Servicio Andaluz de Salud
CONSEJERÍA DE SALUD

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

TESTING DE SISTEMAS PARA ENTORNOS ORACLE REAL APPLICATION CLUSTER

Referencia documento: InfV5_JASAS_RAC_SystemTest_V920.doc
Fecha: 16 de noviembre de 2018
Versión: 9.2.0

Registro de Cambios

Fecha	Autor	Versión	Notas
14 de Abril de 2011	Jonathan Ortiz	2.2.	Versión inicial
14 de Julio de 2011	Jonathan Ortiz	2.3.	Versión actualizada
13 de Octubre de 2011	Jonathan Ortiz	2.4.	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
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 pruebas de Test comentadas por Oracle Soporte y planteadas como buenas prácticas de sistemas para administradores que hagan uso de *Oracle RDBMS* y *Oracle RDBMS Real Application Cluster (RAC)*. Aunque este documento se centra en la versión 12cR2, también aplica en su práctica totalidad a versiones anteriores, hasta la 10gR2.

Este conjunto de pruebas engloba a varias fases de vida de Oracle RAC, desde pruebas antes de la instalación del software, pruebas de validación de la instalación y pruebas de simulación de producción. Junto con ello se recogen un conjunto de pruebas de pérdidas de servicio de los diferentes componentes que forman parte de Oracle 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 casos 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 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 RAC* sobre los que se sostengan los sistemas y aplicaciones.

Objetivos de este documento

Antes de que una nueva máquina/cluster entre en funcionamiento en producción, es importante que se pruebe el sistema en profundidad para verificar que el comportamiento será el esperado. También es recomendable el testeo cuando se introducen cambios en el sistema, ya sean grandes o pequeños.

El objetivo de este documento es doble. En primer lugar, proporcionar una guía de pruebas para testear las respuestas de un entorno activo-activo con Oracle Real Application Cluster 12cR2, antes de pasar a producción, entre lo que se destaca:

- Verificación de la red que soportara el Clusterware de Oracle RAC.
- Verificar que el sistema ha sido correctamente instalado y configurado.
- Asegurar que el sistema sea capaz de alcanzar los objetivos esperados, particularmente de disponibilidad y rendimiento.

En segundo lugar, este documento puede servir como plantilla para realizar dichas pruebas y anotar los resultados obtenidos, verificando así la respuesta del sistema ante posibles fallos.

La configuración del sistema y procedimientos operacionales, deben también ser testeados para asegurar que los fallos de sus componentes y otros problemas pueden ser tratados de la manera más eficiente y con el menor impacto posible.

El propósito de este test es probar la robustez del sistema ante los distintos fallos.

Se recomienda que las pruebas sean ejecutadas bajo una carga de trabajo que simule al de un entorno de producción real, y en exclusividad de otras pruebas para poder medir eficientemente los resultados obtenidos.

Test de Red del Sistema

Debido a la importancia que tiene la red en un entorno de RAC se ha introducido este apartado donde se intenta recoger las consideraciones a nivel de red en un sistema que soporta RAC.

Un error común acerca de la red privada (interconexión del clúster) que utiliza un sistema RAC es que se utiliza solamente como mecanismo de heartbeat. Cuando una base de datos RAC se ejecuta en clúster, utiliza la interconexión del clúster (*interconnect*) para mantener la coherencia de caché y realizar operaciones de fusión de caché (*cache fusion*). En términos de red puro, bloques enteros de base de datos (definido por el tamaño de bloque de base de datos, 8k por defecto) multiplicado por el número de bloques accesibles en una sola lectura (definido por la base de datos como multi-block read count, por defecto 16) pueden potencialmente (y es probable que así sea) ser transferidos en cualquier momento a través de la red de interconexión.

Por ejemplo, en un clúster de 2 nodos que un usuario realiza una consulta que requiere un escaneo completo de una tabla Tabla_A en la instancia 1. Los bloques de base de datos para cumplir con esta consulta no están en la buffer cache en la instancia 1, pero si están en la buffer cache en la instancia 2. Dado que es menos costoso a partir de una perspectiva de rendimiento realizar la lectura desde la red que desde E/S, los bloques de la base de datos se extraen de la caché de la instancia 2 para cumplir con la solicitud de consulta. En este caso, estos bloques se transfieren en lecturas de 128 KB (suponiendo que el tamaño de bloque de base de datos de 8 KB y 16 multi-block read count) de tamaño.

Este ejemplo pone de manifiesto que es de vital importancia el tener configurado la red de interconexión de manera óptima.

Para que la red privada pueda soportar el tráfico de red generado por un cluster RAC mientras mantiene la estabilidad y el rendimiento del cluster, todos los componentes involucrados en la red deben trabajar en armonía, permitiendo que la red funcione sin errores y en su niveles óptimos y esperados de Gigabit Ethernet (GbE), 10 Gigabit Ethernet (10GbE) o Infiniband. Lo ideal sería que este control de rendimiento y estabilidad se llevara a cabo antes de instalar el software de RAC para reducir al mínimo la necesidad y la complejidad de los cambios de configuración una vez que el software de RAC está instalado.

Hay muchas utilidades que se pueden encontrar en Internet para poner a prueba el rendimiento de la red y sus tiempos de respuesta. Este documento se centrará en una utilidad llamada Netperf. Netperf es gratuito y puede ser compilado en prácticamente cualquier plataforma.

