hagan uso de *Oracle RDBMS* y *Oracle RDBMS 12cR2 Real Application Cluster (RAC)*.

Estas recomendaciones están encaminadas a minimizar los posibles problemas de configuración y rendimiento en sistemas de cualquier tamaño y en la gran mayoría de los casos se basan en la experiencia de cosas reales gestionados por Oracle Soporte.

Finalmente, este documento también recoge una serie de conceptos de componentes, módulos y tecnologías relacionadas con *Oracle RDBMS* y *Oracle RDBMS 12cR2 RAC*, que a juicio de Oracle Soporte, deberían tenerse claros para asegurar la aplicación de las recomendaciones recogidas en este documento, y de manera general, entender los productos *Oracle RDBMS* y *Oracle RDBMS 12cR2 RAC* sobre los que se sostengan los sistemas y aplicaciones.

Objetivos de este documento

A lo largo de los puntos de este documento se irá definiendo una guía de buenas prácticas para la administración de bases de datos en clúster a través de *Oracle RDBMS Real Application Cluster* (RAC). Esta guía contendrá tanto prácticas recomendadas como prácticas a evitar y se apoyará en ejemplos y en información que permita analizar las recomendaciones en cada uno de los entornos de desarrollo y preproducción.

Este documento se centra principalmente en las versión *Oracle RDBMS Real Application Cluster* 12cR2, pero siguen estando recogidas las prácticas para las versiones anteriores, hasta 10gR2.

El objetivo de esta guía de buenas prácticas tiene varios objetivos:

- Aprovechamiento de las características del producto Oracle RDBMS Real Application Cluster.
    - Escalabilidad
    - Alta disponibilidad
    - Balanceo de carga
- Facilitar la gestión y administración y de las bases de datos en producción con sistemas basados en Oracle RDBMS Real Application Cluster.

RECOMENDACIONES DE ADMINISTRACIÓN Y CONFIGURACIÓN DE ENTORNOS RAC

Recomendaciones generales son recomendaciones que su aplicación pueden tener un mayor impacto en un entorno RAC, o responder a las cuestiones o problemas que ocurren con más frecuencia en RAC. En este caso, las recomendaciones genéricas hacen referencia a las cuestiones más frecuentes propias de las implementaciones de RAC en cualquier plataforma.

Recomendaciones genéricas de la configuración

- Tener un plan paso a paso para la implementación de su proyecto de RAC es invaluable. El artículo OTN siguiente contiene un esbozo del proyecto de ejemplo: http://www.oracle.com/technetwork/articles/haskins-rac-project-guide-099429.html

- Para simplificar la pila y simplificar las interacciones de proveedores, Oracle recomienda evitar el uso de un clúster de terceros, a menos que sea absolutamente necesario.

- Automatic Storage Management (ASM) se recomienda para el almacenamiento tanto de los ficheros de Clusterware (OCR y Voting Disks) como de los ficheros de base de datos.

- A partir de Oracle Clusterware 12cR2, los ficheros de Clusterware (OCR y voting disk) sólo se pueden almacenar en Oracle ASM.

- Tener un plan de pruebas del sistema para ayudarle a planificar y practicar cortes imprevistos es crucial. El Oracle Maximum Availability Architecture (MAA) de Oracle Database puede ayudar en este sentido.

- Desarrollar una estrategia proactiva de parches, para mantenerse a la vanguardia de los últimos problemas conocidos. Manténgase actualizado con las últimas actualizaciones Patch Set (tal como se documenta en el documento 1360790.1) y estar al tanto de los parches más recientes recomendados (como se documenta en el documento 756671.1). Aunque se detalla en el apartado de recomendaciones de aplicación de Parches en RAC.

Herramientas de Administración en Oracle Real Application Clusters

Como norma no se especifica ninguna herramienta de administración obligatoria, pero si hace referencia a las herramientas proporcionadas por Oracle para la administración.

Como herramienta previa a la administración para revisar un entorno RAC, Oracle proporciona la herramienta conocida como RACcheck.

RACcheck es una herramienta de auditoría sobre la configuración en RAC, diseñada para auditar diversas opciones de configuración importantes dentro Real Application Clusters (RAC), Oracle Clusterware (CRS), Automatic Storage Management (ASM) y entornos Grid Infrastructure.

Esta utilidad se va a utilizar para validar las Best Practices y configuraciones recomendadas, por ello se recomienda encarecidamente utilizar esta herramienta identifica posibles problemas de configuración que pueden afectar la estabilidad del cluster.

Esta herramienta es actualizada con frecuencia por lo que es recomendable visitar la nota de documentación Note#1268927.1 RACcheck - RAC Configuration Audit Tool, donde Oracle realiza las actualizaciones de la herramienta.

La siguiente sección introduce a la administración de Oracle RAC haciendo uso de las siguientes herramientas:

- OracleEnterprise Manager
- SQL*Plus
- SRVCTL command

Oracle Enterprise Manager Grid/Cloud Control proporciona un punto de control centralizado para la administración de un entorno Oracle RAC, permitiendo ejecutar tareas administrativas de forma simultánea. Proporciona un entorno de interfaz gráfica muy amigable para el administrador y de fácil manejo. Permite el control si las tareas se realizaran de forma que afecte a todo el clúster completo de base de datos y tareas específicas a una instancia del clúster.

En 12cR1, Oracle Enterprise Manager Database Express reemplaza a Oracle Enterprise Manager Database Control (DB Control). EM DB Control ya no está disponible en Oracle Database 12c, hay que usar Enterprise Manager Cloud Control 12c/13c o EM Express 12c para administrar las bases de datos 12c.

SQL*Plus ya conocido por los administradores de bases de datos en entornos single instance es la interfaz de comando para operar con las bases de datos Oracle. Es una herramienta muy potente y conocida por los administradores, por lo que no vamos a entrar en detalles de su uso, si reseñar que por defecto SQL*Plus no identifica la instancia actual sobre la que estamos trabajando. Por lo que si hacemos uso de ella y para evitar confusiones se recomienda la modificación del prompt a través del siguiente comando

- `SET SQLPROMPT '_CONNECT_IDENTIFIER>'`

Con esté comando conseguimos identificar sobre que instancia del cluster estamos trabajando.

Por último, tener en cuenta que desde SQL*Plus se permiten ciertos comandos que afectan a la instancia en la que estamos conectado o que afectan a la base de datos complete.

SRVCTL (Server Control) Tool: Esta herramienta de administración de Oracle Real Application Clusters proporciona una interfaz de línea de comando que puede ser usada para gestionar bases de datos en Oracle RAC desde un punto único.

Para gestionar Oracle Database RAC o Single Instance, hay que usar el binario de SRVCTL del Oracle Database home.

Para gestionar Oracle ASM 11gR2/12cR1/12cR2, hay que usar el binario de SRVCTL del Oracle Grid Infrastructure home for a cluster (Grid home)

Se puede hacer uso de ella para arrancar y parar instancias o base de datos, borrar o mover instancias o servicios.

Como norma se recomienda el uso de SRVCTL para la administración de paradas y arranques de bases de datos en Oracle RAC.

CRSCTL (Oracle Clusterware Control): No usar comandos CRSCTL en Oracle entities (como son los resources, resource types y server pools) cuyos nombres comienzan con ora*, a menos que se lo haya indicado explícitamente My Oracle Support. La utilidad SRVCTL es la utilidad correcta para gestionar las Oracle entities.

Desde Clusterware 12cR2, ya no pueden usar los comandos de Oracle Clusterware que tienen el prefijo crs_. Se han des-soportado y eliminado.

Arranque y paradas de instancias y Oracle Real Application Clusters

El procedimiento de parar instancias en Oracle RAC es idéntico al parar instancias en single-instance.

En Oracle RAC, la parada de instancias no interfiere en las operaciones de las otras instancias, pero si puede provocar bajo situaciones de alta carga una sobrecarga por lo que se debe tener en cuenta para realizar las paradas de forma controlada y en horas valle de trabajo.

Para parar una base de datos Oracle RAC completamente es necesario realizar el shutdown de todas las instancias.

Se asume que la parada y arranque de las bases de datos Oracle desde SQL*Plus y Enterprise manager es conocido por los administradores de base de datos por lo que se hará referencia al método recomendado de parada desde la herramienta SRVCTL:

Arranque y Parada con SRVCTL

Para llevar a cabo la parada o arranque de una base de datos desde la command line SRVCTL, es necesario proporcionar el nombre de la base de datos y el nombre de la instancia o incluir más de un nombre de instancia para por ejemplo arrancar mas de una, o sólo indicar la base de datos sin instancia para parar la base de datos completa con todas sus instancias; como por ejemplo:

```
$ srvctl start instance -d db_name -i "inst_name_list" [-o
start_options]
```

Nota, que en el comando anterior habilita todas las instancias de la lista que no estén en ejecución si alguna de ellas ya se encontraba arrancada no realizar ninguna operación sobre ella.

```
$ srvctl start database -d db_name [-o start_options]
```

El comando anterior iniciaría la base de datos db_name completa, con todas sus instancias. A partir de 12c, el parámetro "-d" también se puede especificar con "-db"

Para llevar a cabo la parada de las instancias o base de datos se ejecutan los siguientes comandos:

```
$ srvctl stop instance -d db_name -i "inst_name_list" [-o
stop_options]
```

Nota que este comando parara los servicios relacionados a las instancias que se paren en los nodos donde las instancias se encuentren ejecutándose.

```
$ srvctl stop database -d db_name [-o stop_options]
```

El comando anterior pararía la base de datos db_name completa, con todas sus instancias.

Para verificar el estado de la base de datos:

```
$ srvctl status database -d db_name
```

Para verificar que las instancias se encuentran en ejecución también se puede realizar la siguiente consulta sobre la vista que se muestra a continuación:

```
SQL> SELECT * FROM V$ACTIVE_INSTANCES;

Esta consulta devolverá una salida similar a la siguiente:

INST_NUMBER INST_NAME
----------- ------------------
1 db1-sun:db1
2 db2-sun:db2
3 db3-sun:db3
```

Parada de sesiones en una instancia específica del cluster

Es conocido por los administradores de base de datos el uso del comando:

- *ALTER SYSTEM KILL SESSION*

Para realizar la terminación de una sesión. Este mismo comando se encuentra disponible para entornos Oracle RAC, pero teniendo en cuenta una sintaxis especial que se recomienda su uso en RAC.

Para terminar sesiones en una instancia específica del clúster es necesario realizar los siguientes pasos:

1. Determinar el valor de INST_ID columna disponible en la gv$session para identificar la instancia en la que se encuentra la sesión.

2. Ejecutar el comando especificando como parámetros el Session ID, Serial Number y en entornos RAC el Instance ID, con la siguiente sintaxis:

SQL> ALTER SYSTEM KILL SESSION 'SID,SERIAL,@INSTANCE_ID'

Consideraciones del fichero de parámetros en Oracle Real Application Clusters

Al crear la base de datos, Oracle crea un SPFILE en la ubicación del archivo que se especifique. Esta ubicación puede ser en:

- Oracle Automatic Storage Management (Oracle ASM). Oracle recomienda esta solución sobre las demás.
- Un sistema de ficheros en cluster:
  - En cuyo caso Oracle recomienda Oracle Automatic Storage Management Cluster File System (Oracle ACFS).
  - Un sistema de ficheros en cluster de terceros certificado para Oracle RAC. Ejemplos:
    - Oracle OCFS2 (Linux, only)
    - IBM GPFS (IBM AIX, only)
- Una solución certificada de network file system (NFS)

Si se crea manualmente la base de datos, entonces Oracle recomienda que se cree un SPFILE de un archivo de parámetros de inicialización (PFILE).

Todas las instancias de la base de datos de clúster utilizan el mismo SPFILE en el arranque.

En Oracle RAC se recomienda que se utilice el archivo SPFILE para simplificar la administración y mantener la coherencia de ajuste de parámetros.

Además, permite configurar RMAN para realizar copias de seguridad del SPFILE.

Modificación de los valores de un fichero de parámetros SPFILE para Oracle Real Application Clusters

Para la modificación del fichero de parámetros SPFILE, se recoge como norma general el uso del comando:

Sql> alter system set

Como norma, no se permitirán el uso de parámetros con valores específicos para una única instancia.

En el caso de los parámetros específicos de una instancia en Oracle RAC se seguirá la siguiente nomenclatura

**.OPEN_CURSORS=500 (Parámetros genéricos de toda la base de datos)*

instancia1.UNDO_TABLESPACE=UNDOTS1 (Parámetros específicos de una instancia)

En el caso de que el DBA necesite ejecutar una actualización de un parámetro en el SPFILE de una instancia en Oracle RAC o que aplique a todas las instancias del clúster se realizara siguiendo las siguientes sentencias:

- Ejecución a nivel de toda la base de datos:

ALTER SYSTEM SET OPEN_CURSORS=1500 sid='' SCOPE=SPFILE;*

- Ejecución a nivel de una única instancia:

ALTER SYSTEM SET OPEN_CURSORS=1500 SID='prod1' SCOPE=SPFILE;

Uso de parámetros en n Real Application Clusters

Por defecto, la mayoría de los parámetros son configurados a sus valores por defecto y tendrán el mismo valor a lo largo de todas las instancias. Sin embargo hay ciertos parámetros que pueden tener diferentes valores en las diferentes instancias y parámetros que pueden tener diferentes valores para ello vamos a mostrar las siguientes tablas

Parámetros específicos de base de datos en RAC
ACTIVE_INSTANCE_COUNT
ASM_PREFERRED_READ_FAILURE_GROUPS
CLUSTER_DATABASE
CLUSTER_DATABASE_INSTANCES
CLUSTER_INTERCONNECTS
DB_NAME
DISPATCHERS
GCS_SERVER_PROCESSES
INSTANCE_NAME
RESULT_CACHE_MAX_SIZE
SERVICE_NAMES
SPFILE
THREAD

Parámetros que DEBEN ser idénticos para todas las instancias en base de datos en RAC
COMPATIBLE
CLUSTER_DATABASE
CONTROL_FILES
DB_BLOCK_SIZE
DB_DOMAIN
DB_FILES
DB_NAME
DB_RECOVERY_FILE_DEST
DB_RECOVERY_FILE_DEST_SIZE
DB_UNIQUE_NAME
INSTANCE_TYPE (RDBMS or ASM)
PARALLEL_EXECUTION_MESSAGE_SIZE
REMOTE_LOGIN_PASSWORDFILE
UNDO_MANAGEMENT
DML_LOCKS (sólo si el valor es 0)
RESULT_CACHE_MAX_SIZE (sólo si el valor es 0)

Parámetros que se recomienda que tengan el mismo valor para todas las instancias en base de datos en RAC
ARCHIVE_LAG_TARGET
CLUSTER_DATABASE_INSTANCES
LICENSE_MAX_USERS
LOG_ARCHIVE_FORMAT
SPFILE
TRACE_ENABLED
UNDO_RETENTION

Parámetros que son únicos para cada una de las instancias en base de datos en RAC
INSTANCE_NUMBER
INSTANCE_NAME
ASM_PREFERRED_READ_FAILURE_GROUPS
UNDO_TABLESPACE
ROLLBACK_SEGMENTS (si se indica)

Administración de Jobs en Oracle Real Application Clusters

Se puede administrar Oracle Jobs, tanto a nivel de la base de datos como a nivel de una instancia. Por ejemplo, se puede crear un job a nivel de base de datos del clúster y el trabajo se ejecutará en cualquier instancia activa de la base de datos Oracle RAC. O bien, puede crear un trabajo a nivel de instancia y el trabajo se ejecutará en la instancia concreta en la que se creó. En el caso de un fallo, los trabajos se pueden repetir su ejecución en otra instancia que se encuentre activa.

Dbms_scheduler frente Dbms_job

A partir de Oracle 10gR1, aparece Unified Scheduler (ahora llamado Oracle Scheduler), implementado en el paquete DBMS_SCHEDULER como sustituto de DBMS_JOB para la gestión y ejecución de tareas programadas de replicación, AQ y Streams.

Ahora en 12cR2 el paquete DBMS_JOB ha sido deprecado y será des-soportado en una release futura; por lo que hay que cambiarse a DBMS_SCHEDULER.

Algunas de las ventajas de DBMS_SCHEDULER frente a DBMS_JOB en entornos RAC son las siguientes:

- Los trabajos o tareas son asociadas a un servicio, si la instancia asociada con el servicio fall, el trabajo acompaña al servicio hasta su instancia de backup.
- Cada instancia de RAC tiene su propio coordinador de trabajos que ejecutan los asociados al servicio que soportan.
- Las estadísticas de los servicios combinadas con funcionalidades de DBMS_SCHEDULER permite la gestión eficiente de los trabajos de replicación en entornos RAC.

Dbms_job

DBMS_JOB es un paquete que suele usarse con bastante frecuencia en entornos donde existe replicación de datos con algunas de las opciones incluidas en Oracle Advanced Replication.

Si las tablas replicadas son exclusivamente accedidas desde uno de las instancias, tiene sentido usar la funcionalidad de afinidad a una instancia implementada en el paquete DBMS_JOB. Esto se consigue al especificar el número de la instancia a las operaciones de gestión de los trabajos. Los procedimientos que hacen uso del parámetro INSTANCE son los siguientes:

- DBMS_JOB.SUBMIT
- DBMS_JOB.CHANGE
- DBMS_JOB.INSTANCE
- DBMS_JOB.USER_EXPORT

Adicionalmente, la asignación de una instancia de un trabajo, puede cambiarse en cualquier momento a través del procedimiento DBMS_JOB.INSTANCE.

