



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 para Networking y Oracle SQL*Net*

*Referencia documento:
InfV5_JASAS_SQLNet_BestPractices_v920.doc
Fecha: 16 de noviembre de 2018
Versión: 9.2.0*

Registro de Cambios

Fecha	Autor	Versión	Notas
14 de octubre de 2010	Oracle ACS	1.4.0	Versión inicial
14 de enero de 2011	Oracle ACS	2.1.0	Version revisada
14 de abril de 2011	Oracle ACS	2.2.0	Version revisada
14 de Julio de 2011	Oracle ACS	2.3.0	Version revisada
13 de Octubre de 2011	Oracle ACS	2.4.0	Versión revisada
16 de noviembre de 2018	Oracle ACS	3.1.0	Versión revisada
14 de Marzo de 2013	Oracle ACS	4.1	Revisión de Marzo de 2013, contrato 2012-2014
13 de Junio de 2013	Oracle ACS	4.2	Revisión de Junio de 2013, contrato 2012-2014
17 de Octubre de 2013	Oracle ACS	4.3	Revisión de Octubre de 2013, contrato 2012-2014
16 de Julio de 2015	Gregorio Adame	6.1	Revisión de Julio de 2015, contrato 2015-2016
16 de Diciembre de 2015	Jonathan Ortiz	6.2	Revisión de Diciembre de 2015, contrato 2015-2016
16 de Junio de 2016	Gregorio Adame	7.1	Revisión de Diciembre de 2016, contrato 2016-2017
16 de Noviembre de 2016	Gregorio Adame	7.2	Revisión de Noviembre de 2016, contrato 2016-2017
16 de Junio de 2017	Gregorio Adame	8.1	Actualización documento
16 Noviembre 2017	Gregorio Adame	8.2	Revisión Noviembre de 2017
16 de Junio de 2018	Gregorio Adame	9.1	Revisión Junio de 2018
16 de Noviembre de 2018	Gregorio Adame Nisa	9.2	Revisión Noviembre de 2018

Revisiones

Nombre	Role
Jonathan Ortiz	Advanced Support Engineer

Distribución

Copia	Nombre	Empresa
1	Subdirección de Tecnologías de la Información	Consejería de Salud, Junta de Andalucía
2	Servicio de Coordinación de Informática de la Consejería de Hacienda y Administración Pública	Consejería de Hacienda y Administración Pública, Junta de Andalucía

Índice de Contenidos

INTRODUCCIÓN	4
OBJETIVOS DE ESTE DOCUMENTO	5
BEST PRACTICES DE ORACLE SQL*NET	6
<i>Recomendaciones generales</i>	6
<i>Optimización del parámetro SDU de SQL*Net</i>	7
<i>Recomendaciones para Real Application Cluster</i>	9
<i>Consideraciones sobre la Arquitectura Servidor</i>	13
<i>Consideraciones sobre interconnect e IPC</i>	16
<i>Seguridad</i>	19
USO Y LIMITACIONES DE LOS DATABASE LINKS	20
<i>Limitaciones de los database links</i>	20
<i>Operaciones sobre campos LOB sobre database links</i>	21

Introducción

Este documento recoge una serie de recomendaciones de Oracle Soporte planteadas como buenas prácticas de Oracle SQL*Net. El documento cubre tanto las recomendaciones generales como las recomendaciones para entornos basados en Oracle Real Application Cluster.

Objetivos de este documento

Este documento recoge una serie de recomendaciones de Oracle Soporte planteadas como buenas prácticas de Oracle SQL*Net. El documento cubre tanto las recomendaciones generales como las recomendaciones para entornos basados en Oracle Real Application Cluster.

A lo largo de los puntos de este documento se irá definiendo una guía de buenas prácticas para Oracle SQL*Net 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 versiones Oracle RDBMS Real Application Cluster 10gR2, 11gR1 y 11gR2, 12cR2

Best practices de Oracle SQL*Net

Recomendaciones generales

- No se recomienda el uso de “_” o underscores como parte del nombre de un host, hostname ni como parte del nombre de dominio, domainname.
- Se recomienda la configuración de la dirección IPC como primera dirección dentro de la configuración de direcciones del listener.ora. Esta recomendación está por defecto en todas las bases de datos a partir de 10gR2 y cuya configuración de red se haya realizado a través de NETCA. En caso de fallo, puede afectar negativamente al incrementar la cantidad de tiempo necesaria para que, por un ejemplo una VIP realice failover sobre otro nodo.
- Para evitar errores de tipo ORA-12545, es necesario asegurar que los ficheros de HOSTS y la configuración DNS de los clientes apuntan tanto a la VIP como a las hostnames públicos de los servidores de bases de datos.
- A partir de Oracle 10g, el proceso TNSListener es capaz de integrarse con la seguridad ofrecida por el sistema operativo. Esta funcionalidad posee las siguientes características:

- En el caso de que se invoque algún comando de control o considerado como privilegiado, stop, status, etc, por el mismo usuario que inició el proceso TNSListener, no será necesario proveer ningún password.
- Esta característica está habilitada por defecto, y puede comprobarse tanto en tiempo de arranque como a través del comando LSNRCTL STATUS. En tal caso, la salida debería ser algo parecido a:

```
Security ON: Local OS Authentication
```

- En el caso de que el proceso TNSListener se inicie por un usuario A, y otro usuario B, o por el usuario A desde otro nodo, intenta realizar algún comando privilegiado, error de retorno será:

```
TNS-01190: The user is not authorized to execute the requested listener command
```

- Se consideran comandos privilegiados los siguientes:

- SAVE_CONFIG, STOP, TRACE, SPAWN, RELOAD,
- SET
- {LOG_FILE, LOG_STATUS, INBOUND_CONNECT_TIMEOUT, SAVE_CONFIG_STOP_ON, TRC_FILE, TRC_LEVEL, LOG_DIRECTORY, STARTUP_WAITTIME} SHOW {RULES, TRC_DIRECTORY, LOG_FILE, LOG_STATUS, INBOUND_CONNECT_TIMEOUT, SNMP_VISIBLE, TRC_FILE, TRC_LEVEL, LOG_DIRECTORY, STARTUP_WAITTIME, SAVE_CONFIG_STOP_ON}

- Se recomienda en todo momento hacer uso de esta característica siempre que esté disponible.
- Como recomendación genérica no se recomienda el uso del carácter "." En los nombres de los interfaces de red o Network Interface Card (NIC).
- Configurar apropiadamente Oracle Net Services load balancing para distribuir las conexiones. El load balancing debe ser utilizado junto con Workload Management para proveer la alta disponibilidad. El atributo CLB_GOAL de un servicio puede ser configurado dependiendo los requerimientos de la aplicación, diferentes workloads requieren diferentes objetivos de balanceo de carga. Utilizar servicios separados para cada tipo de workload con diferente CLB_GOAL.

Optimización del parámetro SDU de SQL*Net

Por regla general se suele aumentar el valor del parámetro SQL*NET SDU a un valor alto, de 4Kb a 32Kb, con el fin de reducir los round trips, decrementar el tiempo de respuesta e intentar eliminar la no respuesta por parte del sistema.

En este punto veremos algunas consideraciones a tener en cuenta

SQL*Net Packet Sizes, parámetro SDU

- El SDU o Session Data Unit es parámetro de la capa NS encargada de controlar el tamaño de los envíos y las recepciones desde la capa NT. Este parámetro acepta valores del rango 512 a 32767 bytes, en 11g el valor por defecto es 8Kb, para pre-11g el valor por defecto es 2Kb.
- Para minimizar la sobrecarga y la fragmentación de mensajes debido a las cabeceras de Oracle Net, se suele establecer el valor del parámetro SDU como múltiplo del MSS o Maximum Segment Size del protocolo en uso.

Cómo calcular el MSS:

- Se puede definir el MSS como:
 - $MSS = MTU - TCP \text{ header size} - IP \text{ header size}$
- En el caso de TCP sobre Ethernet:
 - MTU 1500 bytes for Ethernet
 - TCP 20 bytes
 - IP 20 bytes
- Lo que deja 1460 bytes, aproximadamente ya que la cabecera TCP puede ser mayor, para el MSS de TCP/IP Ethernet.
- Tenemos que tener en cuenta que la capa Transparent Network Substrate (TNS) necesita otros 30 bytes adicionales, por lo que el valor real de datos que se envía es de 1430 bytes. Pero teniendo en cuenta que la cabecera TNS se incluye el paquete de datos de TCP, podemos incluir esta cabecera TNS en el tamaño de la SDU.

- Por ejemplo, si se desea enviar 5720 bytes, serán necesarios 4 paquetes TCP, $5720/1420=4$. En cada envío se añadirá 30 bytes extra por paquete como cabecera TNS, $1430+30=1460$. Posteriormente se añadirán 40 bytes como cabeceras TCP/IP, para completar el tamaño de los paquetes sobre Ethernet, $1460+40=1500$ bytes.
- Se recomienda por tanto, configurar adecuadamente los búferes de recepción y envío de TCP para que el proceso de Windowing de TCP/IP funcione correctamente.

Recomendaciones para el tamaño de SDU

- Optimos SDU varían según las aplicaciones
- Incrementar el SDU en el lado del cliente y del servidor
En la negociación de la conexión, prevalece el menor valor.
- Incrementar el valor a 8Kb (suficiente para muchos casos)
- Para escenarios con transferencias a granel (bulk), incrementar el valor a 64 Kb.
Transferencias de LOB y XML DB.
- SDU y MTU son independientes.