Netperf consta de 2 componentes, un componente del lado del cliente que proporciona el motor de las pruebas que se llevarán a cabo y otro componente del lado del servidor que simplemente de escucha y responde a las peticiones formuladas por el componente cliente. Algunas de las pruebas que son posibles con netperf son:

- TCP Stream Performance (this is the default test)
- UDP Stream Performance
- DLPI Connection Oriented Stream Performance
- DLPI Connectionless Stream Performance

- UNIX Domain Stream Socket Performance
- UNIX Domain Datagram Socket Performance
- Fore ATM API Stream Performance
- TCP Request/Response Performance
- UDP Request/Response Performance
- DLPI Connection Oriented Request/Response Performance
- DLPI Connectionless Request/Response Performance
- UNIX Domain Stream Request/Response Performance
- UNIX Domain Datagram Request/Response Performance
- Fore ATM API Stream Request/Response Performance

Los detalles sobre cada una de las pruebas disponibles enumeradas anteriormente se puede encontrar en el manual del usuario netperf que está situado en el sitio web netperf:

http://www.netperf.org/netperf/NetperfPage.html

El código fuente de Netperf está disponible para su descarga en el sitio web netperf.org, que recientemente ha sido migrado a GitHub.

Suponiendo una configuración por defecto para Oracle RAC utilizando TCP para la comunicación de clúster y UDP para el tráfico de RDBMS, se recomienda probar el ancho de banda y la latencia de solicitud /respuesta para los protocolos TCP y UDP a través de la interconexión privada. Las pruebas adicionales pueden incluir el rendimiento del throughput TCP y de solicitud/respuesta sobre la interfaz pública.

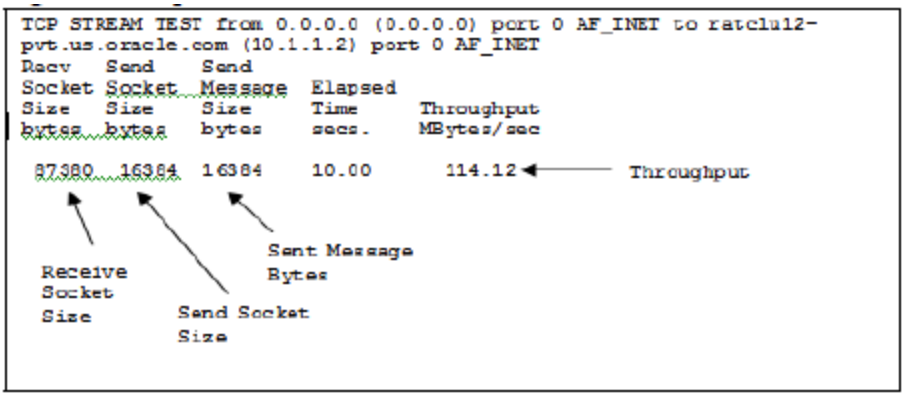

Gráfico – Netperf TCP Stream Test

La gráfica anterior muestra que la media de rendimiento de throughput TCP fue 114,12 MB por segundo. En Gigabit Ethernet el rendimiento más alto posible es de 1000 Mbit/s (125 MBytes por segundo), menos el overhead del TCP a través de Ethernet (~ 5 - 5,5%) y la latencia del cable (depende de la infraestructura), lo que significa que 114,12 MB por segundo está en consonancia con lo esperado con un buen rendimiento.

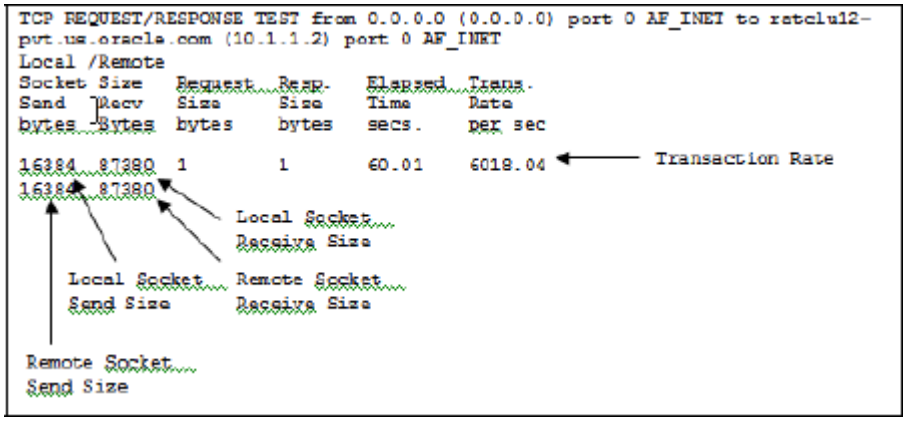

Gráfico – Netperf TCP Request/Response Test

En la gráfica anterior se muestra, la tasa de transacción de 6.018,04 por segundo para un mensaje de 1 byte. La división de un segundo por el ratio de transacción mostrará la latencia de ida y vuelta para cada byte que es de 167 microsegundos. De nuevo, esto está en consonancia con la latencia de espera de Gigabit Ethernet.