Estas recomendaciones no solo aplican en entornos de replicación, ya que por regla general cualquier tipo de trabajo se ve beneficiado al trabajar en un única instancia.

RECOMENDACIONES DE CLUSTERWARE

Más conocido como CRS, Oracle Clusterware es un software de cluster portable que fue integrado como una nueva funcionalidad en Oracle RAC 10gR1 y que proporciona una interfaz de clúster y permite operaciones de alta disponibilidad no disponibles en versiones anteriores.

En entornos Oracle RAC, Oracle Clusterware gestiona todos los recursos de forma automática

Desde Oracle 10gR1, el único software de cluster que se necesita para Oracle RAC es Oracle Clusterware.

Desde 10gR2 Oracle Clusterware es requerido para Oracle RAC. Proporciona la infraestructura necesaria para Oracle Real Application Clusters (RAC).

Puede funcionar sobre un clúster de terceros (Veritas Cluster, HP ServiceGuard), pero no lo necesita.

Recomendaciones generales de la configuración

Recomendaciones generales de la instalación del software de Clusterware junto con el software Oracle Rdbms:

- Desde 10gR2 hasta 11gR1, Oracle Clusterware viene en el instalador de Oracle Clusterware. A partir de 11gR2, Oracle Clusterware y Oracle ASM vienen en el instalador de software de Grid Infrastructure
- Con Oracle Grid Infrastructure 11gR2/12cR1/12cR2 se instalan en el mismo directorio "Grid Infrastructure home":
  - Oracle Automatic Storage Management (Oracle ASM)
  - Oracle Clusterware
- La instalación the los dos productos de forma combinada se denomina Oracle Grid Infrastructure; sin embargo, Oracle Clusterware y Oracle ASM permanecen como productos separados.
- A la hora de instalación, el GI_HOME (que es también el CRS_HOME) y el ORACLE_HOME tienen que ser distintos.
- Como norma, se recomienda el uso de usuarios independientes de sistema operativo para la administración del software de Grid Infrastructure y el software de Oracle Rdbms, esta norma puede ser modificada bajo ciertas circunstancias al uso de un usuario común para CRS y RDBMS para ello consultar con Oracle Soporte.
- Aunque desde 10gR2 en adelante puede ser utilizado para gestionar aplicaciones de 3ºs en un escenario de failover, sin RAC/Oracle implicados,

no se recomienda de forma genérica. Para alguna excepción consultar con Oracle Soporte.

- Clusterware gestiona los recursos, tales como las Virtual IP (VIP), bases de datos, listeners, servicios y así sucesivamente. Estos recursos se nombran generalmente ora.host_name.resource_name. Oracle no soporta la edición de estos recursos, excepto bajo la dirección explícita de My Oracle Support.

- En entornos Oracle RAC, Oracle Clusterware gestiona todos los recursos de forma automática

- CRS puede centralizar la administración de los recursos en ejecución a través de todos los nodos del clúster.

- Mientras RAC es dependiente de CRS, CRS NO depende de RAC, es un producto independiente y se puede usar para gestionar bases de datos single-instances Cold Cluster Failoveren o en modo standalone (Oracle Restart).

- A partir de Oracle Clusterware 12cR2, los ficheros de Clusterware (OCR y voting disk) sólo se pueden almacenar en Oracle ASM.

- En Oracle Clusterware 12cR2, todos los clusters están configurados como Oracle Flex Clusters, lo que significa que un clúster está configurado con uno o más Nodos Hub y puede admitir una gran cantidad de Nodos Leaf

- Los clústeres configurados actualmente en versiones anteriores de Oracle Clusterware se convierten como parte del proceso de actualización a 12cR2, incluida la activación de Oracle Flex ASM (que es un requisito para Oracle Flex Clusters).

Pila (stack) de Oracle Clusterware

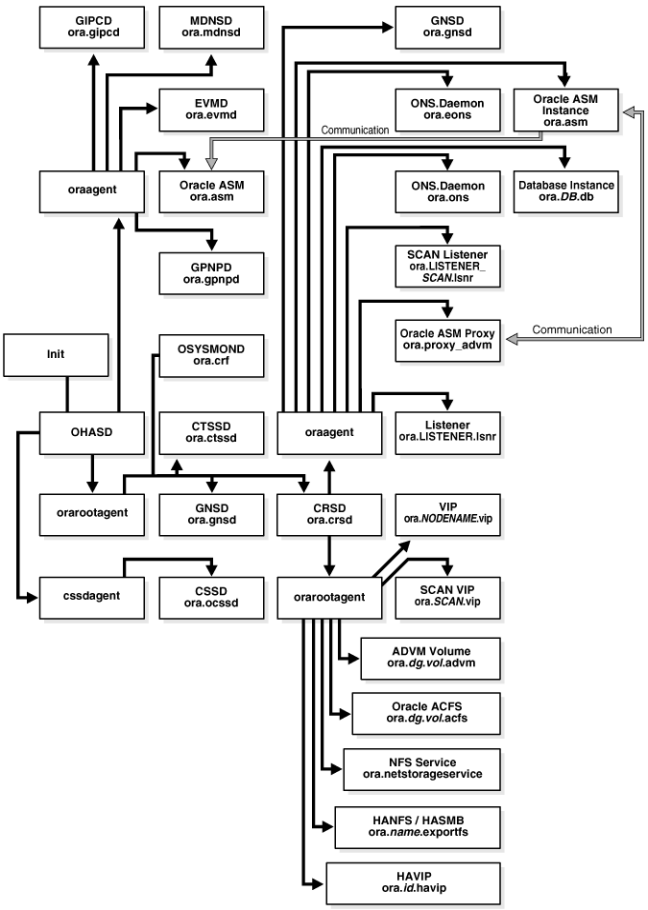
Esta es la pila (stack) de Oracle Clusterware:

- El stack de Cluster Ready Services
    - Cluster Ready Services (CRS) daemon (crsd)
    - Cluster Synchronization Services (CSS)
    - Oracle ASM
    - Cluster Time Synchronization Service (CTSS)
    - Event Management (EVM)
    - Oracle Notification Service (ONS)
    - Oracle Agent (oraagent)
    - Oracle Root Agent (orarootagent)
    - Los componentes Cluster Synchronization Service (CSS), Event Management (EVM) y Oracle Notification Services (ONS) se comunican con otras capas de componentes de cluster en otros nodos del mismo

entorno de cluster de base de datos. Estos componentes son también los principales enlaces de comunicación la base de datos Oracle, las aplicaciones y los componentes HA de Oracle Clusterware. Además, estos procesos de background monitorizan y gestionan las operaciones de base de datos

- El stack de Oracle High Availability Services
  - Oracle High Availability Services daemon (ohasd)
  - Cluster Logger Service (ologgerd)
  - System Monitor Service (osysmond)
  - Grid Plug and Play (GPNPD)
  - Grid Interprocess Communication (GIPC)
  - Multicast Domain Name Service (mDNS)
  - Oracle Grid Naming Service (GNS)
  - Appagent
  - Oracle Agent (oraagent)
  - Oracle Root Agent (orarootagent)
  - scriptagent

Y la siguiente imagen ilustra el Startup de todos los procesos de la pila del cluster:

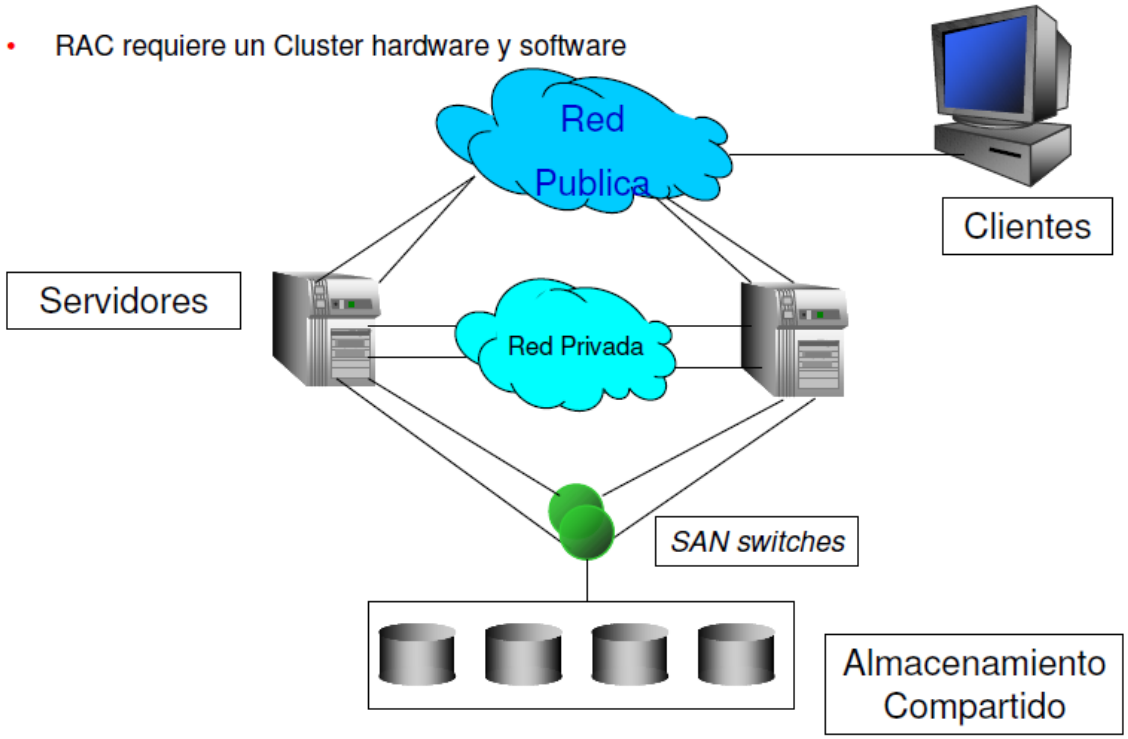


Arquitectura de Clusterware

Clusterware o CRS está formado por los siguientes componentes:

- Voting disk, necesario para los servicios de sincronización del cluster. Usado como "Quorum" en caso de fallo. Proporciona un mecanismo de "heartbeat" para comprobar el estado del clúster.
- Oracle Configuration Repository (OCR "oracle cluster registry"), mantiene información sobre la "alta disponibilidad" de los recursos del clúster: lista de nodos del clúster, accesibilidad de los nodos a la base de datos.
- Los procesos internos del stack que hemos visto en el sub-apartado anterior.
- Clusterware proporciona que los siguientes recursos estén en alta disponibilidad:
  - Node ó Local VIP (Virtual IP)
  - SCAN VIP
  - ONS (Oracle Notification Services)
  - Local Listeners
  - SCAN Listeners
  - Instancias de base de datos
  - Instancias de ASM

Gráfica de Arquitectura básica de RAC:



Certificación de CRS/RAC

Todos los componentes del clúster deben formar parte de una configuración certificada. ***Nota#337737.1***

La siguiente tabla muestra la matriz de compatibilidad entre diferentes versiones de Clusterware/ASM/BasedeDatos:

Clusterware	ASM	Database	Certificado
12.2	12.2	12.2	Si
12.2	12.2	12.1	Si
12.2	12.2	11.2(a)	Si
12.1	12.1	12.1	Si
12.1	12.1	11.2(a)	Si
12.1	12.1	11.1(a)	Si
12.1	12.1	10.2(a)	Si
11.2	11.2(b)	11.2	Si
11.2	11.2(b)	11.1	Si
11.2	11.2(b)	10.2	Si
11.1	11.1	11.1(c)	Si
11.1	11.1	10.2	Si
11.1	11.1	10.1	Si
11.1	10.2	11.1	Si
11.1	10.2	10.2	Si
11.1	10.2	10.1	Si
11.1	10.1(d)	11.1	Si
11.1	10.1(d)	10.2	Si
11.1	10.1	10.1	Si
10.2	10.2	10.2	Si
10.2	10.2	10.1	Si
10.2	10.1(d)	10.2	Si
10.2	10.1	10.1	Si
10.1	10.1	10.1	Si

Notas:

(a) Instancias de base de datos Pre-12.1 requieren de una instancia de ASM residente en el mismo nodo de cada instancia de base de datos. Las instancias de base de datos anteriores a la 12.1 no pueden aprovechar el HA implícito de Flex ASM.

(b) La matriz es válida después de que el rolling upgrade se haya completado. Durante el upgrade puedes utilizar una versión de ASM anterior.

(c) Para entornos Linux revisar la Nota:781628.1

(d) La versión de ASM debe ser por lo menos 10.1.0.3

1. La versión de Grid Infrastructure(GI)/Oracle Clusterware (CRS) debe ser la versión más alta hasta el cuarto dígito en las posibles combinaciones.

2. Grid Infrastructure/Clusterware debe estar instalado en su propio ORACLE_HOME (normalmente conocido como GI_HOME o CRS_HOME)

3. Se puede tener múltiples ORACLE_HOMEs para las versiones de base de datos.

4. En Oracle Database 10gR2 y 11gR1, ASM y base de datos se pueden instalar en ORACLE_HOME separados. A partir de 11gR2 ASM es parte de Grid Infrastructure

5. Si se mezclan versiones de software, la funcionalidad de ASM revierte a la funcionalidad de la versión más antigua en uso. Por ejemplo, una instancia RDBMS 10.1.0.3 que trabaja con una instancia ASM 10.2 no aprovecharía las nuevas características de ASM 10.2. Por el contrario, la instancia 10.2 RDBMS que trabaja con una instancia ASM 10.1 no aprovechará ninguna de las nuevas características 10.2 de ASM.

6. Se pueden (y deben) utilizar diferentes usuarios para los homes de Oracle Grid Infrastructure/Clusterware y RDBMS, siempre y cuando pertenezcan al mismo grupo primario.

7. Los usuarios de base de datos no necesitan tener acceso al OCR y voting disk.

Todos los componentes del clúster deben formar parte de una configuración certificada.

La certificación de los componentes que forman parte del clúster hay que tener en cuenta las siguientes recomendaciones:

- Los protocolos usados por el Cluster Interconnect son los siguientes:
    - Chequear los protocolos soportados (IP/UDP/RDS)
    - CRS usa TCP por defecto
    - RAC usa UDP por defecto

¿Es posible deshabilitar Oracle Clusterware y seguir ejecutando RAC? No, Deshabilitar Oracle Clusterware en un entorno RAC no está soportado por Oracle.

- ¿Podemos usar Oracle Clusterware sin nodeapps? No. Nodeapps (Node Applications) son requeridos como parte de la pila de Oracle Clusterware.
    - Nodeapps está formado por los siguientes recursos:

- VIP (Virtual IP)
- ONS (oracle notification service)
- GSD (global services daemon)
- Tns Listeners

- ¿Cuántos números de nodos puede tener un clúster en HPUX/Solaris/AIX/Windows/Linux?
  - Cuando usamos solo Oracle Clusterware: 63 nodos en 9i ó 10gR1, a partir de 10g Release 2, el número máximo de nodos es 100.
  - Cuando usamos un clusterware de terceros:
    - Sun: 8
    - HP UX: 16
    - HP Tru64: 8
    - IBM AIX:
      - 8 nodos para Physical Shared (CLVM) SSA disk
      - 16 nodos para Physical Shared (CLVM) non-SSA disk
      - 128 nodos para Virtual Shared Disk (VSD)
      - 128 nodos para GPFS
    - Veritas: 8-16 nodos (chequear con Veritas)
- Otras consideraciones de elementos que están soportados con Clusterware son:
  - Ethernet
    - 1GbE, y/o 10GbE y/o Infiniband en NIC's y Switches
    - Los siguientes NFS file servers están soportados
      - EMC Celerra, Fujitsu Filer NR1000 Series, HP Enterprise File
      - Services Clustered Gateway, IBM N Series
      - NetApp FAS, F, G Series
      - Pillar Data Systems Axiom 500, Sun StorageTek 5000 Series
- Consideraciones de elementos que **NO** están soportados con Clustwerare son:
  - Tanto la opción Real Application Cluster como Grid Infrastructure/Oracle Clusterware no soportan plataformas heterogéneas dentro del mismo cluster, es decir, no es posible que un nodo del cluster sea SPARC 64bits y el resto Itanium, o que un nodo sea x86 32 bits y el resto x86 64bits.

- Sin embargo, ambos productos soportan máquinas de diferentes velocidades y tamaños dentro de la misma plataforma y con el mismo sistema operativo o que al menos exista compatibilidad entre binarios, pero no se recomienda.
- Cable cruzado para interconectar 2 nodos de un cluster.
- Compartir el private interconnect con otra aplicación.
- A partir de Oracle Database/GI/Clusterware 12cR1 no se soportan sistemas de 32-bits.
- Desde 12cR1 no es posible almacenar las bases de datos y tecnologías relacionadas como Clusterware, en dispositivos de almacenamiento raw ni block.
- Desde 12cR2, no es posible almacenar los ficheros de Clusterware (OCR y voting disk) en ningún otro almacenamiento que no sea ASM.
- El balanceo de carga en la red Pública a través de múltiples NICs debe ser configurado a nivel de S.O.
- Diferentes arquitecturas en nodos del mismo clúster.
- Desde Clusterware 12cR2, ya no pueden usar los comandos de Oracle Clusterware que tienen el prefijo crs_. Se han des-soportado y eliminado.