Configuración del parámetro SDU

- Para configurar el SDU es necesario establecer correctamente los valor en dos localizaciones:
- Opción A:
 - En el fichero TNSNAMES.ORA del cliente, donde el parámetro debe aparecer en la cláusula DESCRIPTION:

```
TEST =
(DESCRIPTION=
(SDU=8192)
(ADDRESS=(PROTOCOL=TCP) (HOST=oracle.com) (PORT=1521))
(CONNECT_DATA =
(SID = V1110)))
```
 - En el fichero LISTENER.ORA en la cláusula SID_DEC:

```
SID_LIST_LISTENER =
(SID_LIST =
(SID_DESC =
(SDU = 8192)
(SID_NAME = V1110)
(ORACLE_HOME = /oracle/product/11.1.0)))
```
 - A partir de 10.2.0.1 y versiones sucesivas, el tamaño por defecto de parámetro SDU puede cambiarse para conexiones a través de SERVICE_NAME.

- Opción B:
 - En el fichero SQLNET.ORA, a través de:

```
DEFAULT_SDU_SIZE = 8192
```
- De esta forma el parámetro puede establecer por conexión, añadiendo el valor tanto en el fichero de configuración local, TNSNAMES.ORA como en el fichero de configuración LISTENER.ORA, o bien estableciendo el valor para todas las conexiones a través del parámetro DEFAULT_SDU_SIZE del fichero de configuración SQLNET.ORA.

Recomendaciones para Real Application Cluster

- En instalaciones previas a 11.2, se recomendaba que el listener corriera bajo el ASM HOME. Esto permitía la registración de múltiples bases de datos con un listener sin necesidad de estar sujeto a un DATABASE HOME.
- A partir de 11.2 el listener debe correr bajo el GRID INFRASTRUCTURE HOME (por defecto).
- Se recomienda el uso de SCAN LISTENERS para las bases de datos 11.2 y posteriores, para simplificar las conexiones de clientes, y eliminar la necesidad de modificar las cadenas de conexiones cuando se añaden o quitan nodos del cluster. Asimismo, SCAN puede ser integrado con clientes pre-11.2, ver la nota 1058646.1.
- Se recomienda configurar el parámetro REMOTE_LISTENER, para soportar la redirección de carga de trabajo y el balanceo de carga, y para permitir la distribución de peticiones de conexión a través de las instancias del cluster. Para más información consultar la nota 728787.1.
- Para versiones previas a 11.2 el parámetro LOCAL_LISTENER debe apuntar a la VIP del listener local. Para 11.2 el parámetro LOCAL_LISTENER se debe configurar por defecto para permitir a que *oraagent* lo configure de acuerdo a la información almacenada en el OCR.
- Por lo general, un sólo listener por nodo del cluster (SCAN excluido), es necesario para administrar las conexiones para las bases de datos que se ejecutan en el cluster. Si por necesidad o requerimiento, hay más de un listener, es importante que estos sean registrados en el clusterware para garantizar la alta disponibilidad.
- Se recomienda que un servicio de base de datos sea creado para gestionar las conexiones de las aplicaciones a la base de datos. Esto proporcionará a los administradores la flexibilidad necesaria para implementar funciones tales como TAF, FAN, FCF y balanceo de carga de conexión en tiempo de ejecución, además de la capacidad para controlar fácilmente a que instancias pueden conectarse las aplicaciones. Para obtener más información sobre los servicios mirar el capítulo 5 de Oracle Real Application Clusters Administración y Guía de implementación de 11g Release 2 (11.2)).
- En la versión Oracle 11gR1, el uso de IPv6 no está soportado para configuraciones en RAC.

- En necesario descartar cualquier tipo de fallo en los switches, fallos en las HBA o puertos que pierdan paquetes. En la mayoría de los casos donde se producen salidas no ordenadas de nodos del clúster o node evictions, son o bien por excesivo tráfico en el interconnect o bien el switch, las tarjetas de red o ambos están más configuradas.
- En el caso de uso de RAC, es necesario asegurar que el default Gateway se encuentra en la misma red que las VIPs. De cualquier otra manera, pueden encontrarse errores durante la ejecución de RACGVIP y provocar el reinicio de los procesos Listeners.
- En el caso de uso de RAC, hay que contemplar el requisito de que todos los interfaces de red tengan el mismo nombre en todos los nodos del clúster.
- En el caso de uso de RAC, se recomienda el uso de Jumbo Frames siempre que esté soportado por el sistema para los interfaces que soporten el interconnect.
 - La tecnología de red Ethernet es una de las más extendidas como implementación de interconnects en RAC. Ethernet posee un tamaño de frame variable, entre 46 y 1500 bytes y debe ser un valor establecido entre todos los integrantes de la comunicación, hosts y switches.
 - El valor más alto, en este caso 1500 bytes, es conocido como Maximum Transmission Unit (MTU). Cuando un mensaje es mayor de este valor, es fragmentado en paquetes de 1500 o menos bytes.
 - En RAC, el valor correspondiente al DB_BLOCK_SIZE es multiplicado por el valor de MULTI_BLOCK_READ_COUNT para determinar el valor máximo de un mensaje entre dos instancias y el valor PARALLEL_EXECUTION_MESSAGE_SIZE determina el valor máximo de un mensaje de Parallel Query. Estos valores están en el rango de 2Kb a 64Kb o más, por lo que serán fragmentados en múltiples frames en el caso de usar un MTU bajo o por defecto. En 11.2 el valor por defecto para PARALLEL_EXECUTION_MESSAGE_SIZE es 16Kb, el cual puede ser suficiente para muchos casos.
 - El uso de Jumbo Frames permite que el frame Ethernet pueda exceder el máximo de 1500 bytes de MTU tal como se recoge en el estándar, a pesar de que el uso de Jumbo Frames está extendido sigue sin ser parte de éste.
 - Por tanto, aunque los beneficios son claros, no todo lo elementos de la electrónica de red son compatibles y capaces de operar con Jumbo Frames. Por tanto, se recomienda extremar el cuidado en su configuración.
- Se recomienda el uso de direcciones de red no enrutables para la red interconnect. De clase A, desde la 10.0.0.0 hasta 10.255.255.255, de clase B, desde la 172.16.0.0 hasta la 172.31.255.255, de clase C, desde la 192.168.0.0 hasta la 192.168.255.255.
 - En caso contrario, las herramientas CLUVFY y VIPCA pueden reportar errores al esperar este tipo de direcciones.
 - Sin embargo, puede ignorarse y no tenerse en cuenta ya que posible usar estas direcciones IP como IP públicas dentro de una red interna.