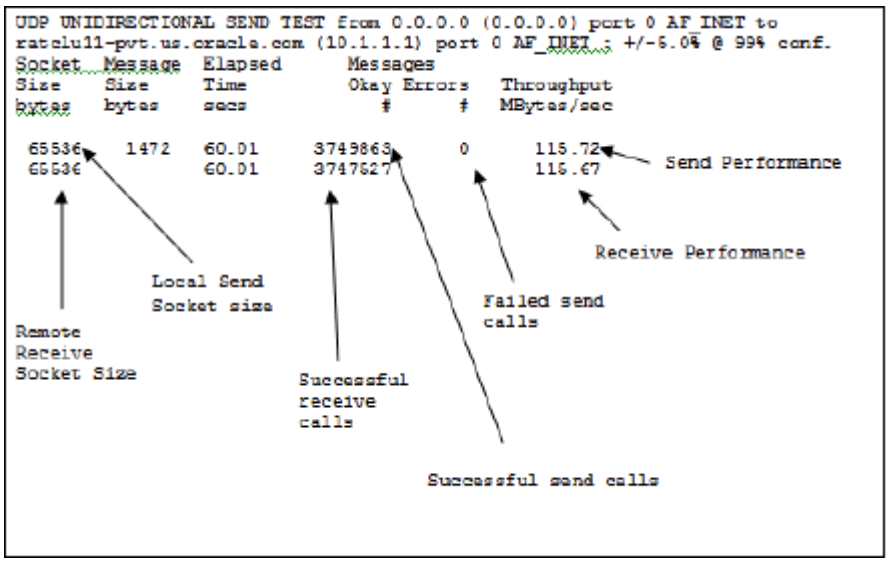

Gráfico – Netperf UDP Stream Test

La gráfica anterior muestra el promedio de throughput UDP fue 115,72 MB por segundo. El Gigabit Ethernet el rendimiento más alto posible es de 1000 Mbit/s (125 MBytes por segundo), menos los gastos de la UDP a través de Ethernet (~ 4,9 a 5,3%) y la latencia de red (dependiendo de la infraestructura), lo que significa que 115,72 MB por segundo está en consonancia con lo esperado de un buen rendimiento.

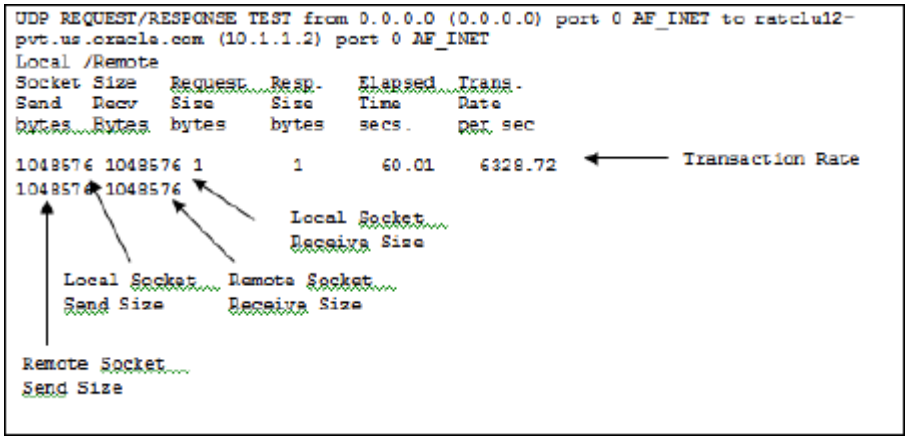

Gráfico – Netperf UDP Request/Response Test

La gráfica anterior muestra, la tasa de transacción UDP de 6.328,72 por segundo para un mensaje de 1 byte. La división de un segundo por el ratio de transacción mostrará la latencia de ida y vuelta para cada mensaje de 1 byte de 158 microsegundos. De nuevo, esto está en consonancia con la latencia de espera de Gigabit Ethernet.

El throughput y latencia de una red Gigabit se ven influidos por varios factores como la velocidad del bus PCI, la latencia del switch, la longitud del cableado, etc. Dicho esto, una red privada de interconexión implementada con las mejores best-práctices (switches redundantes y dedicados) debe alcanzar en torno al 95 % del ancho de banda anunciado como latencia en microsegundos en el rango de 150-200 para un mensaje de 1 byte. El éxito en pruebas de la red se mide de la siguiente manera:

- Si los resultados de la prueba (ancho de banda y latencia) caen dentro de los rangos esperados para GbE, 10GbE y/o Infiniband.
- Si se utiliza una configuración de NIC redundante tolerante a fallos, el fallo de un path tiene que ser cubierto por otro path sin interrupción del servicio.
- Si se usa una configuración de NIC redundante activa/activa, que el tráfico sea dirigido de acuerdo con las especificaciones de implementación activa/activa.
- Si las interfaces de red informan de errores o pérdida de paquetes durante las pruebas.
- Si se informan errores o problemas a nivel de protocolo de red (netstat -s).
- Si hay algún error de red reportado en cualquier log dentro de la tecnología de red (sistema operativo o Switch).

Es muy recomendable investigar y corregir cualquier problema potencial antes de comenzar la instalación del software del RAC. Este enfoque evita tener que realizar cambios de configuración en el software del RAC, debido a que generan cambios de configuración de red, así como reducir significativamente el posible problema de una instalación fallida.

Planning para los tests del sistema

Las pruebas de testing del sistema requieren una planificación cuidadosa para que sean efectivas. Los objetivos de nivel de servicio para el sistema en sí mismo y para las pruebas deben quedar claros y debe ser documentado el plan detallado de pruebas. La base para todas las pruebas es que las best-practices actuales, para la configuración del sistema en Oracle RAC, hayan sido llevadas a cabo antes de las mismas.

Las pruebas deben realizarse en un entorno que refleje el entorno de producción tanto como sea posible, lo ideal sería realizarlo en el mismo antes de la puesta en producción, pero por razones de coste, podría ser necesario utilizar un configuración de hardware reducida en algunas ocasiones. La configuración de software deben ser idénticas. Todas las pruebas serán realizadas durante la ejecución de una prueba de carga de trabajo lo más cercana a la producción como sea posible. Al planificar las pruebas del sistema es sumamente importante entender cómo la aplicación ha sido diseñada para manejar los fallos descritos en este plan y asegurar que los resultados previstos se ajustan al nivel de alta disponibilidad que soporta la aplicación, así como al nivel de alta disponibilidad de la base de datos Oracle.

Generar una realista carga de trabajo de aplicación puede ser complejo y costoso, pero es el factor más importante para la eficacia de las pruebas. Para cada prueba individual del plan, se requiere saber:

- Cuál es el objetivo de cada prueba y cómo se relaciona éste con los objetivos del sistema en general.
- Cómo se realiza exactamente cada prueba y cuáles son los pasos de ejecución.
- Cuáles son los criterios de éxito o fracaso y cuáles son los resultados esperados.
- Cómo se medirá el resultado de la prueba.
- Qué herramientas se utilizarán.
- Qué datos serán recogidos y cuáles son los ficheros de logs a revisar.
- Qué procedimientos operativos y procesos son relevantes.