Componentes principales de Clusterware

Interconnect: Se conoce como el componente que se utiliza para la intercomunicación entre los nodos del clúster.

- CRS y RAC necesitan de los siguientes interfaces de red:
  - Public Interface
  - Private Interface (Interconnect)
  - Virtual Interface (Public)
- El Interface privado o Interconnect es una red privada que comunica todos los nodos que forman parte del clúster.
- El envió de "heartbeat" es realizado a través de la red privada (interconnect)
- Soporta la comunicación entre nodos para RAC Global Cache y Global Enqueue de recursos
- Baja latencia es más importante que un gran ancho de banda.

- Compartida por todos los nodos del clúster.
- Se recomienda red "Gigabit Ethernet" en adelante (10GbE y/o Infiniband)
- Se recomienda la utilización del protocolo UDP.
- Se recomienda configuración redundante del interconnect con interfaces privadas redundantes y múltiples network switches.
- Si la red de Interconnect falla sólo un nodo sobrevive en el clúster

OCR - Oracle Cluster Registry: Es el repositorio que contiene información del clúster así como información de configuración de las BBDD que se encuentran en el clúster. Maneja también información sobre los procesos de control del CRS

- Se recomienda la multiplexación del mismo.
- También mantiene información del estado de los recursos. Copias redundantes deben estar en almacenamiento físico separado.
- La ubicación de los ficheros de OCR puede chequearse en el fichero "ocr.loc" en "/etc/oracle" , "/var/opt/oracle" o mediante la ejecución de "ocrcheck"
- Para su manejo se utilizara la utilidad "ocrconfig", pero también puede ser administrado mediante la utilidad "srvctl", grid control y mediante dbca.
- Su contenido puede ser volcado usando la herramienta "ocrdump" (OCRDUMPFILE)
- A partir de Oracle Clusterware 12cR2, los ficheros de OCR sólo se pueden almacenar en Oracle ASM.

Voting Disk: Es un mecanismo de backup para la comunicación entre los nodos del clúster, para intercambiar información crítica de su estado en caso de que el primer mecanismo falle. Todos los nodos están constantemente escribiendo su estado en los voting disk. Es el mecanismo de "heartbeat" en disco. Gestiona los servidores miembros del clúster mediante chequeos de "salud" y decide sobre la propiedad del clúster ("cluster ownership") entre las diferentes instancias en caso de fallo de Red.

- Se recomienda configurar un número impar de voting disks (3, 5, etc). Aunque también se soporta la configuración de números pares.
- Usado para resolver conflictos en casos de "split brain".
- No es un sustituto del cluster interconnect.
- La pérdida o inaccesibilidad de los voting disks significa la pérdida del nodo.
- A partir de Oracle Clusterware 12cR2, los voting disk sólo se pueden almacenar en Oracle ASM.

VIP – Virtual IP: Es un mecanismo que permite a RAC ofrecer un entorno de alta disponibilidad para usuarios y aplicaciones.

- Proporciona un mecanismo para evitar largos retrasos por timeout en TCP (10 minutos aprox) en fallos de nodos.
- En situaciones de fallo elimina el timeout por el establecimiento de una conexión de red. Los clientes "pasan" directamente a la siguiente dirección de la lista. Ya no es necesario el "tuning" de la pila TCP/IP.
- Se recomienda su configuración mediante VIPCA.
- Cuando un nodo falla, CRS mueve la VIP a uno de los otros nodos supervivientes. Cuando el nodo vuelve a funcionar, la VIP pasa de nuevo a su nodo "propietario".

SCAN – Single Client Access Name: Hay un apartado posterior hablando de este componente.

Recursos en Clusterware

¿Que son los recursos de CRS ? Son estructuras de datos almacenadas en el OCR y que da instrucciones al OCR como:

1. Arrancar un proceso
2. Parar un proceso
3. Donde arrancar un proceso y donde realojarlo en caso de fallo.
4. Cuantos reintentos realiza tras un fallo.
5. Dependencias de procesos
6. Etc.

Toda la información de los recursos es almacenada en el OCR. CRS proporciona que los siguientes recursos (Resources) estén en alta disponibilidad:

- Nodeapps: (Recursos estandar "Core")
- VIP (Virtual IP) [.vip]
- ONS (oracle notification service) [.ons]
- GSD (global services daemon) [.gsd]
- Tns listener [.lsnr]
- Instancias de ASM [.asm]
- Base de datos [.db]

- Instancias de la base de datos [.inst]
- Servicios [.srv y .cs]

Todos los recursos estándar son nombrados con la nomenclatura estándar comenzando por "ora."

Como norma no se permite el uso de recursos No-standard, estos son creados para otros procesos relacionados o no con Oracle (pej. http server)

Todos los recursos estándar deben ser gestionados a través del comando SRVCTL.

No usar comandos CRSCTL con los recursos estándar, a menos que se lo haya indicado explícitamente My Oracle Support

Todos los recursos están monitorizados por CRS. Son arrancados en caso de fallo.

Muestra la información de configuración para un recurso específico:

Ejemplo de salida de la información de configuración de un recurso

```
$ crsctl stat res <NOMBRE_RECURSO> -p
NAME=ora.LISTENER_SCAN1.lsnr
TYPE=ora.scan_listener.type
ACL=owner:oracrs:rwx,pgrp:oinstall:r-x,other::r--
ACTION_FAILURE_TEMPLATE=
ACTION_SCRIPT=
ACTIVE_PLACEMENT=1
AGENT_FILENAME=%CRS_HOME%/bin/oraagent%CRS_EXE_SUFFIX%
AUTO_START=restore
CARDINALITY=1
CHECK_INTERVAL=60
CHECK_TIMEOUT=120
DEFAULT_TEMPLATE=PROPERTY(RESOURCE_CLASS=scan_listener)
PROPERTY(LISTENER_NAME=PARSE(%NAME%, ., 2))
DEGREE=1
DESCRIPTION=Oracle SCAN listener resource
ENABLED=1
ENDPOINTS=TCP:1521
FAILOVER_DELAY=0
FAILURE_INTERVAL=0
FAILURE_THRESHOLD=0
HOSTING_MEMBERS=
LOAD=1
LOGGING_LEVEL=1
NLS_LANG=
NOT_RESTARTING_TEMPLATE=
OFFLINE_CHECK_INTERVAL=0
PLACEMENT=balanced
PORT=1521
PROFILE_CHANGE_TEMPLATE=
REGISTRATION_INVITED_NODES=
REGISTRATION_INVITED_SUBNETS=
RESTART_ATTEMPTS=5
SCRIPT_TIMEOUT=60
SERVER_POOLS=*
START_DEPENDENCIES=hard(ora.scan1.vip) dispersion:active(type:ora.scan_listener.type)
pullup(ora.scan1.vip)
START_TIMEOUT=180
STATE_CHANGE_TEMPLATE=
STOP_DEPENDENCIES=hard(intermediate:ora.scan1.vip)
STOP_TIMEOUT=0
TYPE_VERSION=2.2
UPTIME_THRESHOLD=1d
USR_ORA_ENV=
USR_ORA_OPI=false
VERSION=11.2.0.4.0
```

Como norma no se permiten la creación de servicios dentro de Oracle Clusterware, se define servicios como unidades lógicas de trabajo.

• Ej. CRM, Batch ...

Para la gestión de recursos estándar se recomienda el uso del comando srvctl:

srvctl <operacion> <tipo de recurso> <opciones>

<operaciones> - stop, start, add, remove, modify, config, status

<tipo de recurso> - nodeapps (includes listener and vip), asm,

instance, database, service

<opciones> - Estas opciones son diferentes para cada tipo de recurso.

referencia en la documentación.

Para versiones anteriores a 12cR2 existen los comandos:

crs_stop <nombre completo del recurso>

crs_start <nombre completo del recurso>

crs_relocate <nombre completo del recurso>

crs_register <nombre completo del recurso>

Estos comandos no son recomendados para recursos estándar, sin embargo, algunas veces se debe de hacer uso de ellos, cuando el equivalente comando en srvctl no funciona o no soporta algún tipo de operación como (p.ej. Relocate de VIP). Para el uso de estas excepciones consultar con Oracle Soporte.

Como norma de uso y recomendación cuando añadimos recursos estandards se realizara a través del uso de los wizards disponibles para ello, no es necesario añadirlos desde línea de comandos:

DBCA - database (.db) y services (.srv and .cs)

NETCA - Listeners (.lsnr)

VIPCA - Vip (.vip)

Para consultar la información de los recursos estándar se puede hacer uso de los siguientes comandos

- srvctl status [asm | nodeapps] –n <node list>
- srvctl config [database | service] –d <DB>

- srvctl config [asm|nodeapps] –n [options]
- srvctl –h nos muestra la ayuda para todos los comandos

Habilitación y deshabilitar de los recursos estandards

- srvctl enable|disable <tipo_recurso> -d|-n <database|node> [-i] <instance>

Realocación de recursos estandard a otro nodo

- srvctl relocate service -d APS -s GL -i APS1 –t APS2 [-f]

Best Practices de Backup en Clusterware

Este apartado cubre un conjunto de normas y recomendaciones a la hora de realizar backup del software de Clusterware y sus componentes. Para ello vamos a ir comentando cuestiones como "qué", "cómo" y "cuándo" que pueden surgir a la hora de poner en marcha una política de backup de Oracle Clusterware

En este apartado vamos a responder a las siguientes preguntas

1.- **¿De qué componentes hay que hacer backup?**

- CRS HOME
- Inventario Central
- Voting Disk
- OCR
- Scripts de Init

2.- **¿Cuándo realizar backups de estos componentes?**

- Antes y después de instalar Patchset y MLRs
- Antes y después de añadir recursos
- Antes y después de hacer cambios estructurales en el OCR y/o en el Voting disk
- Antes y después de añadir un nodo

3.- **¿Qué componentes de clusterware tengo que realizar backup?**

Copia de seguridad de los ficheros de **Voting Disk:**

- En 12cR2 Oracle Clusterware realiza automáticamente una copia de seguridad de los datos del archivo de voting en OCR como parte de cualquier

cambio de configuración y restaura automáticamente los datos en cualquier archivo de voting que se agregue.

• Pre-12cR2 sin ASM, se puede usar dd para hacer una copia del voting disk a un fichero (puede ser hecho con el usuario oracle)

$ dd if=/dev/raw/raw5 of=/backups/dd_vote_disk bs=1024k

• Copiar el fichero /backup/dd_vote_disk a cinta

Copia de seguridad de los ficheros de **OCR:**

• En 12cR2 se pueden realizar backups automáticos y manuales:

$ ocrconfig -manualbackup

• Pre-12cR2 se puede usar: Copia de seguridad a ficheros de texto, siempre hacer esto en caso de que no puedas restaurar desde cualquier tipo de backup.

$ ocrdump ocr_text_backup.txt

• Pre-12cR2 sin ASM, se puede usar "dd". Es la forma más simple y sencilla de realizar una copia de seguridad de OCR.

$ dd if=/de/raw/raw1 of=/backups/dd_ocr bs=1024k

Nota: El OCR y su mirror no son idénticos, por lo que es necesario realizar backups por separado de cada uno de ellos.

• Realizar un export backup del OCR. No es necesario parada del CRS.

 # ocrconfig –export file_name

Por último, de forma automática se realizan regularmente backups binarios del OCR desde CRS

• El contenido del OCR es crítico para Oracle Clusterware.

• OCR es automáticamente copiado a fichero

• Cada cuatro horas: CRS mantiene las últimas tres copias.

• Al final de cada día: CRS mantiene las últimas dos copias.

• Al final de cada semana: CRS mantiene las últimas dos copias.

Como norma genérica no se recomienda el cambio de la ruta por defecto del almacén de los backups automáticos. Para mostrar el listado de backups que realiza automáticamente ejecutarlo a través del siguiente comando:

$ ocrconfig –showbackup

Ejemplo de salida de la información de backups de OCR
ocrconfig –showbackup raclinux64 2008/04/13 16:53:58 /u01/app/oracle/product/10.2.0/crs/cdata/cluster1 raclinux63 2008/04/13 10:38:32 /u01/app/oracle/product/10.2.0/crs/cdata/cluster1 raclinux63 2008/04/13 06:38:31 /u01/app/oracle/product/10.2.0/crs/cdata/cluster1 raclinux63 2008/04/12 04:57:27 /u01/app/oracle/product/10.2.0/crs/cdata/cluster1 raclinux63 2008/04/09 14:32:57 /u01/app/oracle/product/10.2.0/crs/cdata/cluster1

ls /u01/app/oracle/product/10.2.0/crs/cdata/cluster1

backup00.ocr backup02.ocr day.ocr week.ocr

backup01.ocr day_.ocr dd_ocr ocr_export

Copiar los ficheros listados anteriormente a cinta.

Por último, se recomienda la realización de copias de seguridad de los ficheros de arranque, para ello realizar copias a disco o cinta de los siguientes ficheros y directorio. Este tipo de backups es obligatorio a la hora de realizar el upgrade de un Patchset o aplicación de parches específicos, pero como buena práctica se recomienda su copia de forma periódica.

Ficheros y directorios de arranque a realizar backup en Clusterware
Ficheros a copiar: */etc/oracle/scls_scr* */etc/oracle/oprocd* */etc/inittab* */etc/oracle/ocr.loc* ***Directorios a copiar:*** *cd /etc* *mkdir init_sav* *find . –name "*crs*" –print* *find . –name "*css*" –print* *find . –name "*evm*" –print*

RECOMENDACIONES DE RED Y CONECTIVIDAD PARA ENTORNOS RAC

En este apartado se comentaran las diferentes opciones que existen de conexión a una base de datos en un entorno RAC y cómo deben quedar configurados los elementos de conexión a una base de datos Oracle en RAC. Este documento se encuentra focalizado en la parte de sistemas por lo que se hará especial hincapié en lo correspondiente a la configuración que deben realizar los administradores de bases de datos y temas como los tipos de conexiones posibles, uso de XA, etc. Se tratara en el documento de buenas prácticas de desarrollo en RAC.

Grid Infrastructure Single Client Access Name (SCAN)

Oracle Clusterware puede usar el Single Client Access Name (SCAN) para la configuración dinámica de direcciones VIP, eliminando la necesidad de realizar la configuración de servidor de forma manual.

SCAN es un nombre de dominio registrado en al menos una y hasta tres direcciones IP, ya sea en DNS o GNS con DHCP. Se recomiendan 3 direcciones IP y siempre que sea posible, configurarlo en DNS.

Se debe configurar lo siguiente:

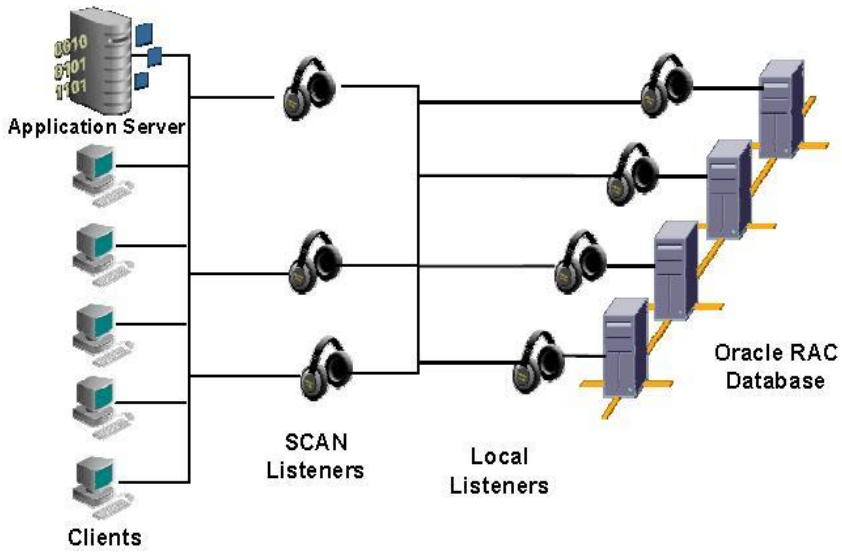
- Una dirección pública y nombre de host para cada nodo del cluster.
- Una dirección VIP para cada nodo. Debe asignar una dirección VIP a cada nodo del clúster. Cada dirección VIP debe estar en la misma subred que la dirección IP pública del nodo y debe ser una dirección a la que se le asigne un nombre en el DNS. Estas VIPs no deben estar en uso desde dentro de la red antes de instalar Oracle Grid Infrastructure.
- Un SCAN name es el hostname virtual, registrado con hasta tres direcciones IP de SCAN para el clúster. Para una alta disponibilidad y escalabilidad, Oracle recomienda que se configuren 3 IPs de SCAN

A tener en cuenta:

- SCAN es una parte elemental de Grid Infrastructure desde 11gR2 y no está soportado eliminarlo. Por lo que debe ser el método de conexión utilizado.
- Oracle recomienda que todas las conexiones a las bases de datos Oracle RAC (y single-instances gestionadas con Clusterware) usen SCAN en su cadena de conexión de cliente.
- Con SCAN, no tiene que cambiar la cadena de conexión del cliente ni siquiera cuando la configuración del clúster cambia (nodos añadidos o eliminados).
- El registro del SCAN name se debe hacer en DNS o GNS, no se puede usar el /etc/hosts para resolver la SCAN
- En Grid Infrastructure 11gR2 y 12cR1, en una instalación "Typical", el SCAN name se correspondía con el cluster name.