- Es necesario asegurar que desde el punto de vista de la configuración de la velocidad y de tipo de comunicación, half-duplex o full-duplex, todo es correcto y que la configuración permite el máximo rendimiento del hardware.
 - La gran mayoría de los problemas asociados a la GLOBAL CACHE se deben a bloques perdidos directamente provocados por configuraciones incompletas o erróneas.
 - Por tanto, es necesario establecer un mecanismo de monitorización de las estadísticas de la GLOBAL CACHE, concretamente, "gc cr block lost" y/o "gc current block lost") para cada uno de los nodos del clúster así como estas mismas estadísticas de forma agregada. Es importante detectar cualquier mal comportamiento de la GLOBAL CACHE y del ENQUEUE SERVICE, y la pérdida de paquetes puede ser un de los síntomas.
 - Hasta hora nos hemos centrados en problemas de rendimiento asociados a una configuración defectuosa. Sin embargo es posible que también se produzcan como efectos de una configuración defectuosa, salidas no ordenadas de los nodos o node eviction. Dado que el software de Clúster de Oracle se basa en el uso de heartbeat entre todos los nodos de éste, si un heartbeat se pierde de forma constante, la situación de node eviction será más que probable.
 - Los siguientes síntomas suelen asociarse a node evictions:
 - Tamaños de búferes de recepción UDP pequeños y desbordamientos de los búferes de los sockets.
 - Corrupción de paquetes detectados por los procesos de checksum de UDP provocado por el envío/recepción.
 - Tamaños de MTU erróneos.
 - Fallos en los cables y en las tarjetas.
 - Redes no dedicadas para el tráfico de interconnect.
 - Gran distancia entre los servidores y los switches.
 - Configuraciones de IPFILTER
 - Firmware desactualizados.
 - Protocolos propietarios.
 - Defectos de configuración en estrategias de bonding y link aggregation.
 - Defecto de configuración de Jumbo Frames
 - Tarjetas erróneamente forzadas como full dúplex.
 - Errores en los mecanismos de control y saturación de redes.
 - Pérdida de paquetes a nivel de sistema operativo, tarjetas o switches.

- Defectos en la configuración de los drivers de las tarjetas de red.
 - Longitud insuficiente en las colas de envío y recepción de las tarjetas.
 - Ancho de banda limitado o saturado.
 - CPU sobrecargadas.
 - Errores en el procesado de los paquetes por parte de los switches.
 - Políticas de QoS que impactan negativamente en el procesado de paquetes en el interconnect.
 - Fallos temporales derivados de la re convergencias del spanning tree.
- Como se ha comentado anteriormente, se recomienda el uso de tecnologías de bonding y/o link aggregation.
 - Se considera una best practice redundar el red privada de interconnect en cualquier RAC. Dependiendo del sistema operativo y del proveedor, pueden usarse drivers a nivel de sistema operativo para realizar bonding, teaming IPMP, Etherchannel, APA, MultiPrivNIC, etc. Sin embargo desde el punto de vista de RAC, basta con dos tarjetas conectadas a dos switches independientes para asegurar la continuidad en caso de un fallo.
 - La configuración de la alta disponibilidad del interconnect es transparente tanto al software de clúster como a la base de datos en RAC. Esto significa que cualquier error debe tratarse desde las capas inferiores, sistema operativo o comunicaciones, que serán por ejemplo, los responsables del failover de una tarjeta a otra.
 - Tanto RAC como el software de clúster de Oracle asumen que han sido desplegados en switches dedicados y físicamente separados de cualquier interferencia exterior.
 - El despliegue sobre switches compartidos o VLAN segmentadas, puede provocar la saturación del interconnect y por tanto la estabilidad del sistema.
 - En el caso de despliegue sobre una VLAN, ésta debe tener un mapeo 1 a 1 hacia una subred no rutable, y la primera no podrá ser de tipo TAGGED ni extender a múltiples switches.
 - Finalmente, es necesario controlar cualquier cambio en el Spanning Tree, ya que podría suponer la inundación temporal de la VLAN y/o de los puertos implicados.
 - Por regla general, en entornos de altos requisitos de servicio, suele usarse Infiniband para el interconnect. Esta tecnología aumenta el rendimiento al disminuir la latencia en la comunicación, particularmente en Oracle 11g al usar el protocolo RDS.