Test de Stress del sistema/Simulación de un entorno de producción

La mejor manera de garantizar que el sistema funcionará bien sin ningún tipo de problemas es simular la carga de trabajo de producción y las condiciones de trabajo antes de publicarlo. Lo ideal sería que el sistema fuera estresado en algo más de lo que se espera en la producción. Además de ejecutar la carga de trabajo de aplicación, todos los procesos del S.O. también deben ser examinados al mismo tiempo, como los procedimientos de ejecución periódica. El resultado de las pruebas se debe mantener y ser comparado con los datos reales cuando se realice la puesta en producción. Las operaciones normales de mantenimiento, como agregar usuarios, añadir espacio en disco, la reorganización de tablas e índices, copias de seguridad, el archivado de datos, etc también se deben probar.

Test de fallos inducidos

La configuración del sistema y los procedimientos operativos deben ser evaluados para asegurarse de que los fallos de componentes y otros problemas pueden ser tratados con la mayor eficacia posible y con el mínimo impacto sobre la disponibilidad del sistema. Esta sección proporciona algunos ejemplos de las pruebas que se pueden utilizar como parte de un plan de pruebas del sistema. La idea es poner a prueba la robustez del sistema frente a diferentes fallos.

Este informe sólo incluye las pruebas para los componentes de RAC. Se necesitan pruebas adicionales para otras partes del sistema pero se salen del objetivo de este documento.

En algunos escenarios de fallo podría no ser posible recuperar el sistema dentro de un marco de tiempo aceptable y habría que especificar un plan de failover para cambiar a un sistema alternativo o a una ubicación alternativa. Esto también puede ser probado.

Test de funcionalidad de componentes

Normalmente no debería ser necesario realizar pruebas de funcionalidad adicionales para cada componente del software en RAC. Sin embargo, para algunos componentes pueden ser útiles realizar pruebas adicionales para asegurarse de que están configurados correctamente. Estas pruebas también ayudarán a los administradores de bases de datos a familiarizarse con los nuevos componentes de Oracle Rdbms 12cR2.

Cluster Infrastructure

Para simplificar las pruebas a menudo es muy útil hacer algunas pruebas básicas en la infraestructura de clúster sin el software de Oracle. Normalmente, esta prueba se llevará a cabo después de instalar el hardware y sistema operativo, pero antes de instalar cualquier software de Oracle. Normalmente algunas de las pruebas que se utilizaran son las siguientes:

- Fallo de un nodo. sin el software de Oracle instalado.
- Reinicio de un nodo fallido.
- Reiniciar todos los nodos a la vez.
- Pérdida de acceso a disco.
- HBA failover. Suponiendo que existen múltiples HBAs con capacidad de failover.
- Failover de la Controladora de Discos. Suponiendo que existen varios controladores de disco con capacidad de failover.
- Fallo de la/s NIC/s Pública/s.
- Fallo de la/s NIC/s de Interconexión.
- Error de almacenamiento NAS. En el caso de un fallo de espejo completo, medir el tiempo que se necesita para completar una reconfiguración de almacenamiento.

Si se utiliza un software de cluster de terceros:

- Simular un fallo en la red de Interconect.
- Simular una pérdida de acceso a los discos de quórum.

Descripción de los Tests

En los siguientes apartados se describen las baterías de pruebas a realizar sobre el entorno. Se debe tener en cuenta que estas pruebas están orientadas tanto a entorno RAC como single instance en activo-pasivo, por lo que se debe aplicar lo que corresponda dependiendo del tipo de entorno que se esté probando.

Los test se han clasificado en diferentes apartados atendiendo, bien a la funcionalidad que comprueban, bien al componente que interviene.

Los apartados en los que se han clasificado los test son los siguientes:

Grupo de Test	Descripción
Testing del Sistema: Escenarios de paradas	Este grupo de tests define una serie de escenarios de desastre que determinan un tipo de test a realizar. Se deben realizar los test que sean aplicables al entorno que se quiere validar, ya que no siempre los escenarios que se presentan son los que pueden ocurrir en el entorno, por las características del propio CPD, máquinas, etc.
Testing del Sistema: Fallos en los procesos del Clusterware	Este grupo de test se refiere a posibles fallos en los procesos del Clusterware.
Test de componentes: Tools de diagnóstico	Pruebas de que las herramientas de diagnóstico se ejecutan correctamente.