- Si el SCAN name y el nombre del clúster se ingresan en el mismo durante la instalación, los requisitos del nombre del clúster se aplican al SCAN name.
- En Grid Infrastructure 11gR2 y 12cR1, en la instalación "Advanced", se indican en campos separados. Y a partir de Grid Infrastructure 12cR2, el SCAN name y el nombre del clúster se ingresan en campos separados durante la instalación.
- El nombre del clúster "cluster name" no distingue entre mayúsculas y minúsculas (es case-insensitive), debe ser único en la organización, debe tener al menos un carácter de longitud y no más de 15 caracteres, debe ser alfanumérico, no puede comenzar con un número y puede contener guiones (-), pero los caracteres de subrayado (_) no están permitidos.
- Seleccione el nombre de clúster cuidadosamente. Después de la instalación, solo puede cambiar el nombre del clúster reinstalando Oracle Grid Infrastructure.
- Si el SCAN name y el nombre del clúster se ingresan en campos separados durante la instalación, los requisitos del nombre del clúster no se aplican al SCAN name, que puede tener más de 15 caracteres.
- Desde Oracle Database 12cR2, los SCAN listeners soportan el protocolo Http
- Desde Grid Infrastructure 12cR1, se soporta IPv6 en la red pública; esto incluye SCAN, IPs y VIPs públicas
- Desde Grid Infrastructure 12cR1, SCAN admite varias subredes en el clúster (una SCAN por subred)

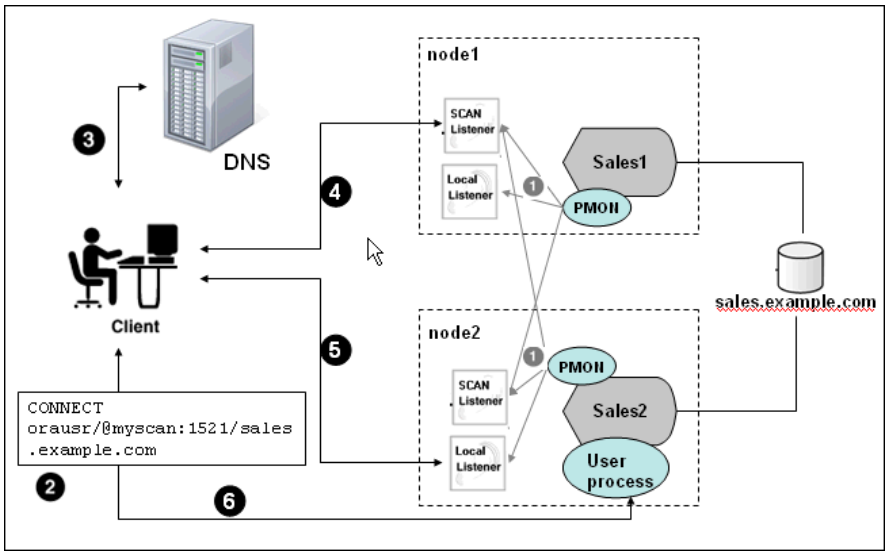
Cómo se realizan las conexiones usando SCAN:



Cuando un cliente envía una petición de conexión al SCAN name, este resuelve por DNS (o GNS con DHCP) a una de las 3 IPs de SCAN, que contacta con su SCAN listener que está escuchando en su SCAN VIP por el puerto especificado (SCAN port). Como todos los servicios en el cluster están registrados con el SCAN listener, el SCAN listener responde al cliente con la dirección del local listener de la instancia y nodo con

menos carga (cada SCAN listener mantiene las estadísticas de carga del cluster actualizadas) de los que ofrecen actualmente el servicio de BBDD. El cliente conecta con el local listener indicado, quien inicia un proceso de servidor dedicado para la conexión a la base de datos. Finalmente, el cliente conecta directamente con el proceso de servidor dedicado y accede a la base de datos por la instancia que se está ejecutando en el nodo del local listener devuelto. Todas estas operaciones son transparentes para el cliente, sin necesidad de realizar ninguna configuración explícita.

Load Balancing en Grid Infrastructure usando Single Client Access Name (SCAN):



1.- El proceso PMON de cada instancia registra el servicio de base de datos con el listener por defecto del nodo local y con cada SCAN listener, que está especificado por el parámetro de base de datos REMOTE_LISTENER. Los listeners son dinámicamente actualizados de la cantidad de trabajo que está siendo manejado por las instancias y los dispatchers.

2.- El cliente realiza una conexión usando el descriptor de la conexión con: usuario/passwd, nombre de SCAN, puerto y nombre de la BBDD:

```
SQL> connect orausr/@scan_name:1521/sales.example.com
bash$ sqlplus orausr@sales (sqlnet.ora)
```

3.- El cliente consulta al DNS para resolver el scan_name. El nombre de SCAN estará asociado normalmente a 3 IP's en DNS con round-robin. El DNS devuelve al cliente esas 3 direcciones asignadas al scan_name, el cliente envía una petición de conexión a la primera dirección IP. Si la petición de conexión falla, entonces el cliente envía una nueva petición de conexión usando la siguiente dirección IP.

4.- Cuando la petición de conexión es satisfactoria, el cliente conecta con el SCAN listener del cluster que contiene la base de datos "sales" en este ejemplo. El SCAN listener compara la carga de trabajo de las instancias sales1 y sales2 de la base de datos sales y la carga de trabajo de los nodos en los que se ejecuta. En este ejemplo, el node2 tiene menos carga que el node1, por lo que el SCAN listener selecciona el node2 y envía la dirección del local listener de ese nodo (node2) de vuelta al cliente.

5.- El cliente conecta con el local listener del node2, que está escuchando en la VIP local y puerto especificado. El local listener inicia un proceso de servidor dedicado para la conexión a la base de datos.

6.- El cliente conecta directamente con el proceso de servidor dedicado del node2 y accede a la instancia de base de datos sales2.

Consideraciones generales de Red

- Son necesarias al menos 2 interfaces físicas por nodo:
  - 1 para la red pública
  - 1 para la red de interconexión privada (interconnect). The interconnect network is a private network using a switch (or multiple switches) that only the nodes in the cluster can access
- Pero está recomendado el uso de 4 interfaces físicas por nodo:
  - 2 interfaces de red para la red pública, unidas para proporcionar una dirección (bonding/IPMP/aggregation...)
  - 2 interfaces de red para la interconnect redundadas con:
    - Anterior a 11.2.0.2 con una tecnología de IP failover de terceros (bonding/IPMP/aggregation...)
    - A partir de 11.2.0.2 con HAIP (Highly Available IP (address)) que es nativo de Grid Infrastructure
- Oracle no soporta el uso de cables de conexión cruzada para la red de interconnect
- 1 Nombre de SCAN (Single Client Access Name) que es el nombre de cluster usado por los clientes para conectar con el cluster. El nombre de SCAN estará asociado normalmente a 3 IP's, ya sea en DNS ó GNS. De esta forma, cuando un cliente interroga al DNS por el nombre de SCAN, éste responde con una IP distinta cada vez (de entre las tres configuradas) y de forma cíclica
- Direcciones IP
  - Son necesarias al menos 3 IP’s fijas por nodo. Dos de ellas públicas (la física del nodo y la VIP) y otra será la IP privada que servirá de interconnect entre ambos nodos
  - Son necesarias también 3 IP's de Single Client Access Name (SCAN) para el cluster.
- Guión bajo no debe ser usado en un hostname o nombre de dominio de acuerdo a la norma RFC952 - DoD Internet host table specification. Lo mismo aplica para la red, gateway. Referencia: http://www.faqs.org/rfcs/rfc952.html

- Asegúrate que el default Gateway está en la misma subred que la VIP y SCAN VIPs. De otra forma, esto puede causar problemas con racgvip y causar que la vip y el listener se reinicien periódicamente.

- Como recomendación, usa el mismo nombre de interfaces en todos los nodos. Para chequear usa ifconfig (en Unix) o ipconfig (en Windows).

- Usar Jumbo Frames está soportado y es posible a nivel de sistema. Como documento de referencia usar la siguiente nota#341788.1

- Usar direcciones de red especificas para el private interconnect, como por ejemplo: Clase A: 10.0.0.0 - 10.255.255.255, Clase B: 172.16.0.0 - 172.31.255.255 o Clase C: 192.168.0.0 - 192.168.255.255.

- Asegúrate que las interfaces de red están configuradas exactamente igual en términos de velocidad, dúplex, etc. Puedes hacer uso de varias herramientas como: ethtool, iperf, netperf, spray y tcp. Documentadas en la nota#563566.1

- Se recomienda configurar las interfaces de red Pública con redundancia / tolerancia a fallos con tecnologías como bonding. Documentado en la nota#787420.1.

- Desde la versión 11.2.0.2 en adelante, se recomienda configurar las interfaces de red de Interconnect con redundancia / tolerancia a fallos con la funcionalidad embebida de Oracle Clusterware "Redundant Interconnect/HAIP".

- Configurar la dirección IPC como la primera en el fichero de listener.ora. Para bases de datos que vienen migradas de anteriores versionas a 10gR2 el netca no configura la entrada IPC en el fichero listener.ora. En 10gR2 este es el comportamiento por defecto pero si se migra habría que añadirlo manualmente. Con ello evitamos que la cantidad de tiempo que VIP necesita para failover es mucho menor que si no se encuentra configurada, como documento de referencia indicar la nota#403743.1

- Incrementar el tamaño de SDU a un valor mayor (por ej: de 4KB 8KB, a 32KB).
- Desde Grid Infrastructure 12cR1, se soporta IPv6 en la red pública; esto incluye SCAN, IPs y VIPs públicas
- Desde Grid Infrastructure 12cR2, se soporta IPv6 en la red de interconnect

- Los nombres de las interfaces de red (NIC) no deben contener “.”

Configuración de componentes de red en Oracle RAC

Configuración del fichero de Listener (listener.ora)

En entornos de Oracle RAC 11gR2 y superior se pueden configurar los listeners dentro del fichero listener.ora del GI_HOME de cada uno de los nodos del cluster, en el caso que se haga uso de software local o el fichero listener.ora único cuando se hace uso de software compartido para todos los nodos del clúster.

- En cualquier configuración de base de datos Oracle, los listeners definen las instancias como locales o remotas (En un entorno single-instance, normalmente son todas locales). Se puede ver el comportamiento y la configuración ejecutando el siguiente comando:
  - lsnrctl services <listener_name>"
- Un servicio de listener que se encuentra registrado por una instancia, esto es lo que se conoce como local, y el remote listener especifica una lista de instancias donde se registraran los servicios.
- Para el registro de listener local se recomienda hacer uso de la siguiente nomenclatura:

```
Entrada en el fichero de parámetros spfile.ora:

sid.local_listener=listener_sid

Entrada en el fichero tnsnames.ora:

listener_sid=(address=(protocol=tcp)(host=node1-vip)(port=1522))
```

- Si se indica el parámetro de base de datos REMOTE_LISTENER manualmente, hay que configurarlo a `scan_name:scan_port`

Configuración de los servicios de red (Fichero Tnsnames.ora)

El fichero de tnsnames.ora es creado en cada nodo. Cada entrada en este fichero equivale a un identificador de conexión que incluye la ruta de red contra un servicio que se encuentra disponible en el servidor de base de datos.

Tipo de servicio de red	Descripción

Conexión de base de datos	Clientes que conectan a cualquier instancia usando la entrada de servicios de red.
	Ejemplo recomendado antes de 11gR2 o sin SCAN:
	`db.example.com=`
	`(description=`
	`(load_balance=on)`
	`(address=(protocol=tcp)(host=node1-vip)(port=1521))`
	`(address=(protocol=tcp)(host=node2-vip)(port=1521))`
	`(connect_data=(service_name=db.example.com)))`
	Ejemplo recomendado desde 11gR2 con SCAN:
	`db.example.com=`
	`(description=`
	`(address=(protocol=tcp)(host= scan-name-rac.example.com)(port=1521))`
	`(connect_data=(service_name=db.example.com)))`

Tipo de servicio de red	Descripción

Conexión a instancias	Por defecto la conexión se realizara a conexión de la base de datos tal y como se comenta en la entrada anterior, pero si por algún motivo se permite la conexión directa a una instancia directamente, se debe realizar de la siguiente forma:
	`db1.example.com=`
	`(description=`
	`(address=(protocol=tcp)(host=node1-vip)(port=1521))`
	`(connect_data=`
	`(service_name=db.example.com)`
	`(instance_name=db1)))`

Servicios	Cuando se configura servicios de alta disponibilidad:
	Ejemplo recomendado antes de 11gR2 o sin SCAN:
	`db_servicio =`
	`(description =`
	`(address=(protocol=tcp)(host=node1-vip)(port=1521))`
	`(address=(protocol=tcp)(host=node2-vip)(port=1521))`
	`(load_balance=yes)`
	`(connect_data=`
	`(server = dedicated)`
	`(service_name = servicio.example.com)`
	`)))`
	Ejemplo recomendado desde 11gR2 con SCAN:
	`db_servicio =`
	`(description=`
	`(address=(protocol=tcp)(host= scan-name-rac.example.com)(port=1521))`
	`(connect_data=(service_name=db_servicio)))`

Comportamiento de las VIP

Las "node VIP" o también llamadas "local VIP" o simplemente VIP son las IPs virtuales de los nodos. Oracle Clusterware/Real Application Cluster aloja las "node VIP" en la red pública. Las VIP de nodo son las direcciones que usan los clientes para conectarse a una base de datos Oracle RAC.

En el sub-apartado anterior hemos visto cómo se realizan las conexiones típicas de un cliente de base de datos a una instancia de base de datos Oracle RAC, usando SCAN que es el método recomendado.

Las node VIP de RAC están diseñadas para trabajar en nodos del clúster donde existen instancias de bases de datos activas y tan solo aceptan conexiones cuando están activas en nodo donde fueron asignadas y que llamaremos HOME NODE.

Si un nodo falla, Oracle Clusterware conmuta su dirección VIP a otro nodo del cluster disponible en el que la dirección VIP pueda aceptar conexiones TCP, pero no acepta conexiones a la base de datos Oracle.

Los clientes que intentan conectarse a una VIP que no reside en su HOME NODE reciben un error de rechazo de conexión rápida en lugar de esperar un timeout de conexión TCP. El cliente intentará inmediatamente una conexión a la siguiente dirección de la lista o SCAN VIP.

Cuando la red en la que se configura la VIP vuelve a estar en línea, Oracle Clusterware devuelve la VIP a su HOME NODE, donde se aceptan las conexiones.

En general, las direcciones VIP fallan cuando:

- El nodo en el que se ejecuta una dirección VIP falla
- Todas las interfaces de la dirección VIP fallan
- Todas las interfaces de la dirección VIP están desconectadas de la red

La recomendación general es usar SCAN con database Services para redirigir los clientes a los diferentes nodos del clúster, ya que pueden ser dinámicamente realojados y modificados en cualquier momento.

Oracle WebLogic Server 12c Active GridLink (AGL)

En servidores de aplicaciones de tipo JEE, las interacciones con la base de datos son manejadas normalmente a través de data source (DS) JDBC.

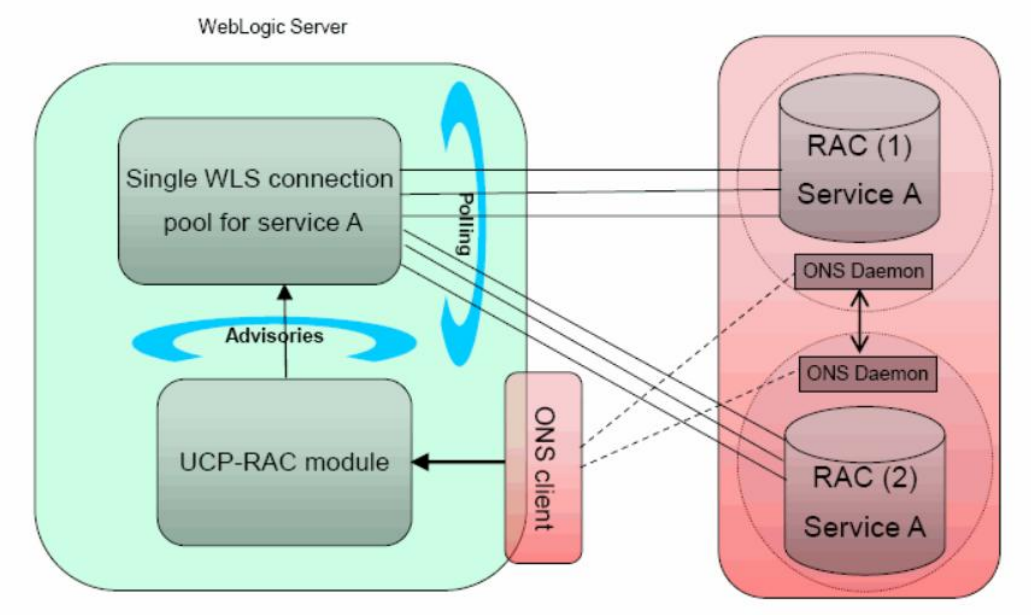
En Oracle WebLogic Server (WLS), se configura la conectividad con la base de datos mediante la configuración de DS y éste de tipo GridLink, desplegando éstos como recursos JDBC en los distintos servidores y/o clústeres en un dominio de WebLogic.