- El protocolo Reliable Datagram Sockets (RDS) permite la multiplexación paquetes UDP sobre una conexión Infiniband, consiguiendo, frente a Gigabit Ethernet, aumentos del ancho de banda de entre x10 y x30 mientras que a la misma vez consigue reducir la latencia en cifras similares.

Consideraciones sobre la Arquitectura Servidor

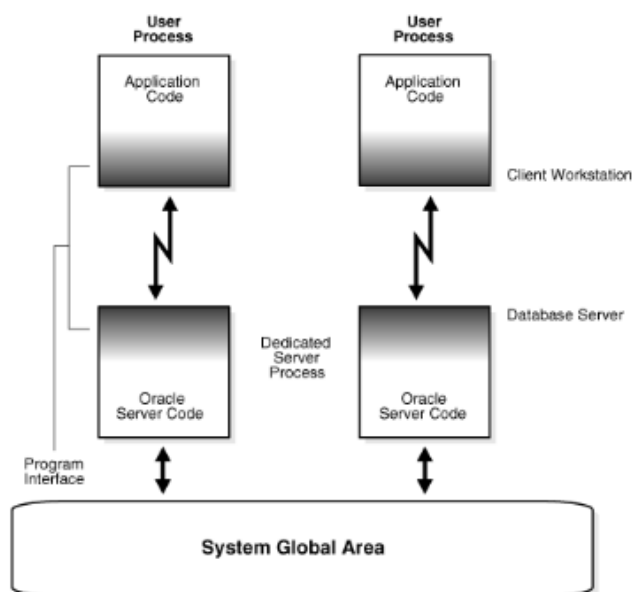
La selección de la correcta arquitectura servidor es clave para cubrir los requerimientos de escalabilidad.

Oracle Database Server soporta las 3 arquitecturas:

- Dedicated Server (defecto)
- Shared Server (MTS)
- Database Resident Connection Pool (a partir de 11g)

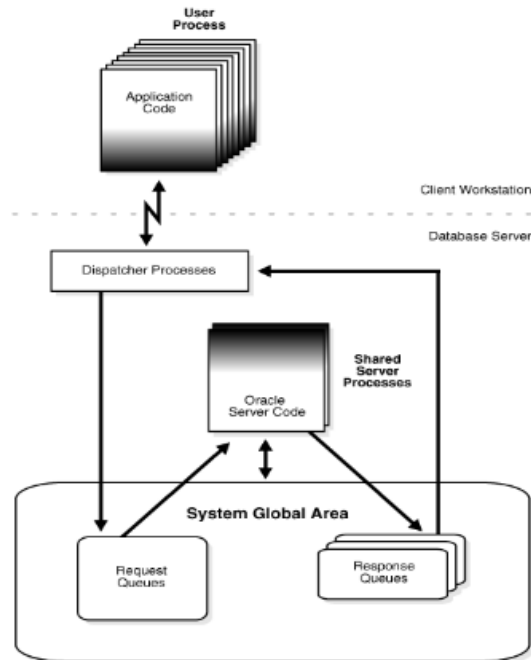
Dedicated Server

- Cada conexión cliente tiene su propio proceso (thread en Windows).
- Un proceso dedicado asegura baja latencia.
- Tiene que arrancar un nuevo proceso durante la conexión.
- Tiene que matar un proceso durante la desconexión.
- Límites de escalabilidad:
 - Memoria.
 - Número de procesos.