Testing del Sistema: Escenarios de paradas

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 1	Reinicio controlado del nodo	• Iniciar la carga de trabajo • Identificar las instancias con mayor número de conexiones de cliente • Reiniciar el nodo en el que se encuentre la instancia con mayor carga ○ Para Solaris: 'reboot'	• Las instancias y otros recursos del cluster que se estaban ejecutando en el nodo reiniciado (No aparecerá valor para el campo 'Server' desde la salida: crsctl stat res -t). • La node VIP se conmuta a un nodo superviviente y su estado se verá como "INTERMEDIATE" con un state_details a "FAILED_OVER". • Las SCAN VIP(s) que se estaban ejecutando en el nodo reiniciado conmutarán a nodos supervivientes. • Los SCAN Listener(s) que se estaban ejecutando en el nodo reiniciado conmutarán a nodos supervivientes. • Un instance recovery es realizado por otra instancia. • Los database services son movidos a instancias disponibles, si la instancia que se ha parado estaba especificada como preferred. • Las conexiones de cliente son movidas de forma transparente o reconectadas (dependiendo del tipo de conexión) a instancias supervivientes. • Las instancias supervivientes continúan procesando su carga de trabajo	• Tiempo para la detección del fallo del nodo o de la instancia • Tiempo en completar la recuperación de la instancia. • Tiempo en restaurar la actividad de los clientes al mismo nivel (suponiendo que los demás nodos tienen la capacidad suficiente para ejecutar la carga de trabajo) • Duración de la reconfiguración de la base de datos. • El éxito de conmutación de las SCAN VIPs y los SCAN listeners • El tiempo total hasta que la instancia del nodo reiniciado sea iniciada de nuevo automáticamente por Clusterware y acepte nuevas conexiones	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 2	Fallo no planificado del nodo del OCR Master	• Iniciar la carga de trabajo • Identificar el nodo donde se encuentra el OCR Master con el siguiente commando: `grep -i "OCR MASTER" $CRS_HOME/log/<node_name>/crsd/crsd.l*` • Apagar el nodo que contiene el OCR Master.	Igual que en el test 1	Igual que en el test 1	
Test 3	Rebotar todos los nodos al mismo tiempo.	• Realizar un reboot en todos los nodos al mismo tiempo o Para Solaris: 'reboot'	• Todos los nodos, instancias deben ser reiniciados sin problemas.	• Tiempo para que todos los recursos vuelvan a estar disponibles. Chequearlo con "crsctl stat res –t".	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 4	Fallo de una instancia	• Iniciar la carga de trabajo • Identificar que instancia tiene mayor número de conexiones: o Para Solaris: # ps –ef | grep pmon o kill del proceso pmon: # kill –9 <pmon pid>	• Una de las otras instancias realizará el instance recovery. • Los servicios son movidos a las instancias disponibles si la instancia preferred es la que falla. • Las conexiones de clientes son movidas de forma transparente o reconectadas a las instancias supervivientes (dependiendo de la configuración se tendrá un comportamiento u otro) • Después de la parada, el resto de instancias continuaran con la carga de trabajo. • La instancia fallida será reiniciada por Oracle Clusterware, a menos que esta opción este deshabilitada.	• Tiempo para la detección del fallo de la instancia • Tiempo para completar la recuperación de la instancia (revisar el alert) • Tiempo para restaurar la actividad de los clientes al mismo nivel (suponiendo que los demás nodos tienen la capacidad suficiente para ejecutar la carga de trabajo) • Duración de la reconfiguración de la base de datos en el failover. • El tiempo total hasta que la instancia fallida sea iniciada de nuevo automáticamente por Clusterware y acepte nuevas conexiones.	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 5	Parada forzosa de una instancia	• Usar un 'shutdown abort'	• Una de las otras instancias realizará el instance recovery. • Los servicios son movidos a las instancias disponibles si la instancia preferred es la que falla. • Las conexiones de clientes son movidas de forma transparente o reconectadas a las instancias supervivientes (dependiendo de la configuración se tendrá un comportamiento u otro) • Después de la parada, el resto de instancias continuarán con la carga de trabajo. • La instancia fallida **NO** será reiniciada por Clusterware, debido a que el usuario invocó un shutdown.	• Tiempo para la detección del fallo de la instancia • Tiempo para completar la recuperación de la instancia (revisar el alert) • Tiempo para restaurar la actividad de los clientes al mismo nivel (suponiendo que los demás nodos tienen la capacidad suficiente para ejecutar la carga de trabajo) • Duración de la reconfiguración de la base de datos en el failover.	
Test 6	Reinicio de una Instancia que ha fallado	• Reinicio automático por Oracle Clusterware si se trata de una fallo no controlada • Reinicio manual necesario si se emitió un comando de "shutdown" • Reinicio manual cuando se deshabilitó la opción "Auto Start" para la instancia relacionada	• La instancia se vuelve unir al cluster de RAC sin ningún problema. • Las conexiones de clientes y la carga de trabajo serán balanceadas a las otras instancias.	• Tiempo hasta que los servicios y la carga de trabajo sea rebalanceada al resto de instancias.	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 7	Fallo múltiple de instancias.	• Iniciar la carga de trabajo • Identificar dos instancias de la misma bbdd: o Para Solaris: # ps –ef | grep pmon o kill de los procesos pmon: # kill –9 <pmon pid>	• Mismos que el Test 4 para ambas instancias	• Mismas que el Test 4	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 8	Fallo de la instancia de ASM	• Iniciar la carga de trabajo • Identificar el proceso de una de las instancias de ASM: o Para Solaris: # ps –ef | grep pmon o kill de los procesos pmon: # kill –9 <pmon pid>	• Los recursos *.dg, *.acfs, *.asm y *.db del nodo con el fallo en la instancia de ASM pasarán a offline (crsctl stat res –t). Estos recursos serán automáticamente reiniciados por Oracle Clusterware. • Una de las otras instancias realizará el instance recovery. • Los servicios son movidos a las instancias disponibles si la instancia preferred es la que falla. • Las conexiones de clientes son movidas de forma transparente o reconectadas a las instancias supervivientes (dependiendo de la configuración se tendrá un comportamiento u otro) • Después de la parada, el resto de instancias continuarán con la carga de trabajo. • El Clusterware alert log mostrará que el crsd pasa offline debido a que el OCR está inaccesible. El CRSD se reiniciará automáticamente	• Tiempo en detectar el fallo de la instancia • Tiempo para completar la recuperación de la instancia (revisar el alert) • Tiempo para restaurar la actividad de los clientes al mismo nivel (suponiendo que los demás nodos tienen la capacidad suficiente para ejecutar la carga de trabajo) • Duración de la reconfiguración de la base de datos en el failover. • El tiempo total hasta que la instancia de ASM fallida y la de bbdd dependiente sea iniciada de nuevo automáticamente por Clusterware y acepte nuevas conexiones.	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 9	Fallo del Listener	• Para Solaris: ○ Obtener el PID del proceso del listener: # ps –ef | grep tnslsnr ○ Kill del listener: # kill –9 <listener pid>	• No impacta en las sesiones ya conectadas a la base de datos. • Nuevas conexiones son redirigidas al listener de otro nodo (dependiendo de la configuración del cliente) • La instancia local de bbdd no recibirá nuevas conexiones. • El fallo del listener es detectado por el ORAAGENT y Clusterware lo reiniciará automáticamente, Revisar crsd.log y oraagent_<GIowner>.log	• Tiempo para que el clusterware detecte el fallo del listener y lo reinicie.	
Test 10	Fallo del SCAN Listener	• Para Solaris: ○ Obtener el PID del proceso del SCAN listener: # ps -ef | grep tnslsnr ○ Kill del SCAN listener: # kill -9 <SCANlistenerpid>	• No impacta en las sesiones ya conectadas a la base de datos. • Las nuevas conexiones son redirigidas a los otros SCAN Listeners supervivientes • El fallo del listener es detectado por el CRSD ORAAGENT y Clusterware lo reiniciará automáticamente, Revisar crsd.log y oraagent_<GIowner>.log	• Tiempo para que el clusterware detecte el fallo del SCAN listener y lo reinicie.	
Test 11	Fallo de la Red Pública	• Desconectar todos los cables de red de la red pública. **Nota:** No se recomienda usar ifconfig para parar la interfaz de red.	• Chequear con "crsctl stat res –t" ○ El ora.*.network y los recursos de listener son parados en ese nodo. ○ Los SCAN VIP y Listener realizarán failover para el nodo superviviente. • La instancia de base de datos permanecerá arrancada pero será des-registrada de los remote listeners. • Los servicios de base de datos realizarán failover a los nodos supervivientes.	• Tiempo en detectar el fallo de red y realojar los recursos.	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 12	Fallo de una NIC de la Red Pública	• Asumiendo que interfaces duplicadas son configuradas para la redundancia (bonding, teaming, etc) • Desconectar el cable de red de una de las interfaces de la red pública. **Nota:** No se recomienda usar ifconfig para parar la interfaz de red.	• El trafico de red debería realizar failover hacia la otra interfaz de red pública sin impactar en los recursos del clusterware.	• Tiempo para realizar el failover sobre la otra tarjeta de red. Con bonding /teaming configurado este debería ser menor de 100ms.	