Cada DS que se configura contiene un pool de conexiones a una base de datos. Estos DS se crean en su proceso de creación, en el despliegue o targeting de la instancia del DS en un servidor y/ clúster así como en el proceso de arranque del servidor.

Un único origen de datos AGL proporciona conectividad entre WebLogic Server y un servicio de Oracle Database, que puede incluir uno o más clústeres de Oracle RAC.

Oracle no recomienda el uso de datasources genéricos con clústeres de Oracle RAC.

Las aplicaciones usan un DS GridLink a través de JNDI o en el contexto local de la aplicación (java:comp/env) de la misma forma que con un DS tradicional y este es el que realmente solicita la conexión a base de datos.



Un origen de datos AGL incluye las características de fuentes de datos genéricas más el siguiente soporte para Oracle RAC:

- Fast Connection Failover
- Runtime Connection Load Balancing
- GridLink Affinity
- SCAN Addresses
- Secure Communication using Oracle Wallet with ONS Listener.

El alcance de este documento no se centrara en el análisis de las características anteriores citadas, sino en desarrollar la correcta configuración y requerimientos para el uso de Active GridLink como configuración estándar de datasources contra base de datos en Real Application Cluster.

Para crear un origen de datos AGL en su dominio WebLogic, puede utilizar la Consola de administración de WebLogic Server o la Herramienta de comandos WebLogic (WLST).

Los siguientes apartados proporcionan una descripción general de los pasos básicos utilizados en el asistente de configuración de origen de datos para crear un origen de datos mediante la Consola de administración de WebLogic Server:

- JDBC Data Source Properties
- Configure Transaction Options
- Configure Connection Properties
- Test Connections
- ONS Client Configuration
- Test ONS Client Configuration
- Target the Data Source

Oracle WebLogic Server Multidatasources (MDS)

En servidores de aplicaciones de tipo JEE, las interacciones con la base de datos son manejadas normalmente a través de data source (DS) JDBC.

En Oracle WebLogic Server (WLS), se configura la conectividad con la base de datos mediante la configuración de DS y multi data sources (MDS) JDBC, desplegando éstos como recursos JDBC en los distintos servidores y/o clústeres en un dominio de WebLogic.

Cada DS que se configura contiene un pool de conexiones a una base de datos. Estos DS se crean en su proceso de creación, en el despliegue o targeting de la instancia del DS en un servidor y/ clúster así como en el proceso de arranque del servidor.

Un MDS es una abstracción de un conjunto de DS que proporciona balanceo de carga o funciones de failover en todo el conjunto de DS asociados al MDS.

Las aplicaciones usan un MDS a través de JNDI o en el contexto local de la aplicación (java:comp/env) de la misma forma que con un DS tradicional y éstos son los que realmente solicitan la conexión a base de datos.

Cada MDS determina que DS es el más adecuado para satisfacer las necesidades de la petición en función del algoritmo seleccionado en la configuración del MDS.

Existen dos algoritmos disponibles para los MDS:

- Balanceo de carga: El acceso a los pools en WLS se hace mediante el método de round-robin. Al solicitase una nueva conexión, WLS seleccionará una conexión basada en el siguiente pool de conexiones en el orden especificado.

- Alta disponibilidad: Los pools de conexiones se muestran en el orden que se determina cuando se produce una conmutación o failover en el pool de conexiones. WLS intentará obtener de conexiones de base de datos empezando por el primer pool de conexiones de la lista. Si por alguna razón, ese pool falla, utiliza el siguiente según el orden especificado.

Configurando Multi data sources con o sin transacciones globales

Hay tres consideraciones para determinar cómo debe ser configurado los MDS de WLS para hacer uso de las capacidades y funcionalidades de Oracle RAC.

La tabla siguiente muestra las claves:

¿Balanceo de carga?	¿Failover?	¿XA?	¿JDBC Store?	Recomendación
Si	Si	Si		MDS con soporte XA
Si	Si		Si	MDS sin soporte XA
	Si		Si	Connect time Failover sin XA (1)

(1) No incluida en la arquitectura de referencia, solo con fin informativo.

Multi data sources con transacciones globales

Si tus aplicaciones requieren soporte para transacciones globales al acceder a Oracle RAC se debe considerar el uso de MDS con la opción de configuración de soporte de transacciones distribuidas (XA). Teniendo en cuenta que en caso de failover, será manejado por los MDS en lugar de Oracle RAC.

En un escenario con 2PC, si hay un fallo en una instancia de Oracle RAC antes del PREPARE, la operación se reintenta hasta que expira el tiempo de RETRY. Si hay un fallo después de PREPARE, la transacción se pasa a otra instancia.

Esta opción requiere que todos los DS definidos para el MDS usen un driver XA-enabled, o que ninguno lo tenga. Además, todas las propiedades relacionadas con XA deben tener el mismo valor para todos los DS.

Hay varios atributos que es necesario configurar en los DS, a continuación vemos como:

- Primero, es necesario establecer el tipo de driver, en este caso, el driver será ORACLE JDBC THIN.

```
<url>jdbc:oracle:thin:@lcqsol24:1521:SNRAC1</url>
<drivername>
oracle.jdbc.xa.client.OracleXADataSource</driver-name>
```

- Configurar KeepXAConnTillTxComplete="true", para forzar a los DS a reservar una conexión física a la base de datos y a permanecer en el mismo ámbito de la transacción durante su ciclo de vida para esta aplicación en particular.

- Configurar XARetryDurationSeconds="300", el valor es el tiempo durante el cual el *transaction manager* de WLS reintentará para llamadas XA de tipo recover, commit o rollback.

- Configurar TestConnectionsOnReserve="true", se usa para verificar las conexiones a la base de datos. Puede ser acompañado con un atributo TestTableName.

Parámetro	Valor
Driver JDBC	Oracle JDBC Thin
KeepXAConnTillTxComplete	TRUE
XARetryDurationSeconds	300
TestConnectionsOnReserve	TRUE

En el siguiente ejemplo, se muestra la configuración parcial que incluye las propiedades requeridas para utilizar MDS con Oracle RAC soportando transacciones globales de tipo XA:

```
<jdbc-driver-params>
   <url>jdbc:oracle:thin:@lcqsol24:1521:SNRAC1</url>
<driver-name>
oracle.jdbc.xa.client.OracleXADataSource</driver-name>
……
</jdbc-driver-params>
   <jdbc-connection-pool-params>
   <test-table-name>SQL SELECT 1 FROM DUAL</test-table-name>
   <profile-type>0</profile-type>
</jdbc-connection-pool-params>
<jdbc-data-source-params>
   <jndi-name>oracleRACXAJndiName</jndi-name>
   <global-transactions-protocol>TwoPhaseCommit
   </global-transactions-protocol>
</jdbc-data-source-params>
<jdbc-xa-params>
   <keep-xa-conn-till-tx-complete>true</keep-xa-conn-tilltx-
   complete>
   <xa-end-only-once>true</xa-end-only-once>
   <xa-set-transaction-timeout>true</xa-set-transactiontimeout>
   <xa-transaction-timeout>120</xa-transaction-timeout>
   <xa-retry-duration-seconds>300</xa-retry-durationseconds>
```

Multi data sources sin transacciones globales

Si las aplicaciones no requieren soporte para transacciones globales de tipo XA, se deben configurar MDS para manejar los procedeos de failover y el balanceo de carga sin especificar las propiedades de configuración XA.

Desde la perspectiva de la configuración, la única diferencia con respecto a la configuración anterior es que no hay que especificar los atributos relacionados con XA.

A continuación encontramos un ejemplo de configuración:

```
<jdbc-driver-params>
   <url>jdbc:oracle:thin:@lcqsol24:1521:snrac1</url>
   <driver-name>oracle.jdbc.OracleDriver</driver-name>
   <properties>
     <property>
       <name>user</name>
       <value>wlsqa</value>
     </property>
   </properties>
   <password-encrypted>{3DES}aP/xScCS8uI=</passwordencrypted>
</jdbc-driver-params>
<jdbc-connection-pool-params>
   <test-connections-on-reserve>true</test-connections-onreserve>
   <test-table-name>SQL SELECT 1 FROM DUAL</test-table-name>
</jdbc-connection-pool-params>
<jdbc-data-source-params>
   <jndi-name>jdbcDataSource</jndi-name>
</jdbc-data-source-params>
```

Network Failover (NF)

Network failover o *connection failover (NF* o *CF)* es el mecanismo de failover por defecto, el más básico y el único disponible para el uso del driver *JDBC* de tipo *THIN* aunque también está disponible para el *driver JDBC Thick OCI8*.

NF asegura que las nuevas y solo las nuevas conexiones que establecen los servidores de aplicaciones contra una instancia de la base de datos en *RAC* que ha tenido una situación de no-disponibilidad se crean contra una de las instancias de backup o supervivientes de la base de datos en clúster incluso si se usa el alias *TNS* correspondiente a la instancia no disponible.

NF es el único mecanismo de failover donde las conexiones existentes no son automáticamente recreadas en las instancias del *RAC* supervivientes. Estas conexiones no podrán ser usadas en caso de failover y recibirán el error *ORA-03113* tras la primera ejecución tras el failover y el error *ORA-03114* en las sucesivas por parte del propio cliente.

Las conexiones en proceso de conexión durante la no-disponibilidad de la instancia, incluidas las operaciones de *Oracle Advanced Queueing*, pueden fallar con una gran variedad de excepciones dependiente en el momento que se encuentre.

Esta situación conlleva, por ejemplo, que se tenga que reiniciar un contenedor *J2EE* para que las aplicaciones contenidas puedan crear de nuevo el pool de conexiones contra la base de datos o que la aplicación tenga que implementar una funcionalidad de reinicio del *pool* de conexiones de manera interna a la aplicación.

Transparent Application Failover (TAF)

Transparent Application Failover (TAF) usa un mecanismo de failover en tiempo de ejecución creado para entornos de *HA*, como *RAC* y *Oracle DataGuard* que permite el restablecimiento de las conexiones con la base de datos.

Esto permite que las aplicaciones clientes se reconecten automáticamente a la base de datos si la conexión falla y opcionalmente, continuar la sentencia *SQL* o reejecutando la sentencia *DML* o *DDL* que estaban en progreso.

Esta reconexión automática es posible gracias al interfaz *Oracle Call Interface (OCI)* en su versión 8. Por tanto *TAF* solo está disponible en el tipo de *driver JDBC Thick* de tipo *OCI8*, y pasa su activación es necesario configurar el parámetro *FAILOVER_MODE* como parte de la cláusula *CONNECT_DATA* del alias *TNS* usada para la creación de la conexión *JDBC* o bien codificarla mediante la URL de conexión.

TAF posee el mejor mecanismos de failover para las conexiones en proceso de conexión a una instancia que es parte de una base de datos de un clúster RAC. Además asegura que las conexiones existentes y que no están siendo usadas en el momento del failover son reconectadas las instancias de bases de datos supervivientes.

Sin embargo, existe la posibilidad de que *TAF* no pueda reejecutar una operación transaccional realizada desde la última operación de *commit*. Cuando esto ocurra el cliente normalmente eleva el error *ORA-25408 "Cannot safely replay call"* y será responsabilidad de la aplicación realizar explícitamente un *rollback* de la transacción en proceso antes de poder usar la nueva conexión establecida sobre otra de las instancias de la base de datos.

Es importante dejar claro cuáles son las funcionales esperadas y cuáles son las funcionalidades que no implementa *TAF*, por tanto, el nivel de protección de *TAF* cubre los siguientes elementos:

- Conexiones creadas contra la base de datos.
- Estado de las sesiones de usuario.
- Sentencias en modo *prepared*
- Cursores activos de sentencias *SELECT* que comenzaron a devolver resultados en el momento del fallo.

TAF no protege ni soporta failover en las siguientes situaciones o elementos:

- Aplicaciones que no usen la interfaz *OCI8* ó superior.
- Variables de aplicación de tipo *server-side*, como estados de los paquetes *PL/SQL*.
- Las transacciones activas de sentencias *UPDATE*.

TAF implementa un sistema de mensajería asíncrona basada en eventos que permite a la base de datos comunicarse con los clientes que estén usando este tipo de conexión.

Fast Connection Failover (FCF)

Fast Connection Failover (FCF) permite que las aplicaciones *JDBC* aprovechen las ventajas de mecanismos de failover independientemente del tipo de driver *JDBC*, thin o thick. La única requisito es que solo está disponible a partir de la versión 10.1 de los drivers *JDBC* de Oracle.

Este caso, existe un gestor de conexiones o cache manager que limpia todas las conexiones inválidas de una aplicación *JDBC* después de un failover. En el momento que una aplicación *JDBC* intenta usar unas de esas conexiones invalidas, ésta recibe un error *ORA-17008, "Closed Connection"* y es la aplicación la que debe manejar la excepción y reconectarse.

FCF no reconecta las conexiones existentes, si el nodo que alberga la instancia cae, la aplicación debe cerrar las conexiones y adquirir nuevas conexiones a las instancias supervivientes del *RAC*. La forma de notificación que tiene *FCF* para comunicar a la aplicación *JDBC* que debe iniciar el proceso de reconexiones es a través de una excepción.

Estas excepciones están provocadas por diversos tipos de errores *ORA*, cada uno de ellos caracteriza a diferentes situaciones de caídas, cortes, etc. Por tanto la aplicación de prevenir todos estos tipos de *ORA* y reacción en consecuencia en función de las reglas de negocio.

Como soporte en este tipo de situaciones el *API FCF* ofrece una función para comprobar si el error producido es fatal o si por el contrario existe la posibilidad de continuar. Esta función se implementa a través del método *isFatalConnectionError()* de la clase *OracleConnectionCacheManager*.

En caso de que este método devuelva *TRUE*, se debe ejecutar el método *getConnection* sobre el *datasource*.

Para que una aplicación *JDBC* pueda hacer uso de la funcionalidad de *FCF* debe configurarse de la siguiente forma:

- Se debe configurar e iniciar *ONS*. Si *ONS* no está correctamente configurado, la creación de la cache de conexiones fallará levantando una excepción *ONSException* cuando se realice la primera petición *getConnection()*.

- Además, se debe habilitar la propiedad *FastConnectionFailoverEnabled* antes de realizar la primera llamada al método *getConnection* de *OracleDataSource*, ya que cuando *FCF* está activo, se aplica a todas las conexiones del *connection cache*. Si la aplicación *JDBC* crea conexiones de forma explícita usando *Connection Cache Manager*, ésta debe habilitar el atributo *setFastConnectionFailoverEnabled* antes de adquirir la conexión.

- Por último, se debe usar *SERVICE_NAME* y no el *SID* de la base de datos propiedad *URL* del *OracleDataSource*.

RECOMENDACIONES DE ALMACENAMIENTO EN RAC

Este apartado describe las recomendaciones respecto al almacenamiento en Oracle Real Application Clusters (Oracle RAC).

También incluye los topics que se muestran en la siguiente lista:

- Almacenamiento en Oracle Real Application Clusters
- Acceso a datafiles Access en Oracle Real Application Clusters
- Almacenamiento de los ficheros de Redo Log en Oracle RAC
- Gestión automatic de Undo en Oracle Real Application Clusters
- Automatic Storage Management in Oracle Real Application Clusters

Tipo de almacenamiento en entornos RAC

Como recomendación todos los archivos de datos o datafiles (incluyendo el tablespace de UNDO para cada instancia) y ficheros de redolog (al menos dos para cada instancia) deben residir o en un diskgroup de discos ASM, o en un sistema de archivos en clúster, o en raw devices compartidos.

Además, Oracle recomienda que se utilice un único fichero de parámetros compartidos (SPFILE) con entradas específicas para cada instancia. Alternativamente, se puede utilizar un sistema de archivos local para almacenar los archivos de parámetros específicos de cada instancia (PFiles).

A menos que se indique lo contrario, la base de datos Oracle seguirá las recomendaciones de almacenamiento siguiendo las best practices de Oracle Management Files (OMF), la gestión automática de espacio en los segmentos, y así sucesivamente.

Si no se hace uso de ASM, si la plataforma donde se desplegara Oracle RAC no es compatible con un sistema de archivos en clúster, o si no se quiere utilizar un sistema de archivos en clúster para el almacenamiento de archivos de base de datos, entonces, necesita crear raw devices que den soporte a Oracle Real Application Clusters, tal y como se indica en las guías de instalación y configuración. Sin embargo, se recomienda que se utilice para el almacenamiento de base de datos ASM como sistema de archivos.

Consideraciones de almacenamiento en RAC

Estas consideraciones son recogidas como recomendaciones de los casos más comunes reportados a Oracle Soporte:

- Asegurar el correcto uso de las opciones de montaje en discos NFS cuando se utiliza con RAC. Las opciones documentadas se encuentran detallada en

la siguiente ***Nota#359515.1*** - *Mount Options for Oracle files when used with NAS devices*, para cada plataforma.

- Implementar múltiples paths de acceso al almacenamiento usando dos o más HBAs o hacer uso de software de multi-pathing sobre estas HBAs. Donde sea posible, usar el pseudo controladores (multi-path I/O) Como por ejemplo son: EMC PowerPath, Veritas DMP, Sun Traffic Manager, Hitachi HDLM, IBM SDDPC, Linux 2.6 Device Mapper.