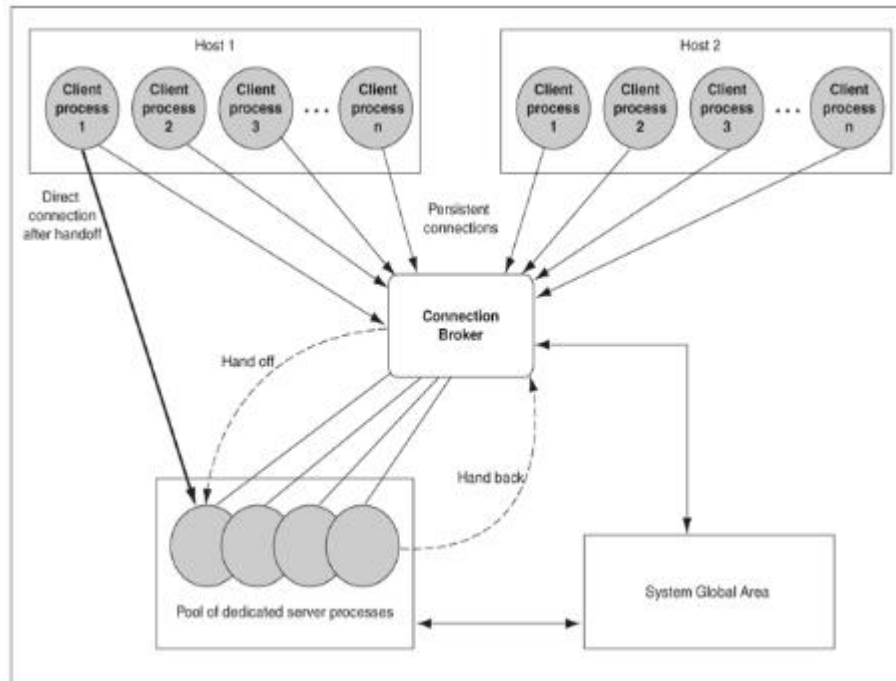
Shared Server (MTS)

- Cada servidor maneja múltiples clientes.
- Los dispatchers retransmiten peticiones y respuestas entre clientes y servidores.
- Conexiones “idle” consumen poca memoria.
- Bueno para un gran número de conexiones con algunas “idle”.
- La latencia aumenta (debido al man-in-the-middle).



Database Resident Connection Pool

- Genera un pool de servidores dedicados compartidos a través de clientes y procesos.
- Bajo costo de conexión y desconexión.
- Servidor (server) “locked” (bloqueado) en la conexión.
- Servidor (server) “released” (liberado) en la desconexión.
- Baja latencia de servidores dedicados.
- Gran escalabilidad con un driver de cliente DRCP.



Ejemplo de uso:

- El pool es habilitado por el DBA utilizando:

```
EXECUTE
DBMS_CONNECTION_POOL.START_POOL('SYS_DEFAULT_CONNECTION_POOL')
;
```

- Cambiar la cadena de conexión (tnsnames.ora del cliente):

```
(DESCRIPTION=
(AADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales)(SERVER=pooled))
```

- Se puede conectar fácilmente con la siguiente sintaxis:

```
sqlplus jouser@sales-server:1521/sales:POOLED
```

Dedicated versus Shared versus DRCP

- Utilizar Dedicated para:
 - Conexiones que necesiten alto rendimiento
 - Operaciones de larga duración o intensiva transferencia de datos
- Utilizar Shared para:
 - Sesiones que permanecen idle durante algún tiempo.
 - Clientes que frecuentemente conectan y desconectan.
 - Cuando tenga problemas de memoria y debe manejar un alto número de conexiones de clientes.
- Utilizar DRCP:

- Cuando tenga miles de usuarios que necesitan acceder a la base de datos por un período corto de tiempo.
- Aplicaciones que en su mayoría utilizan las mismas credenciales de base de datos e idénticas configuraciones de sesión.
- PHP (OCI8 extension), Python (cx_Oracle), Perl (DBI)

Consideraciones sobre interconnect e IPC

Optimización para la gestión de pérdida de bloques

Hasta la versión Oracle 11g, el timeout utilizado para la detección de bloques perdidos se realizaba a través del parámetro `_side_channel_batch_timeout`, que por defecto tiene el valor de 6 segundos.

En sistemas con alta carga de trabajo, con un número significativos de peticiones multibloque y con redes congestionadas, este tiempo de espera de 6 segundos podía causar efectos en cascada al incrementar el número de solicitudes de reenvío de bloques con la consiguiente ralentización del tráfico de red del InterConnect.

A partir de Oracle 11g, se han realizado una serie de mejoras en este algoritmo de detección de bloques perdidos, entre otras:

- Se configurado el parámetro a 500 ms, y ahora es controlado por el parámetro `_side_channel_batch_timeout_ms`.
- Se ha eliminado situaciones donde, falsamente, se detectaban bloques perdidos.
- Este nuevo añade la posibilidad de realizar lecturas a disco de bloques CR una vez se confirme que se han perdido, reduciendo así la congestión en el tráfico de red debidos a los reintentos. La decisión de usar uno u otro algoritmo se basa en la carga general del sistema, los ratios de E/S y los tiempos de recepción de los bloques CR a través del InterConnect.

Private interconnect

Independientemente del protocolo IPC que se esté usando, debe existir una red privada exclusivamente para el tráfico tanto de RAC como de CRS/CSS y es almacenada en el OCR. Es importante asegurar el uso exclusivo por parte del RAC y no confundir en ningún momento con el tráfico desde los servidores de aplicaciones o aplicaciones a través de la red pública.

Para averiguar la dirección IP que está usando CRS/CSS, podemos usar el comando `$CRS_HOME/bin/ocrdump <filename>` y localizar el valor de la clave `SYSTEM.css.node_numbers.node<n>.privatename` en el fichero de traza generado.

Nuevas estadísticas IPC en Oracle11g

A partir de Oracle 11g existen nuevas estadísticas de IPC, entre otras las siguientes:

- Estadísticas sobre latencias en el InterConnect.