Test 13	Fallo de la red de Interconnect en 11.2.0.2 y superior. El método de node-eviction fue cambiado a partir de esta versión con la introducción del "Reboot less Restart" o "Reboot less node fencing"	• Desconectar todos los cables de red de la red de interconnect.
		
		Nota: No se recomienda usar ifconfig para parar la interfaz de red.
			• CSSD detectará la situación de split-brain y el que arrancó en primer lugar sobrevivirá)
			• Se realizará un split-brain y el nodo con el fallo en la interconnect realizará el node eviction "reboot less node fencing": realizará un reinicio ordenado de todo el stack de Clusterware. Si esto no recupera el estado, el nodo será reiniciado
			• Anterior a 11.2.0.2, el nodo donde se realiza el node eviction, será reiniciado
			• Revisar los logs:
			`$CRS_HOME/log/<nodename>/cssd/ocssd.log`
			`$CRS_HOME/log/<nodename>/alert<nodename>.log`
				• Tiempo en detectar el split brain e iniciar el desacople del nodo.
				• Ver las medidas del Test de fallo del nodo.	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 14	Fallo de una NIC de la red de Interconnect en 11.2.0.2 y superior con la redundancia de interconnect HAIP	• Asumiendo que la red de Interconnect tiene interfaces redundantes y éstas están configuradas con la tecnología HAIP embebida de Oracle Clusterware 11.2.0.2 y superior • Desconectar el cable de red de una de las interfaces de la red de interconnect. **Nota:** No se recomienda usar ifconfig para parar la interfaz de red.	• El HAIP hará failover a otra de las NICs supervivientes de la configuración. • Clusterware y las bases de datos en RAC no se verán afectadas • Cuando el cable de red vuelva a conectarse, HAIP hará de nuevo el failover a la situación original	• Tiempo en realizar el failover sobre la otra tarjeta de red.	
Test 15	Fallo del Switch de Interconnect (En configuraciones con redundancia de Switches)	• En una red con configuración de switches redundantes, apagar un switch.	• El tráfico de red debería realizar failover sobre el otro switch sin impactar en la interconnect ni en los recursos del Clusterware ni instancias.	• Tiempo en realizar el failover	
Test 16	Pérdida de acceso desde un nodo al sistema de discos desde un Path (OCR, Voting Disk o Datafiles)	• Desconectar el cable de conexión al almacenamiento externo (SCSI, FC or LAN cable) desde el nodo al sistema de discos.	• Si multi-pathing está configurado, el multi-pathing debería proporcionar transparencia al fallo y ningún recurso o instancia se verá afectado • Si está en single-pathing y Clusterware 11.2.0.2 o superior, se procederá con un node eviction "reboot less node fencing" y si esto no recupera el estado, el nodo será reiniciado • Si está en single-pathing y Clusterware previo a 11.2.0.2, el nodo será reiniciado	• El Path failover debería ser visible a nivel de logs del sistema operativo y transparente para recursos e instancias. • En caso de un node eviction tipo pre o post 11.2.0.2, tomar el tiempo en finalizar	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 17	Añadir un nodo al RAC	• Seguir el procedimiento indicado en Oracle® Real Application Clusters Administration and Deployment Guide 12c Release 2 Capítulos 11 ó 10 para añadir un nodo al cluster. • Verificar que has añadido el nodo del cluster correctamente	• El Nuevo nodo es añadido satisfactoriamente al cluster. • La nueva instancia de la base de datos comenzara a dar servicio y aceptar peticiones de clientes. • Si la base de datos es policy managed, una instancia de esta base de datos será creada e iniciada automáticamente en este nodo • Si la base de datos es admin managed, hay que extender la base de datos	• Tiempo en añadir el Nuevo nodo al cluster • Tiempo en extender la base de datos al nuevo nodo dependiendo de su policy • Tiempo en que empieza a dar servicio a nuevas conexiones	
Test 18	Eliminar un nodo del RAC	• Seguir el procedimiento indicado en Oracle® Real Application Clusters Administration and Deployment Guide 12c Release 2 Capítulo 11 para eliminar un nodo del cluster. • Verificar que has eliminado el nodo del cluster correctamente	• Las conexiones a la base de datos que se encuentran en la instancia alojada en ese nodo son repartidas por failover al resto de instancias supervivientes. • El nodo es eliminado del cluster de forma correcta.	• Tiempo en eliminar el nodo del cluster.	
Test 19	Pérdida de un disco de ASM	• Asumiendo una redundancia Normal • Hacer el Power off / pull out / offline (dependiendo de la configuración) en un disco de ASM	• Ningún impacto en recursos, servicios ni instancias • ASM comenzará el rebalanceo hacia los otros discos de ASM del mismo DiskGroup (ver alert.log de ASM)	• Monitorizar el proceso: select * from v$asm_operation;	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 20		• Hacer el Power on / insert / online (dependiendo de la configuración) del disco de ASM	• Ningún impacto en recursos, servicios ni instancias • ASM comenzará el rebalanceo hacia los otros discos de ASM del mismo DiskGroup (ver alert.log de ASM)	• Monitorizar el proceso: select * from v$asm_operation;	