- Si se hace uso de ASM, seguir el siguiente apartado de ASM de este informe y la *12cR2 Automatic Storage Management Administrator's Guide:* https://docs.oracle.com/database/122/OSTMG/toc.htm

- Para información sobre best practices de migración a Oracle ASM desde entornos no-ASM, seguir la siguiente guía de MAA: http://www.oracle.com/technetwork/database/features/availability/maa-096107.html

Almacenamiento para ficheros de Clusterware y de BBDD

Los ficheros de Clusterware (OCR y Voting Disks) pueden ser almacenados en:

- 10gR2 y 11gR1:
  - raw/block devices compartidos
  - Un sistema de ficheros compartido o de cluster certificado:
    - Oracle Cluster File System (OCFS)
    - Oracle Cluster File System 2 (OCFS2)
    - General Parallel File System (GPFS) on POWER
    - NFS (no soportado en POWER ni IBM zSeries basado en Linux)
- 11gR2 y 12cR1:
  - ASM
  - Un sistema de ficheros compartido o de cluster certificado. La lista completa está disponible en los siguientes links:
    - RAC Technologies Matrix for Linux Platforms
    - http://www.oracle.com/technetwork/database/options/clustering/tech-generic-linux-new-086754.html
    - Oracle RAC Technologies Certification Matrix for UNIX Platforms
    - http://www.oracle.com/technetwork/database/options/clustering/tech-generic-linux-new-086754.html
    - Oracle RAC Technologies Certification Matrix for Microsoft Windows Platforms
    - http://www.oracle.com/technetwork/database/clustering/tech-generic-windows-new-166584.html
- 12cR2:

- Sólo en ASM

Las bases de datos Single-Instances pueden ser almacenadas en cualquier sistema de ficheros certificado, sea o no de cluster

Las bases de datos en RAC pueden ser almacenadas en:

- 10gR2 y 11gR1:
  - ASM
  - raw/block devices compartidos
  - Un sistema de ficheros compartido o de cluster certificado:
    - Oracle Cluster File System (OCFS)
    - Oracle Cluster File System 2 (OCFS2)
    - General Parallel File System (GPFS) on POWER
    - OSCP-Certified NAS Network File System (NFS)
- 11gR2, 12cR1 y 12cR2:
  - ASM
  - Un sistema de ficheros compartido o de cluster certificado. En este caso se recomienda Oracle ACFS. La lista completa está disponible en los siguientes links:
    - RAC Technologies Matrix for Linux Platforms
    - http://www.oracle.com/technetwork/database/options/clustering/tech-generic-linux-new-086754.html
    - Oracle RAC Technologies Certification Matrix for UNIX Platforms
    - http://www.oracle.com/technetwork/database/options/clustering/tech-generic-linux-new-086754.html
    - Oracle RAC Technologies Certification Matrix for Microsoft Windows Platforms
    - http://www.oracle.com/technetwork/database/clustering/tech-generic-windows-new-166584.html

Oracle ASM

Oracle ASM es un gestor de volúmenes y un filesystem para los archivos de base de datos Oracle tanto Single-Instance como RAC.

Oracle ASM es la solución de gestión del almacenamiento recomendada por Oracle que ofrece una alternativa a los gestores de volúmenes convencionales, sistemas de ficheros y raw/block devices.

Proporciona muchas de las mismas ventajas que las tecnologías de almacenamiento RAID o LVM.

A partir de Oracle Clusterware 12cR2, los ficheros de Clusterware (OCR y voting disk) sólo se pueden almacenar en Oracle ASM.

Beneficios de Oracle ASM:

- Permite crear un diskgroup a partir de una colección de dispositivos de disco individuales.
- Implementa striping y mirroring (a través de niveles de redundancia) para mejorar el rendimiento de I/O y la fiabilidad de los datos.
- Simplifica la administración de los ficheros de base de datos, manejando sólo los diskgroups.
- El rendimiento es comparable al rendimiento que ofrecen los raw devices.
- Se pueden añadir y quitar discos de un diskgroup en caliente, mientras que la base de datos continúa accediendo a los ficheros del diskgroup. Oracle ASM automáticamente re-distribuye el contenido de los archivos.
- Oracle ASM usa Oracle Managed Files (OMF).
- Oracle Automatic Storage Management Cluster File System (Oracle ACFS) es una tecnología multi-plataforma, filesystem escalable y de gestión del almacenamiento que extiende la funcionalidad de ASM para soportar ficheros que no son de base de datos.
- Striping:
    - Balancea las cargas de entrada y salida (I/O) entre todos los discos que pertenecen al mismo diskgroup.
    - Reduce la latencia de I/O.
- Mirroring - Niveles de Redundancia de ASM:
    - Externa: el contenido del diskgroup no es “mirroreado” por ASM, se delega a la cabina
        - Se necesita mínimo 1 disco en el diskgroup
    - Normal: el contenido del diskgroup es “two-way mirroring” por defecto
        - Se necesitan mínimo 2 discos en el diskgroup
        - Para los ficheros de Oracle Clusterware, la redundancia Normal provee 3 ficheros de voting disk, 1 de OCR y 2 copias (una primaria y una de espejo secundario). Con redundancia normal, el cluster puede sobrevivir a la pérdida de un grupo de fallo
        - Mínimo 3 discos en el diskgroup
    - Alta: el contenido del diskgroup es “three-way mirroring” por defecto
        - Se necesitan mínimo 3 discos en el diskgroup
        - Para los ficheros de Oracle Clusterware, la redundancia Alta provee 5 ficheros de voting disk, 1 de OCR y 3 copias (una primaria y dos de espejo secundario). Con

redundancia alta, el cluster puede sobrevivir a la pérdida de dos grupos de fallo

- Mínimo 5 discos en el diskgroup

Requerimientos/Recomendaciones para Oracle ASM

- Oracle recomienda utilizar un diskgroup de ASM en redundancia Normal o Alta con 3 o 5 discos respectivamente de 1GB para almacenar los ficheros de OCR y Voting Disks.
- Serán necesarios al menos 2 discos más para almacenar las bases de datos: uno para el diskgroup de DATOS y el otro para el FRA.
    - Recomendado un mínimo de 4 discos de igual tamaño y rendimiento para cada diskgroup.
- Se recomienda que todos los discos de Oracle ASM de un mismo diskgroup tengan similares características de rendimiento, almacenamiento y disponibilidad, ya que en configuraciones con discos de velocidad mixta (Ej: 10K y 15K RPM), el rendimiento de I/O estará limitado por la unidad de velocidad más baja.
- Se recomienda que todos los discos de Oracle ASM de un mismo diskgroup tengan la misma capacidad para mantener el balanceo.
- En Linux, usar la funcionalidad Oracle ASMLib para proveer consistencia de nombres de dispositivos y persistencia de permisos
- Oracle ASM Filter Driver (Oracle ASMFD) está disponible a partir de 12cR1 en Linux y a partir de 12cR2 en Solaris.
- Para proveer de redundancia al acceso al almacenamiento, se recomienda configurar como mínimo dos conexiones desde cada servidor al almacenamiento (multipath).
- Toda instalación de Oracle Grid Infrastructure 12cR2, ya sea Oracle Standalone Cluster o Oracle Domain Services Cluster, contiene un Grid Infrastructure Management Repository (GIMR), también denominado la Management Database (MGMTDB). Se recomienda crear un diskgroup dedicado para este repositorio con redundancia externa y los siguientes tamaños estimados:
    - Disk Space for 72hrs*
        - 12.1.0.2:
            - 5.2GB (<5 nodes)
            - 500MB each additional node
        - 12.2.0.1:
            - 36GB (<5 nodes)
            - 4.7GB each additional node
        - 12.2 DSC:
            - 188GB (<5 member clusters)
            - 35GB each additional cluster

Oracle Flexible Architecture

Optimal Flexible Architecture (OFA) , como se conoce a la guía de buenas prácticas de nomenclatura y configuración de base de datos Oracle, asegura instalaciones fiables y mejora la administración de las bases de datos. Estas recomendaciones están guiadas para optimizar la forma en que las instalaciones de software de Oracle se organizan, simplifica la gestión de instalaciones y mejora la capacidad de una buena administración ofreciendo las opciones de OFA por defecto, haciendo bases de datos Oracle que se instalen más en concordancia con las especificaciones de la OFA.

Durante la instalación, se le pide que especifique una ubicación para la ruta de (ORACLE_BASE), que tiene que ser propietaria del usuario que realiza la instalación. Puede elegir un ORACLE_BASE existente, o elegir otra ubicación dentro de un directorio que no tiene la estructura de un directorio ORACLE_BASE.

Utilizando la ruta del Oracle Base ayuda para facilitar la organización de las instalaciones de Oracle, y ayuda a garantizar que las instalaciones de varias bases de datos mantienen una configuración en concordancia con la OFA.

Durante la instalación, ORACLE_BASE es el único valor requerido como entrada, ya que el ORACLE_HOME es una ubicación secundaria dentro de la elegida sobre la para la ORACLE_BASE. Además, Oracle recomienda que se establezca la variable de entorno ORACLE_BASE además de la variable ORACLE_HOME, para ser cargadas en los perfiles de los usuarios que se dispongan a arrancar las bases de datos Oracle. Tenga en cuenta que ORACLE_BASE puede convertirse en una variable de entorno necesario para el inicio de base de datos en una futura versión.

Acceso a datafiles en Oracle RAC

Todas las instancias de Oracle RAC deben de ser capaces de acceder a todos los archivos de datos. Si un archivo de datos tiene que ser recuperado cuando la base de datos se encuentra abierta, entonces la primera instancia de Oracle RAC que arrancó es la que se encarga de realizar la recuperación y verificar el acceso al datafile. El resto de instancias que se encuentren arrancadas, también verifican que su acceso a los archivos de datos recuperados es correcto. Del mismo modo, cuando se añade un tablespace o un datafile o cuando se pone online un datafile o tablespace, todas las instancias comprueban el acceso al archivo o archivos.

Si se añade un archivo de datos a un disco que otras instancias no pueden tener acceso, entonces, la verificación falla. La verificación también falla si diferentes instancias intentan acceder a diferentes copias del mismo datafile. Si la verificación falla para cualquier instancia, es necesario diagnosticar y solucionar el problema. A continuación, ejecute la instrucción:

ALTER SYSTEM CHECK DATAFILES

Para comprobar el acceso a los archivos de datos en cada una de las instancias.

Almacenamiento de ficheros de Redo Log en Oracle RAC

Cada instancia tiene su propio grupo de redologs online. Es necesario crear estos grupos de redo log y establecer los miembros del grupo, como se describe en la guías de administración de base de datos Oracle. Para añadir un grupo de redolog a una instancia específica, se puede hacer a través del siguiente comando de sql:

ALTER DATABASE ADD LOGFILE

Indicando la instancia en concreto sobre la que se quiere realizar la operación. Si no se especifica la instancia al añadir el grupo de redo, el grupo de redolog es añadido a la instancia a la que se está conectado.

Cada instancia debe tener al menos dos grupos de redologs. Es necesario asignar el grupo de redolog antes de habilitar una nueva instancia. Cuando el grupo de redolog actual se llena, la instancia comienza a escribir al siguiente grupo de archivo de redo. Si la base de datos está en modo ARCHIVELOG, a continuación, cada instancia debe guardar cada grupo de redologs como ficheros de archivelog que son registrados contra el fichero de control.

Durante la recuperación de base de datos, todas las instancias habilitadas son verificadas para ver si la recuperación es necesaria. Si se elimina una instancia de su base de datos Oracle RAC, debe deshabilitar la instancia para evitar que esta no tenga que ser chequeado durante procesos de recuperación de la base de datos.

Gestión de Undo en Oracle RAC

Las bases de datos Oracle gestionan automáticamente los segmentos de undo específicos a un tablespace de undo que se asigna a una instancia. Sólo la instancia asignada al tablespace de undo puede modificar el contenido de ese tablespace. Sin embargo, todas las instancias siempre pueden leer todos los bloques de undo en un entorno de RAC en caso de que fuese necesario para coherencia de datos. Además, cualquier instancia puede actualizar cualquier tablespace de durante la recuperación de transacciones, mientras que el tablespace de undo no se está utilizando por otra instancia para generar undo o recuperar transacciones.

Se pueden asigna tablespaces de Undo en las bases de datos Oracle RAC especificando un valor diferente para el parámetro UNDO_TABLESPACE para cada instancia especificándolo en el Spfile.

No se puede, simultáneamente, hacer uso de gestión automática y manual de undo tablespaces en una base de datos Oracle RAC. En otras palabras, todas las instancias de una base de datos Oracle RAC deben aplicarse la misma política en el modo de gestión de espacio de Undo.

RECOMENDACIONES DE BACKUP & RECOVERY PARA ENTORNOS RAC

Esta sección explica cómo realizar un backup y restore de bases de datos en Oracle RAC haciendo uso de Recovery Manager (RMAN), así como el uso de best practices y pasos a tener en cuenta en este tipo de arquitecturas.

Esta sección también incluye la descripción como una instancia de RAC realiza una recuperación, parallel backup, recovery con SQL*Plus y uso de Flash Recovery Area en Oracle RAC.

Almacenamiento de Archivelogs para entornos RAC

En un entorno non-cluster filesystem, cada nodo puede tomar únicamente backup de sus propios archivelogs en local. Ya que no tienen acceso al resto de los nodos.

Como norma, no se permitirá este tipo de configuraciones y será obligatorio el acceso desde cualquier nodo del cluster a todos destinos de archivelogs creados por cualquier nodo.

Del párrafo anterior, queda como recomendación que solo se permitirá el uso de Cluster File System para el almacenamiento de archivelogs en entornos RAC. Teniendo en cuenta que no se hace uso de ASM. En el caso de que se utilice esta tecnología será el destino elegido para el almacenamiento de archivelogs.

Uso de RMAN para la creación de Backups en Oracle RAC

Como norma, se hará uso de la herramienta RMAN como software de backup de base de datos Oracle.

RMAN te permite la realización de backup, restores, recuperación de datafiles, ficheros de control, SPFILES y archivelogs.

RMAN es incluido con el software de Oracle Rdbms y por supuesto con el software de Oracle Real Application Cluster.

El procedimiento para usar RMAN en entornos RAC no difiere substancialmente del uso realizado para entornos single-instance.

A continuación nombraremos algunas de las características especiales para un Entorno RAC

- **Conexión de Canales para Instancias en Cluster**

  La conexión de los canales de RMAN para conectarse a una de las instancias está determinado por la cadena de conexión que se le especifique en la configuración del canal. Por ejemplo:

  CONFIGURE CHANNEL … user1/pwd1@service_name

En el ejemplo anterior, se especifica el service name como punto de entrada a la instancia que realizara el backup. En caso de que este service name tenga activado el balanceo de carga. Los canales serán alojados en el nodo que decida el algoritmo de balanceo de carga.

Como norma no se permitirá, el uso de este tipo de servicios para la alocación de canales. Sino que se definirá de manera explícita la instancia en el que el canal será alojado. Siguiendo el ejemplo que se muestra a continuación:

```
CONFIGURE DEVICE TYPE sbt PARALLELISM 3;
CONFIGURE CHANNEL 1.. CONNECT 'user1/pwd1@node1';
CONFIGURE CHANNEL 2.. CONNECT 'user2/pwd2@node2';
CONFIGURE CHANNEL 3.. CONNECT 'user3/pwd3@node3';
```

La norma anterior, no excluye el uso de varias instancias para la realización del backup. Como regla general se permitirá el uso de varias instancias para la alocación de canales que puedan ser usados en la realización del backup de la base de datos, si por motivos de rendimiento se encuentra una diferencia grande, se podrá permitir la alocación de los canales de backup en una única instancia.

La restricción se centra en el uso de servicios con load balancing quedan totalmente prohibidos para la alocación de canales.

Durante la realización del backup todas las instancias en las cuales se conecte un canal deben estar todas en el mismo estado o todas montadas todas abiertas. Por ejemplo, si en la instancia del nodo 1 se encuentra montada mientras en el nodo2 y nodo3 se encuentran abiertas el backup fallara.

- **Se permite la funcionalidad conocida como "Node Affinity Awareness of Fast Connections"**

  En alguna configuración de cluster, algunos nodos del cluster tienen un acceso más rápido a ciertos ficheros de datos que a otros. RMAN detectara automáticamente este comportamiento, también conocido como **Node Affinity Awareness.** Entonces cuando se decida a través de qué canal se hace backup de un datafile se delegara en RMAN para que haga uso de esta funcionalidad.

- **Borrado de Archived Redo Logs después de la ejecución de un Backup.**

  Como comentamos anteriormente en este documento, los archivelogs deben de estar visible desde cualquier instancia del cluster. Por lo que está permitido el borrado de archivelogs después de un backup de archives desde cualquiera de las instancias desde las que se ejecute. A no ser que se posea una configuración Data Guard y los archives aún no hayan sido aplicados por la/s Standby.

Restauraciones en entornos Oracle RAC

Para restauraciones y recuperaciones en entornos RAC, no es necesario configurar una instancia para realizar la configuración de Backup. En principio se aceptará cualquier instancia para la ejecución del Backup de la base de datos, ya que todos los datafiles están accesibles desde cualquier nodo.