- A partir de Oracle 11g, se crea un nuevo proceso de tipo de background que realiza periódicamente envío y recepciones de paquetes ICMP para comprobar la latencia de red entre todas las instancias del clúster. Es capaz de mantener estadísticas correspondientes a dos tipos de mensajes, de 500bytes y de 8Kbytes, y están disponible a través de AWR y a través de la vista X\$KSXPING.
- Es importante destacar que estas estadísticas incluyen tanto las latencias de los ping se ejecutan de forma periódica, por lo que si se observa un valor alto, puede deberse tanto a una latencia IPC como una latencia de un PING planificado.
- Esto conlleva que las latencias del proceso LMSn puedan ser mucho menores, ya que este tipo de procesos, por defecto, trabajan en real-time, al contrario que los procesos encargados de los PING que usan time-sharing scheduling.
- Finalmente, indicar que la medida de la latencia por este tipo de procesos se realiza entre ellos mismos, es decir, de un proceso PING de una instancia con otro proceso PING de otra instancia, por lo que el detalle anterior aplicaría doblemente.
- Throughput de interconnect por cliente.
 - A partir de Oracle 11g, la capa IPC recoge estadísticas de tráfico enviado y recibido por cliente. Estas estadísticas están disponibles a través de la AWR y la vista X\$KSPCLIENT y aportan información sobre la cantidad de ancho de banda en uso por cada cliente, por lo que problemas de asimetría en el tráfico son fácilmente detectables.
- Estadísticas por interfaces
 - Las estadísticas de los interfaces de red también están disponibles tanto a través de AWR como de la vista X\$KSXPIF. Estas estadísticas son similares a las que genera el comando netstat en Linux, métricas de envío y recepción, errores, paquetes “dropped”, etc.
 - Similarly to the statistics available through netstat (8) on Linux, statistics like sent / received metrics, errors and dropped packets could be viewed for performance analysis and/or diagnostics.
 - Es importante destacar que estas estadísticas son dependientes del sistema operativo por lo que pueden cambiar de una plataforma a otra.

Otras consideraciones sobre IPC

- Parámetro cluster_interconnects
 - Dado que la información de la configuración de Interconnect se almacena en el OCR, ya no es necesario a partir de Oracle 10g establecer el parámetro CLUSTER_INTERCONNECTS.
 - A pesar de que este parámetro soporta el balanceo de carga a través de los interfaces indicados, no soportan capacidades de FAILOVER, por lo que en caso de éstos pasen a estar no disponibles, se tratará como un fallo general y se comenzarán los abandonos forzados de los nodos, node evicting.

- Failover y balanceo de carga a nivel de interfaces.
 - Como se ha mencionado anteriormente, no se recomienda el uso del parámetro `CLUSTER_INTERCONNECTS` para realizar funciones de balanceo de carga. Además, dado que CRS no soporta ningún tipo de balanceo de carga o failover, es el sistema operativo el responsable de realizar estas labores.
 - Entre las distintas opciones disponibles, existen desde las puramente basadas en técnicas de failover como aquellas que tiene ambas funcionalidades, que en caso de estar disponibles serán las recomendadas por Oracle.
 - Realmente la tecnología detrás de estas funcionalidades pertenece a Cisco, pero puede encontrarse licenciadas en caso todos los sistemas operativos con distintos nombres, HP's Autoport aggregation, IBM Etherchannel, SUN Trunking etc.
 - En todas ellas, el procedimiento es el mismo, se ofrece un IP lógica desde el punto de vista del usuario y que es la que se especifica a los instaladores de productos Oracle.
- Protocolos IPC
 - De forma general se suele recomendar UDP sobre Gigabit Ethernet, aunque en aquellos entornos con altas necesidades, suele recomendarse UDP sobre Infiniband.
 - Otros protocolos como HMP (HP/UX), RSM (Solaris) o LLT (Symantec/Veritas) no son recomendados debido a deficiencias de estabilidad.
- Switches o cables cruzados
 - El uso de cables cruzados en clústeres de 2 nodos no está soportado.
- Jumbo frames
 - Tal como se ha comentado anteriormente, existe una significativa ganancia en el uso de protocolos IP y Jumbo frames al reducir la fragmentación de los mensajes y la sobrecarga del kernel a la hora de ordenarlos.
 - Para ello es fundamental que todos los elementos de comunicaciones, tarjetas y switches, soporten tamaños de frame de al menos, 9kbytes.