Testing del Sistema: Fallos en los procesos del Clusterware

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas

Test 21	Fallo del proceso: CRSD	• Para Solaris:
		o Obtener el PID para el proceso CRSD:
		# ps –ef | grep crsd
		o Matar al proceso CRSD:
		# kill –9 <crsd pid>
			• El fallo del proceso CRSD es detectado por el orarootagent y será reiniciado. Revisar:
			`$GI_HOME/log/<nodename>/crsd/crsd.log`
			`$GI_HOME/log/<nodename>/agent/ohasd/orarootagent_root/orarootagent_root.log`
				• Tiempo en reiniciar el proceso CRSD	

Test 22	Fallo del proceso: EVMD	• Para Solaris:
		o Obtener el PID para el proceso EVMD:
		# ps –ef | grep evmd
		o Matar al proceso EVMD:
		# kill –9 <evmd pid>
			• El fallo del proceso EVMD es detectado y será reiniciado. Revisar:
			`$CRS_HOME/log/<nodename>/evmd/evmd.log`
			`$GI_HOME/log/<nodename>/agent/ohasd/oraagent_grid/oraagent_grid.log`
				• Tiempo en reiniciar el proceso EVMD	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 23	Fallo del proceso: CSSD	• Para Solaris: o Obtener el PID para el proceso CSSD: # ps –ef | grep cssd o Matar al proceso CSSD: # kill –9 <cssd pid>	• El nodo será reiniciado. • Se realizara una reconfiguración del Cluster.	• Tiempo en el que se detecta el fallo y la reconfiguración de los nodos supervivientes. • Tiempo en el que el nodo que falla vuelve a estar online.	

Test 24	Fallo del proceso: CRSD ORAAGENT RDBMS	• Obtener el PID del proceso CRSD ORAAGENT RDBMS:
		# cat $GI_HOME/log/<nodename>/agent/crsd/oraag ent_<rdbms_owner>/oraagent_<rdbms_owner >.pid
		• Matar al proceso:
		# kill –9 <pid>
			• El fallo del proceso es detectado y será reiniciado. Revisar:
			`$GI_HOME/log/<nodename>/crsd/crsd.log`
			`$GI_HOME/log/<nodename>/agent/ohasd/oraagent_grid/oraagent_grid.log`
			$GI_HOME/log/<nodename>/agent/crsd/oraag ent_<rdbms_owner>/oraagent_<rdbms_owner>.log
				• Tiempo de reinicio del proceso	

Test 25	Fallo del proceso: CRSD ORAAGENT GI	• Obtener el PID del proceso CRSD ORAAGENT GI:
		# cat $GI_HOME/log/<nodename>/agent/crsd/oraag ent_<GI_owner>/oraagent_<GI_owner>.pid
		• Matar al proceso:
		# kill –9 <pid>
			• El fallo del proceso es detectado y será reiniciado. Revisar:
			`$GI_HOME/log/<nodename>/crsd/crsd.log`
			`$GI_HOME/log/<nodename>/agent/ohasd/oraagent_grid/oraagent_grid.log`
			$GI_HOME/log/<nodename>/agent/crsd/oraag ent_<GI_owner>/oraagent_<GI_owner>.log
				• Tiempo de reinicio del proceso	

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 26	Se pueden hacer así sucesivamente y uno a uno con el resto de procesos: CRSD ORAROOTAGENT, OHASD ORAAGENT, OHASD ORAROOTAGENT, CSSDAGENT y CSSMONITOR	• Obtener el PID del proceso • Matar al proceso: # kill –9 <pid>	• El fallo del proceso en cuestión es detectado y será reiniciado.	• Tiempo de reinicio de cada proceso	

Test de componentes: Tools de diagnóstico

Test ID	Descripción	Procedimiento	Resultados esperados	Medidas	Resultados obtenidos/Notas
Test 27	Procedimientos de diagnóstico	• Iniciar la carga de trabajo • Ejecutar los procedimientos para recopilar la información de diagnóstico desde las herramientas: TFA, hanganalyze, racdiag.sql, ORAchk, oswatcher...(ver siguiente apartado)	• Las herramientas de diagnóstico se han ejecutado correctamente. • El tiempo de ejecución de estas herramientas es aceptable.		