En caso de fallo de una instancia que necesite recuperación debido a que por un fallo software o hardware han provocado la indisponibilidad de la instancia. La base de datos automáticamente usa los redologs online para ejecutar la recuperación. Por lo que no es necesario tener en cuenta ninguna funcionalidad especial.

- **Recuperaciones en Oracle RAC**

  Las recuperaciones deben ser a través de un cliente como RMAN, el procedimiento de recuperación de una base de datos a través de RMAN no difiere del procedimiento para un entorno single-instance.

  Habrá que tener en cuenta que el nodo desde que se ejecuta la recuperación debe estar disponible para restaurar todos los datafiles necesarios y disponible también para leer todos los ficheros de archived redologs en disco o desde backup.

- **Parallel Recovery en Oracle RAC**

  Como norma, se permite el uso de esta funcionalidad que permite a Oracle automáticamente seleccionar el grado óptimo de paralelismo para la recuperación de una instancia o la base de datos completa. El uso de parallel instance recovery se hace en tres fases de la recuperación

  - Restauración de datafiles.
  - Aplicación de Backup incrementales
  - Aplicación de archivelogs

  En el caso de que se desee deshabilitar el uso de paralelismo a la hora de recuperación de una instancia, se puede deshabilitar a través de modificar el siguiente parámetro a 0

  - *RECOVERY_PARALLELISM= 0.*

Uso de Fast Recovery Area en Oracle RAC

Para hacer uso de Fast Recovery Area (antes llamada flash recovery area) en Oracle RAC, la fast recovery area en adelante FRA, debe estar alojada en un disk group de ASM, en un Cluster FileSystem accesible desde cualquier instancia del Oracle RAC que se encuentre dentro del cluster.

Como norma, no se prohíbe el uso de la FRA, solo tener en cuenta los requisitos que en un entorno RAC necesita estar disponible desde cualquiera de los nodos del cluster. Para ello el parámetro DB_RECOVERY_FILE_DEST debe tener el mismo valor en todas las instancias.

RECOMENDACIONES DE UPGRADE Y APLICACIÓN DE PATCH EN ENTORNOS RAC

Esta sección está dirigida a tener en cuenta tanto los usuarios que deseen realizar una nueva instalación de Oracle Real Application Clusters, o para aquellos usuarios que ya se encuentran con una instalación de Oracle Real Application Cluster realizada y están pensando en implantar una estrategia proactiva de actualización de la instalación existente.

Consideraciones de Upgrade e Instalación de parches

Para nuevas instalaciones, se recomienda encarecidamente que se aplique el último patchset y PSU disponible para su versión y plataforma elegida.

En los casos en que la última versión de RDBMS no pueda ser utilizada debido a retrasos en la certificación o por liberación del software, esta soportado tener el CRS Home en el último nivel de patch que el software de Rdbms (ver sub-apartado *Certificación de CRS/RAC* del presente documento).

Como buenas prácticas desde Oracle soporte se recomienda tener en cuenta los siguientes puntos:

- La versión y release de Oracle Clusterware debe ser mayor o igual que la de Oracle ASM y Oracle Database que se ejecuten en el cluster
- La versión y release de Oracle ASM debe ser mayor o igual que la de Oracle Database que se ejecute en el cluster
- En caso de 11gR2 y superior, Oracle Clusterware y Oracle ASM serán siempre la misma versión ya que ambos se proveen con el mismo home de Grid Infrastructure
- Antes de parchear la base de datos o el clúster usando opatch chequear la disponibilidad de espacio libre y usar la documentación referenciada en la Nota#550522.1 disponible en MyOracleSupport (https://support.oracle.com)
- Para realizar el parcheo del software Oracle Grid Infrastructure, seguir la guía 12cR2 de Administración de Clusterware y el Readme del parche.
- Poner en marcha una estrategia proactiva, para fijar los últimos problemas conocidos a través de mantener actualizado el software con la aplicación de los Patch Set Updates (PSU) que son liberados de forma cuatrimestral por Oracle. Tener en cuenta las Notas de MOS 1360790.1 y 756671.1
- Si se va a instalar versiones Rdbms correspondientes a 10.2.0.4 y 11.1.0.7 PatchSet, Oracle recomienda encarecidamente aplicar el Patch#8199533 con el cual deshabilitiamos la configuración NUMA y evitamos todos los casos relacionados.

Rolling Upgrade

En primer lugar, debemos de definir el concepto de Rolling Upgrade, que se podría traducir como una actualización de forma gradual del software. El adjetivo de gradual es lo que nos diferencia del resto de actualizaciones y lo hace posible en entornos RAC, donde es posible ir actualizando instalaciones de forma independiente y progresiva.

En un rolling upgrade, nos referimos a la actualización del software (Oracle Database, Oracle Grid Infrastructure o el propio sistema operativo), mientras que el clúster está operativo se realiza el cierre de un nodo, la actualización del software en ese nodo, y luego lo reintegramos dentro del cluster, y así sucesivamente un nodo tras otro hasta que todos los nodos del clúster están en el nuevo nivel de parche del software.

Esta respuesta es para clusterwares ejecutados sobre Oracle Clusterware. Si el proveedor de clúster es un tercero, es necesario consultar con el proveedor acerca del soporte para el rolling upgrade en su software.

Para el software de base de datos Oracle Rdbms, es posible sólo para algunos parches individuales que están marcados como compatibles con rolling upgrade. La mayoría de bundle patch y Actualizaciones de Parches Críticos (CPUs), PSUs... son compatibles con rolling upgrade. Patchset y software de versión de base de datos 10g o 11g, las actualizaciones no son compatibles con rolling upgrade, una de las razones de que esto no pueda ser posible es que a través de las distintas versiones de Rdbms, puede haber versiones incompatibles de funcionalidades. Para actualizar este software haciendo uso de rolling upgrade es necesario el uso de una base de datos auxiliar con Logical Standby Oracle Database 10g o 11g, véase la nota#300479.1 para más detalles.

El software de clúster Oracle Clusterware es siempre compatible con rolling upgrade en todas sus versiones, mientras que si se usa el software ASM es compatible con rolling upgrade a partir de la versión 11.1.0.6 y superior.

A partir de Oracle 11g Release 2, los binarios para Oracle Clusterware y ASM se encuentran unidos en un único Oracle Home lo que se conoce como Grid Infrastructure Home. El Grid Infrastructure Home es compatible con rolling upgrade. El Oracle Clusterware y Oracle Real Application Clusters ambos soportan rolling upgrade de OS software cuando la versión de la Base de Datos Oracle está certificada en ambas versiones del sistema operativo (y el sistema operativo es el mismo, no Linux y Windows o AIX y Solaris, o 32 y 64 bits, etc.) Así que se puede aplicar un parche al sistema operativo, un conjunto de parches (como EL4u4 a EL4u6) o una release (EL4 a EL5).

RECOMENDACIONES DE PERFORMANCE Y TUNING PARA ENTORNOS RAC

Gran parte de los ajustes de rendimiento en RAC no es diferente de "instancia única". Sin embargo, hay algunas áreas adicionales para comprobar en un entorno RAC en los problemas de ajuste de rendimiento. A continuación se detallan algunas de las áreas donde el ajuste de rendimiento y diagnóstico son específicos para RAC.

Para administrar eficazmente el clúster de Oracle, más conocido como RAC, es importante saber cómo supervisar, solucionar problemas y recopilar los datos, tanto para tu propio diagnóstico, y también para proporcionar información a Oracle Soporte. A continuación se le dará información sobre la manera de hacer las dos cosas.

Caracterización de la carga de trabajo

Caracterización de la carga de trabajo a través de un análisis general de cómo la aplicación interactúa con la base de datos una buena práctica en las primeras etapas de ajuste de rendimiento de base de datos del sistema.

Esta sección no pretende describir todas las estadísticas específicas de RAC, sino más bien proporcionar algunas directrices sobre cómo iniciar el análisis del rendimiento.

El punto de partida tradicional consiste en obtener los informes de Statspack, o Automatic Workload Repository (AWR) si éste último se encentra licenciado, durante los períodos críticos de rendimiento.

Consideraciones de parámetros para performance tuning en RAC

PRE_PAGE_SGA: Modificar el parámetro PRE_PAGE_SGA a false. Si esta configurado a true, puede incrementar considerablemente el tiempo de establecer una conexión.

ACTIVE_INSTANCE_COUNT: En versiones 10g y 11g, el parámetro ACTIVE_INSTANCE_COUNT no debe ser modificado.

PARALLEL_EXECUTION_MESSAGE_SIZE: En versions anteriores a 11gR2, es recomendable configurar el parámetro PARALLEL_EXECUTION_MESSAGE_SIZE a 8192 que por defecto es 2048.

Directrices generales para entornos RAC

El punto más conflictivo a tener en cuenta en un entorno RAC, es el acceso eficiente a los datos, los puntos que se deben tener en cuenta son:

- Cuando se realizan insert, update o delete, de alta concurrencia en los pequeños conjuntos de dato puede afectar tiempos de respuesta.

Concurrente acceso de escritura a las filas del mismo bloque incremente los tiempos de acceso al bloque. Si la afinidad local de la caché, es decir, la retención y la disponibilidad de datos para los usuarios locales de una instancia disminuye ya que los bloques son compartidos con más frecuencia entre las instancias, la probabilidad de que la contención y la latencia de memoria repercutan en las transacciones. Las latencias aumentan cuando:

- Un bloque de caché local es solicitada por una instancia remota pero el proceso de liberación se aplazó porque hay transacciones activas pendientes en el bloque ("_GC_DEFER_TIME" referencia).
- El redo para los cambios recientes se debe escribir en el fichero de log antes de enviar el bloque a la otra instancia, por lo tanto requieren una operación de E/S y esperar a la notificación que en la operación fue realizada.

- Cuando varios bloques se analizan de forma secuencial, durante un full scan por ejemplo, o incluso en accesos no secuenciales, como en un index range scan, el número de bloques de accedidos y por lo tanto la posibilidad de que los datos no se encuentren en la caché local y requieren una lectura desde el disco o memoria caché puede aumentar la distancia. Esto puede agregar más latencias y sobrecarga de la CPU a una consulta.

Por lo que para las técnicas de performance tienen que tener en cuenta el diseño de la base de datos de acuerdo a sus características de acceso y que puedan ser optimizados.

Diferentes tamaños de bloque

Oracle permite tener diferentes tamaños de bloque de base de datos en la misma base de datos. Aprovechando esta característica se puede hacer una diferencia aun mayor en un entorno RAC que en un entorno de instancia única. Si se utiliza adecuadamente, el número de disco I / O y las operaciones pertinentes de GCS se deben reducir.

La recomendación general para la determinación del tamaño de bloque de la base de datos es el mismo que en single instance. Que siga las directrices documentadas en la guía *Oracle11g Database Performance Guide.*

Nuevos parámetros init.ora permite que el tamaño de estos depósitos separados para ser configurado individualmente, por ejemplo,

DB_2K_CACHE_SIZE

DB_4K_CACHE_SIZE

Tiene que garantizarse que una caché para un tamaño de bloque que no sea el tamaño de bloque por defecto sea definido para cada instancia.

Los diferentes tamaños de bloque son especialmente útiles cuando:

- Table scan son frecuentes, o se realizan muchas lecturas sobre los objetos. Las consultas tendrían que leer menos datos del disco o de la caché de la otra instancia, por lo tanto evitar sobrecarga y aumentar la tasa de aciertos de caché local.

- Cuando se realizan inserciones masivas de una gran cantidad de datos. En este caso, un tamaño de bloque más grande permite cargar más datos en un bloque antes de solicitar nuevo espacio. Esto aplica para entornos Datawarehouse y OLTP.

Ni que decir que la posibilidad de que los bloques sean de mayor tamaño durante accesos para la modificación desde varias instancias puede causar más concurrencia y penalización por lo que hay que considerar el cambio de esta parametrización en entornos mixtos que además de muchas cargas también realicen muchos accesos a datos.

Tamaños más pequeños del bloque deben ser tenidos en cuenta para entornos con una alta frecuencia de consulta o acceso a datos.

Puede ser difícil decidir qué tamaños de bloque elegir, a menos que haya un claro modelo de uso de la base de datos. Sin embargo, hay una serie de técnicas que podemos usar para elegir estos tamaños, como

- Tracear una aplicación típica con el evento 10046 y mirar el patrón de accesos.

- Analizar las estadísticas de objetos.

- Obtención de los planes de ejecución de las consultas SQL que se ejecutan con mayor frecuencia y teniendo en cuenta la E/S.

- Realizar consultas en V$BH o V$CACHE y comparar la distribución de los bloques. Por ejemplo, si hay una SQL que accede con frecuencia a 2 bloques adyacentes con 2KB de tamaño de bloque, debemos utilizar un tamaño de bloque de 4 KB o más para este objeto, de manera que las leemos en un solo acceso a disco.

Tenga en cuenta que una decisión sobre el tamaño del bloque correcto para evitar la contención es difícil, ya que no existe un método claro que nos proporcione el mejor tamaño. El objetivo primordial de una correcta elección de tamaño del bloque debe ser definido para reducir la E / S o transferencias globales de caché.

Con las últimas versiones de Oracle RAC cada vez es menos necesario ajustes específicos de tamaño de bloque, ya que se han incorporado más. Por ejemplo, Oracle11g introduce, entre otras características:

- Read-mostly locking
- Auto affinity
- Reader bypass locking

Que hace más efectiva la lectura y concurrencia inter-cache de RAC.

Monitorización y Tuning de rendimiento en RAC

Este apartado cubre como monitorizar y hacer tuning de rendimiento en Oracle Real Application Cluster.

- **Monitorización de Oracle Real Application Clusters y Oracle Clusterware**

  Como herramientas de monitorización se puede hacer uso de Oracle Enterprise Manager como método de monitorización de los entornos Oracle RAC y Oracle Clusterware.

  Desde cualquier ubicación con acceso al navegador web, en principio podremos administrar nuestras bases de datos Oracle en RAC. Principalmente se debe acceder a las siguientes páginas para observar el rendimiento de la base de datos

  - La página de Cluster Database
  - La página de Interconnects
  - La pagina de Cluster Performance

- **Verificando la configuración de Interconnect para Oracle RAC**

  El interconnect y la comunicación entre nodos puede afectar al rendimiento de Cache Fusión. Además, el ancho de banda de interconnect, y la latencia y la eficiencia del protocolo IPC determinara la velocidad con la cual la Cache Fusion procesa la transferencia de bloques.

  Para verificar la configuración de interconnect, realizar las siguientes consultas:

```
SQL> SELECT * FROM V$CLUSTER_INTERCONNECTS;

NAME IP_ADDRESS IS_ SOURCE
--------------- ---------------- --- ----------------------
eth2 10.137.20.181 NO Oracle Cluster Repository

SQL> SELECT * FROM V$CONFIGURED_INTERCONNECTS;

NAME IP_ADDRESS IS_ SOURCE
--------------- ---------------- --- ----------------------
eth2 10.137.20.181 NO Oracle Cluster Repository
eth0 10.137.8.225 YES Oracle Cluster Repository
```

  Una vez verificado que el interconnect está operativo, difícilmente se puede influir en su rendimiento sin embargo se puede influir en la eficiencia del protocolo de interconnect ajustando el tamaño de buffer de IPC.

- **Vistas de rendimiento en Oracle Real Application Clusters**

  Cada instancia tiene un conjunto de vistas específicas, con el prefijo v$ para monitorizar el rendimiento de una instancia, también conocidas como vistas dinámicas. En entornos RAC existen las vistas dinámicas globales que monitorizan el rendimiento desde todas las instancias, están pueden ser consultadas con el prefijo gv$

  Consultando una vista gv$ recibe la misma información que desde la v$ pero para todas las instancias que se encuentren disponibles en el cluster. Además cada vista GV$ contiene una columna extra con el nombre de INST_ID que indica la instancia a la que pertenece la fila correspondiente.

  Se puede usar la columna INST_ID como filtro para recibir la información de las v$ para un subconjunto de instancia, por ejemplo:

  *SQL> SELECT * FROM GV$LOCK WHERE INST_ID = 2 OR INST_ID = 5;*

Estadísticas de rendimiento en Oracle Real Application Clusters

En este apartado se proporciona información sobre las vistas v$ y GV$ que proporcionan las estadísticas que se pueden utilizar para evaluar las transferencias entre bloques en el clúster. Usar estas estadísticas para analizar las tasas de interconexión de transferencia en bloques, así como el rendimiento general de su base de datos Oracle RAC.

Oracle RAC contempla estadísticas donde aparecen contadores de solicitud de mensajes o estadísticas de tiempos entre los nodos del Cluster.

Las instantáneas de las estadísticas generadas por AWR y Statspack pueden ser evaluadas por la elaboración de informes que muestran los datos de resumen, tales como perfiles de carga y esperas en los puntos críticos recogidos en cada instancia. La mayor parte de los datos relevantes se resumen en el siguiente epígrafe como puntos de revisión de estadísticas en Oracle RAC:

- Global cache load profile
- Global cache efficiency percentages—workload characteristics
- Global cache and Enqueue Service (GES)—messaging statistics
- Global enqueue statistics
- Global CR statistics
- Global CURRENT served statistics
- Global cache transfer statistics.

Eventos de Global Cache

En un sistema de base de datos de multi-instancias donde los bloques de datos son compartidos a través de la caché distribuida para la lectura y para escritura, el acceso caché remota consumirá uso de CPU y tiempo de espera. Por lo que un grupo específico de eventos mantiene el seguimiento de tiempos de espera para las transferencias de caché a caché.

Algunos de los eventos esperas relacionados con RAC están en la categoría de "idle" events, es decir, que su aparición no supone ninguna sobrecarga para la base de datos y normalmente son relacionados a procesos de background como LMD o LMS.

- **GCS remote message**

  Cuando no hay solicitudes GCS encoladas para LMS en la cola de procesamiento o en su cola de recepción, el proceso LMS se irá a un estado de standby a esperar hasta que se publique una solicitud o hasta que se produzca un evento de espera.

- **GES remote message**

  El proceso LMD sólo se ocupa de los mensajes entrantes GES. Cuando no hay mensajes de LMD, se irá a dormir hasta que sea despertado o hasta que ocurra un timeout. Este comportamiento es análogo al de los LMS descrito anteriormente.

- **GC current/cr request:**

  Estos eventos de espera son relevantes sólo cuando una solicitud GC para un cr o current buffer está en curso. Actúan como marcadores de posición hasta que la solicitud se completa.

- **GC [current/cr] [23]-way**

  Un current o cr block se solicitó y fue recibió después de 2 o 3 saltos de red. La solicitud fue procesada inmediatamente, es decir, que no estaba ocupada o congestionada.

  El tiempo de ida y vuelta de peticiones se puede ver afectada por las latencias de red.

  Diagnóstico: Estos tiempos de espera son normalmente afectados por

  - El ancho de banda y la velocidad de transferencia de la red
  - La longitud del path code del protocol IPC
  - Sistemas saturados por: alto consume de CPU, alto número de procesos o longitud en las colas de ejecución.

  Si estos eventos consumen un alto tiempo de espera, el chequeo debería incluir:

  - Revisión de las latencias medias para gc current y gc cr blocks
  - Revisión de la saturación del ancho de banda de la red.
  - Revisión de la configuración del Private interconnect

- Revisión de los tamaños de los Socket buffer de envio y recibo.
- Revisión de la utilización de la CPU

Aplicaciones con altas esperas en estos eventos se caracterizan por:

- Altos accesos de lectura/escritura
- Alto número de lecturas en los bloques y las cabeceras de UNDO.

- **GC [current/cr] block busy**

Un bloque current o cr se solicitó y se recibió, pero no fue enviada inmediatamente por el LMS por alguna condición especial que retrasó el envío se encontró durante el proceso de liberación del bloque.

El tiempo de ida y vuelta de peticiones se puede ver afectada por las latencias de red.

Un tiempo de espera mayor de estos eventos es estrictamente indicativo de alta concurrencia y de la contención de bloques. Puede ser exacerbada por el aumento de los tiempos de E/S y carga total del sistema, o también por concurrencia de muchos de los procesos.

Es importante darse cuenta de que el tiempo predominante no se gasta en la transferencia de red o de envío y recepción de procesamiento del IPC, pero si en el proceso de liberación del bloque.

Diagnóstico: Este evento de espera se ve normalmente afectado por

- Block flush time (log file sync) para transferencias cr

Si estos eventos consumen un alto tiempo de espera, el chequeo debería incluir:

- Consultar la vista v$instance_cache_transfer para identificar instancias que estan contribuyendo significativamente a busy current o cr blocks.
- Revisar las latencias medias para realizar el flush de los bloques current (locales) y cr.
- Revisión del tiempo medio de pin con otras instancias.
- Revisar los tiempos de Log file sync y rendimiento del LGWR IO en las otras instancias del cluster.
- Revisar el rendimiento del DBWR en las otras instancias del cluster.

Aplicaciones con altas esperas en estos eventos se caracterizan por:

- Bloques calientes
- Alta concurrencia de updates en los mismos bloques. Tablas que son diseñadas como colas.
- Minimizar las latencias de LGWR IO,
- Evitar contención usando distribución de bloques, indices particionados por hash.

- **GC [current/cr] grant 2-way**

  Un bloque actual o cr se solicitó y se recibió un mensaje de concesión del bloque. La concesión fue dada sin retrasos significativos. Esto implica que el master del bloque no es el nodo local y que el bloque solicitado no se almacena en cache de ninguna otra instancia.

  El tiempo de ida y vuelta de peticiones se puede ver afectada por las latencias de red.

  Diagnóstico: Este evento de espera se ve normalmente afectado por

  - El ancho de banda y la velocidad de transferencia de la red.
  - Carga del sistema

  Si estos eventos consumen un alto tiempo de espera, el chequeo debería incluir:

  - Saturación en el ancho de banda de la red.
  - Revisar la configuración del Private interconnect
  - Revisar el uso de CPU

  Aplicaciones con altas esperas en estos eventos se caracterizan por:

  - Pocas transferencias intercache.
  - Altos ratios de IO
  - Revisar SQL y planes de acceso
  - Tuning de buffer cache.

- **GC current grant busy**

  Este evento de espera no debe ocurrir de forma normal y cualquier ocurrencia de el, debe ser reportado a desarrollo inmediatamente.

  Aplicaciones con altas esperas en estos eventos se caracterizan por:

- El tuning de aplicaciones con alto tiempo de esperas para este evento debe concentrarse en afinar los sistemas y la identificación de SQL y objetos. El caso más probable para un "busy current grant" se produce cuando varias sesiones en diferentes instancias leen el mismo bloque y un acceso exclusivo se solicita, es decir, en un acceso S se convierte a X. Estos escenarios pueden ocurrir en clusters donde la fragmentación de bloques de indice predomina.

- **GC [current/cr] block/grant congested**

  Un bloque actual o cr se solicitó y se recibió un mensaje de bloqueo o de conseción. La sugerencia de la congestión implica que la solicitud pasó más de 1 ms en las colas internas después de que la capa de ipc entregó el mensaje y antes de que el proceso LMS lo recogiera. Estos encolmientos pueden afectar a la latencia de la solicitud entre masters y esclavos.

  Diagnóstico: Este evento de espera se ve normalmente afectado por

  - La ejecución del proceso LMS es interumpida por procesoso con mayor prioridad.
  - LMS esta sobrecargado.
  - El sistema esta sobrecargado y se encuentra paginando o haciendo swapping.

  Si estos eventos consumen un alto tiempo de espera, el tuning se debería concentrar en la parte de S.O. y revisar el uso de memoria por parte de los procesos LMS.

- **GC cr failure**

  Un bloque de cr se solicitó y un estado de error se ha recibido, informando de que un bloque perdido se ha producido. El evento esta a veces acompañado de varios timeouts.

  Diagnóstico: Este evento de espera se ve normalmente afectado por

  - Perdida de bloques
  - Errores de Checksum
  - Errores de lectura de CR
  - Formato de bloque invalid o invalid scn dentro del bloque.

  Estos diagnósticos deberian incluir:

  - Trazas 10046 con level 8
  - Trazas 10708 con level 7

- Estadisticas de "gc blocks lost"
- Estadisticas de red "Netstat"

Las causas pueden ser múltiples, pero las más comunes son pérdida de bloques y los errores de checksum y podría ser causado por

- Fallos hardware en la red.
- La eliminación de paquetes por parte de los Switches.
- Fallos en el reensamblado de los paquetes.
- Overflow en los buffer de Comunicaciones.
- Problemas en el firmware o driver de red.

Cuando desbordamientos en los socket buffer son observados, incrementar el tamaño de UDP socket buffer recibido. Un valor de 256k suele ser suficiente. En otros casos, una actualización del firmware del adaptador de red y los drivers, deben ser considerados.

- **GC current retry**

Un bloque actual se pidió y un estado de error fue recibido, provocado por algún acontecimiento de carácter excepcional, como un bloque perdido.

Diagnóstico: Este evento de espera se debería resolver siguiendo el mismo patrón que gc cr failure

- **GC current/cr multi block request**

Una capa superior esta intentando leer un bloque contiguo o no contiguo y se pasa un vector de direcciones de bloque de la capa de GCS. Para las solicitudes de bloque contiguo, se realiza un intento de combinar varias solicitudes en un mensaje para consumir menos CPU.

El caché se espera a que todas las solicitudes se hayan completado. Se puede recibir conseciones o bloques. Debido a las limitaciones OSD para los búferes DMA, sólo 16 bloques contiguos pueden ser recibidos por una solicitud multibloque.

Una solicitud de cr contiguos generalmente es causada por un escaneo completo de tabla o índex full scan.

Normalmente, no hay problemas de rendimiento mayor con lecturas multibloque, a menos que los bloques se hayan perdido.

Diagnóstico: El diagnostic debería incluir trazas 10046 a nivel 8. Si la perdida de bloques esta causando problemas de rendimiento. Entonces

también es normal revisar la capa de S.O. para fijar problemas parecidos a los eventos de espera comentados anteriormente.

Aplicaciones con altas esperas en estos eventos se caracterizan por:

- SQL con (full table scans)
- Insercinoes de grandes volumenes de datos.

El punto de revision debe centrarse en el tuning sql y la optimización del espacio.

- **GC buffer busy**

Para algunas aplicaciones, el balanceo de carga de los usuarios hace que en una instancia local y otra remota puedan acceder con más frecuencia a un cierto rango de direcciones de bloques que se encuentran cacheadas por otras instnacias.

Si un bloque de datos o índice que se lee en la caché del búfer desde disco o desde otra instancia, los tiempos de pin pueden aumentar proporcionalmente con el tiempo que tarda en atender la solicitud de bloque. En resumen, las relaciones pueden resumirse en las siguientes formulas:

- **Pin time** = (time to read the block into cache) + (time to modify/process the buffer)
- **Busy time** = (average pin time) * (number of interested users waiting ahead of me)

En un entorno de base de datos en clúster, hay tres tipos de esperas que están relacionadas con global busy buffers, es decir, los bloques que tienen una solicitud pendiente de la IPC y para la cual los usuarios están esperando.

Un usuario en la misma instancia ha iniciado una operación remota sobre el mismo recurso y esta esperando ha que sea completada la operación en curso sobre este recurso o esta esperando a que el bloque que fue solicitado a otra instancia y todavía no ha sido liberado para ser usado por la instancia local.

Otras esperas importantes

- **CR request retry**

Cuando una solicitud de un bloque CR no llega antes de los mensajes de confirmación de envio, el bloque se considera perdido y la capa de buffer caché espera en este evento antes de volver a intentarlo. Esencialmente, este evento es un temporizador de sleep.

- **Enqueues**

Ya es conocido por todos los administradores de base de datos los "enqueues" pero cuando RAC se encuentra habilitado. La mayoría de las peticiones globales para los enqueues son asíncronas. Cualquier retraso en el RAC puede afectar a los tiempos de bloqueo de enqueues. La mayoría de las esperas por estos eventos son enqueues de los siguientes subtipos:

- TX: Transaction enqueue
- TM: TM enqueue
- HW: High-Water Mark enqueue
- SQ: Sequence enqueue.

En todos los casos anteriores, estas esperas son sincronas y pueden constituir un punto crítico de serialización.

- **Enqueue TX**

Existe cierto patrón de eventos de espera y estadísticas que son características de la serialización debido a la fragmentación de índices. Esta situación puede provocarse por la carga de trabajo y por la alta cantidad de datos insertados. Esta situación puede ser dramática en el caso de que el número de procesos que acceden a los índices de forma concurrente sea alta y puede convertir a los indices en verdaderos puntos calientes y por tanto, crear un problema de rendimiento.

Esta situación puede provocarse, entre otras causas, por:

- Índices que se incrementa de forma monótona según una clave, como por ejemplo, una secuencia
- Fragmentación de los bloques que contienen las hojas de los árboles B-Tree sobre los que se soportan los índices.
- Escasa profundidad del árbol del índice, que provoca que todos los accesos al arbol se realicen a través de la hoja raiz.

Estas situaciones puntos globales calientes sobre bloques de índices y la fragmentación de las hojas de los índices puede evitarse siguiendo las siguientes recomendaciones:

- Uso de particiones por hash o por rango.
- Si las claves son generadas a partir de una secuencia, incrementar la cache de la secuencia. Este punto se tratará con detalle en un apartado de este mismo documento.
- Añadir el número de la instancia de la base de datos a las claves.
- Uso de distribuciones uniformes o aleatorias para creación de las claves.

Para mejorar el diagnóstico,los eventos de espera TX enqueue han sido clasificados de la siguiente manera:

- enq: TX - row lock contention

  Esto significa que la session esta esperando por filas bloqueadas por otra session. Esta espera es provocada por la aplicación.

- enq: TX - index contention

  Este evento ocurre durante el split de un índice. En RAC, esta espera en conjunction con 'gc buffer busy', 'gc current block busy' y 'gc current split' puede indicar que hay una contención inter-instancia en los bloques de índices.

- enq: TX – contention

  Si este evento se encuentra en el top de los eventos de espera en un RAC, esto puede ser porque hay transaciones esperando por enqueues TX remotos.

  Si el número de instancias es mayor a 16, puede tener impactos en el comportamiento de la optimización de transaciones y puede verse afectado por el bug referenciado en Bug#6457594.

- enq: TX - allocate ITL entry

  La session esta esperando para que haya espacio en el bloque para realizar una transación sobre una fila que se encuentra en este bloque.

- **Enqueue HW**

  En ciertas ocasiones se presentan eventos de espera y estadísticas hinerentes a la actividad de la aplicación, en especial cuando ésta tiene como fundamental función la inserción de datos que crean nuevos bloques que a su vez son alojados en un segmento.

  En Oracle, la marca de agua o High Watermark (HWM) de un segmento es el puntero a un bloque de datos donde existen bloques formateados y libres para la inserción de un dato nuevo. Si el aplicativo inserta datos de forma intensiva, se debe crear nuevos bloques después de buscar en las freelists o en los bloques de tipo L1 y no encontrar espacio libre.

  Este proceso conlleva el formateo de los bloqueos, insertarlos en la cabecera del segmento y avanzar la marca de agua. Estas tareas se realizan miestras se mantiene un enqueue de tipo HWM de forma exclusiva en los nuevos bloques que se libera al completarse la operación.

  Los síntomas más comunes de esta situación son:

  - Un alto porcentage esperas por el enqueue *HWM*

- Un alto porcentage esperas por eventos *GC CURRENT GRANT*.

Estos síntomas están provocados por la serialización del uso de la HWM con la consecuente relentización de todo el proceso de creación de nuevos bloques. En un entorno basado en RAC, la duración de esta operación de gestión de espacio es proporcional al tiempo necesario para solicitar y adquirir los enqueue de tipo HWM más el tiempo necesario para adquirir bloqueos globales para todos los nuevos bloques. Este tiempo es normalmente corto, ya que en circunstancias normales no existen conflictos entre los nodos por los bloques nuevos.

Para encontrar la causa de este tipo de problemas, y determinar qué segmento es que esta creciendo más rápido, podemos optar bien por extrapolar las ratios de datos insertados a través de la vista V$SQLAREA o bien capturando las columnas ID1 y ID2 de los enqueue de tipo HWM, donde ID1 es el número del tablespace donde reside el bloque e ID2 es la dirección del bloque de la cabecera del segmento.

El estudio de las vistas V$ENQUEUE_STAT, V$SQLAREA y V$SESSION_WAIT así como los eventos 10046 y 10706 permiten la localización del segmento en cuestión.

Como se mencionó anteriormente, escenario donde este tipo de eventos de espera se producen es en aquellos donde la carga de datos sea una de las tareas fundamentales de la lógica de negocio, y por tanto la gestión de espacio debe acelerar en la medida de lo posible.

Por tanto, se realizan las siguientes recomendaciones en este sentido:

- Se recomienda definir un tamaño de extensión grande e uniforme para los segmentos manejados localmente y con gestión automática de espacio si estos son candidatos a grandes inserciones de datos.

- Si la concurrencia de procesos de instancias no locales es baja, se recomienda usar un tamaño de bloque grande para reducir la cantidad de creación de bloques nuevos.

- Si aplica, use grupos de *Freelists* para los bloques de datos.

- Es este caso, y si la concurrencia entre instancias es baja, usar el prealojo de extensiones y bloqueos usando funciones *HASH* para alinear los bloquoes con los límites de las extensiones.

- Tan solo recordar, que en caso de usar grupos de Freelists, puede provocarse la fragmentación de bloques en caso de que se eliminen filas o se añadan o elimen instancias a la base de datos y por tanto, los bloques de datos con afinidad a un nodo en especial pueden terminar en la *freelist* de otra instancia.

- Por tanto, de forma genérica se recomienda la gestión de espacio automática o *automatic segment space management* (*ASSM*) en la creación de tablespaces y en especial en bases de datos en *RAC*.

- **Library Cache Lock**

  En muchas aplicaciones ejecutando sql como cursors, triggers, paquetes un gran cantidad de tiempo es gastado esperando en library cache locks, principalmente en tablas y procedimientos. Para revisar este tipo de esperas y bloqueos lo mas aconsejable es comenzar a revisar la vista v$librarycache que contiene contadores de rendimiento en cada uno de los bloqueos específicos.

- **DFS lock handle / CI enqueue**

  Un evento de DFS lock handle puede esperar en un entorno en RAC bajo ciertas circunstancias, como cuando las llamadas cross-instancia son frecuentes o cuando su realización requiere una cantidad significativa de tiempo. Un ejemplo de ello, puede ser un checkpoint cuando tablas son borradas o truncadas.