Herramientas de Diagnóstico para RAC

Oracle proporciona muchas herramientas de diagnóstico para una base de datos Oracle: hanganalyze, racdiag.sql, OSWatcher, Procwatcher, systemstate...

Ante un error o problema, hay que ejecutar las herramientas correctas, en el momento correcto. Si además se usa Oracle Clusterware, hay que recopilar los datos de todos los nodos de la base de datos. Es posible que se necesite usar muchas de estas herramientas diferentes que solo usa una u otra vez y cada una tiene su propia sintaxis.

Una vez que se ha logrado obtener todos estos datos, pueden ser masivos y sólo una fracción de lo que se ha recopilado suele ser útil. Esto si se ha podido todo, si se ha sido lo suficientemente rápido antes de que se sobrescriba...

Mientras tanto, todavía se tiene el problema y aún necesita repararlo.

Aquí es donde entra Oracle Trace File Analyzer (TFA) con el Database Support Tools Bundle. Oracle Trace File Analyzer ayuda a realizar una monitorización del estado en tiempo real, detección de fallas y diagnóstico a través de una única interfaz. Consolidará de forma segura todos los datos de diagnóstico distribuidos.

Está continuamente disponible y observa los registros en busca de problemas significativos que pueden afectar su servicio. Si lo desea, también puede recopilar automáticamente los datos de diagnósticos pertinentes cuando vea estos problemas.

Oracle Trace File Analyzer sabe lo que es relevante en los ficheros de log. Esto le permite recortarlos al tamaño más pequeño y aun así reunir todo lo necesario. También recopila datos entre los nodos del clúster y los consolida todo en un solo lugar. Una vez hecho esto, puede incluso subir los datos recopilados automáticamente a Oracle Support.

La instalación de Oracle Grid Infrastructure desde las versiones 11.2.0.4 y 12.1.0.2 incluye embebida una instalación de TFA. Sin embargo, esta instalación de TFA no incluye muchas de las herramientas de base de datos.

Se puede usar Oracle TFA con todas las versiones soportadas de Oracle Database y Clusterware.

Funciona en los mismos Sistemas Operativos y versiones soportados por la base de datos.

¿Qué puede hacer TFA?

- TFA le ofrece una interfaz para todas sus necesidades de diagnóstico.
- Obtiene todos los diagnósticos correctos en el momento correcto.
- Puede recopilar datos en su clúster y reunirlos en un solo lugar.
- Automatic Diagnostic Collection
- On-demand Analysis and Collection
- Analiza los logs y busca errores
- Puede hacer Data Masking de los datos recolectados

Herramientas incluidas en TFA con el Database Support Tools Bundle:

- ORAchk: Oracle Stack Health Checks on non-engineered systems. See 1268927.2 for more details.
- EXAchk: Oracle Stack Health Checks on Engineered Systems. See 1070954.1 for more details.
- OSWatcher: Collect and archive OS metrics, useful for instance / node evictions & performance Issues. See 301137.1 for more details.
- procwatcher: Automate & capture database performance diagnostics & session level hangs. See 459694.1 for more details.
- oratop: Near real-time database monitoring. See 1500864.1 for more details.
- alertsummary: Provides summary of events for one or more database or ASM alert files from all nodes
- ls / dir : Lists all files TFA knows about for a given file name pattern across all nodes
- pstack: Generate process stack for specified processes across all nodes
- grep / findstr : Search alert or trace files with a given database and file name pattern, for a search string
- summary: High level summary of the configuration
- vi / notepad : Open alert or trace files for viewing a given database and file name pattern in the vi editor
- tail: Run a tail on an alert or trace files for a given database and file name pattern
- param: Show all database and OS parameters that match a specified pattern
- dbglevel: Set and unset multiple CRS trace levels with one command
- history: Show the shell history for the tfactl shell
- changes: Report any noted changes in the system setup over a given time period. This includes database a parameters, OS parameters, patches applied etc
- calog: Reports major events from the Cluster Event log
- events: Reports warnings and errors seen in the logs
- managelogs: Shows disk space usage and purges ADR log and trace files
- ps / tasklist : Finds processes
- triage: Summarize oswatcher/exawatcher data

Cada una de estas tools pueden ser ejecutadas usando el comando de TFA tfactl en modo shell, ocultando la complejidad de cada una de ellas al proporcionar una única interfaz y sintaxis.

Oracle además, ahora en la versión 12cR2, ha sacado el Oracle Autonomous Health Framework (AHF), que es una colección de tools de nueva generación, que trabajan juntas de forma autónoma en 24x7 para mantener los sistemas de bases de datos sanos y en funcionamiento mientras se minimiza el tiempo de reacción humana. Estas tools son las ya existentes ORAchk, Cluster Verification Utility, Trace File Analyzer, Cluster Health Monitor, Quality of Service Management, Memory Guardy las nuevas Cluster Health Advisor y Hang Manager.

Conclusiones y Recomendaciones

La batería de test propuestos puede no aplicar a un sistema concreto, o haberse probado ya con anterioridad, por lo que se deben definir las pruebas de verificación de un sistema en conjunto entre los DBAs y administradores de sistemas de forma que los escenarios que se prueben se correspondan con la realidad de los entornos.

En el caso de un sistema en cluster en el que las bases de datos no estén en RAC, algunas de las pruebas no tienen sentido, ya que se asume que hay 1 o varias bases de datos en RAC en el cluster.

También algunas de las pruebas pueden no tener sentido, ya que asumen que se puede añadir y quitar un nodo, lo que depende de los recursos que se dispongan.

Por último, tener en cuenta las herramientas de diagnóstico disponibles y tenerlas preparadas para minimizar el tiempo de reacción ante problemas.