Seguridad

Comprobar Nodos Válidos por TCP

Hasta la versión Oracle 11g, el timeout utilizado para la detección de bloques

- Utilizar TCP Invited Nodes:
 - Lista de IPs o hostnames que son permitidos para conectar con la base de datos.
- Utilizar TCP Excluded Nodes:
 - Lista de IPs o hostnames que NO son permitidos para conectar con la base de datos.
- Los nodos invitados tienen prioridad sobre los excluidos.
- Para su habilitación, configurar el fichero sqlnet.ora (en el lado del servidor):

```
VALIDNODE_CHECKING = YES
```

```
TCP.INVITED_NODES = (hostname1, hostname2)
```

```
TCP.EXCLUDED_NODES = (hostname3, hostname4)
```

Uso y limitaciones de los database links

Limitaciones de los database links

No están permitidas las siguientes operaciones sobre database links:

- Establecer privilegios (grant privilege) sobre objetos remotos.
- Funciones de tipo pipeline.
- Ejecución de una actualización (update) remoto que use una subconsulta que referencie a un database link.
- Actualizaciones de columnas de tipo LONG como bind variable, así como actualizaciones de columnas de otro tipo de columnas a partir de una subconsulta que referencie a una database link y con un bind variable.
- Operación de COMMIT en una sesión donde se realiza una llamada a un procedimiento por RPC con un parámetro de tipo OUT o una llamada a una función.
 - Como workaround para algunos casos, puede usarse el pragma AUTONOMOUS_TRANSACTION en la función de llamada o en el procedimiento remoto.
- Comando DESCRIBE sobre objetos remotos, salvo que éstos sean de tipo tabla, vista, procedimiento y/o función.
- Operaciones de tipo ANALYZE sobre objetos remotos.
- Definiciones de integridad referencial.
- Establecimiento de roles (grant role) sobre usuarios en una base de datos remota.
- Operaciones de tipo DDL sobre bases de datos remotas.
- Acceso a tables remotas que contengan tipos definidos por el usuario.
- Obtención de roles de tipo non-defaults en una base de datos remota.
 - Por ejemplo, si el usuario HR se conecta a una base de datos local y ejecuta un procedimiento almacenado que usa un usuario fijo en la conexión a otra base de datos, como Scott por ejemplo, el usuario de HR recibe todos los roles por defecto de Scott en la base de datos remota y por tanto no está permitido el uso de SET ROLE.
- Ejecuciones de consultas con hash joins en entornos remotos que usen configuraciones SHARED SERVER.
- Creación de tablas donde sus partition-extended o subpartition-extended contengan un database link o un sinónimo que apunte a una tabla con un database link.
- Operaciones con operadores de tipo SPATIAL que usen índices espaciales sobre tables remota.

- Por ejemplo,

```
select count(*)
from customers@dblink
where SDO_ANYINTERACT (geom, window_geom) = 'TRUE';
```
- El acceso a tipos definidos por el usuario (User Defined Types).
 - Conexión a bases de datos remotas para realizar operaciones de SELECT, INSERT o UPDATE sobre tipos definidos por el usuario o el uso de objeto REF en una tabla remota.
 - Para ello se recomienda que en tiempo de CREATE TYPE se use la cláusula opcional OID para crear un identificador de objeto definido por el usuario, permitiendo así que el objeto pueda ser usado por múltiples bases de datos.
 - Bloques PL/SQL que declaren una variable local que sea de un tipo remoto definido por el usuario. Aplica igualmente sobre argumentos o valores de retorno en procedimientos y/o funciones PL/SQL.
- En el caso de uso de transacciones XA, el acceso a través de database link implica que la base de datos remota debe usar una configuración de tipo SHARED SERVER y ésta puede ser, exclusivamente, una base de datos Oracle.
- El uso de TAF para el acceso a bases de datos remotas a través de database links.
- A partir de 11g, 11.1.0.6, el uso de la función SYS_CONTEXT sobre database links.

Operaciones sobre campos LOB sobre database links

Selección de campos LOB a través de database links

En este apartado del documento se recogen las distintas posibilidades que existen a la hora de consultar campos LOB, tanto CLOB como BLOB, en bases de datos remotas.

Es necesario identificar dos escenarios, el primero, donde las bases de datos implicadas se corresponden con versiones estrictamente anteriores a Oracle RDBMS 9iR2, y un segundo escenario donde las bases de datos, ambas, son al menos, Oracle RDBMS 10gR2.

En el primer escenario, esta funcionalidad directamente no está soportada, y al seleccionar un campo LOB a través de una database link se obtiene el error ORA-22992.

Veámos un ejemplo de este escenario:

Base de datos que posee los datos, remota, SID_A

```
conn user/pass
create table tab1(col1 number, col2 clob);
insert into tab1 values(101, 'Hola mundo');
```

```
insert into tab1 values(102, 'Hello world');  
commit;
```

Base de datos que solicita los datos, local, SID_B

```
conn user/pass  
create database link link_to_A connect to user  
identified by pass using 'A';  
select * from test@link_to_A;  
ERROR at line 1:  
ORA-22992:cannot use LOB locators selected from  
remote tables
```

En el caso se usar PL/SQL, el resultado es equivalente.

```
set serveroutput on  
declare  
var1 varchar(6000);  
BEGIN  
SELECT col2 INTO var1 FROM tab1@link_to_A  
      where col1=101;  
dbms_output.put_line(var1);  
END;  
/  
ERROR at line 4:  
ORA-22992:cannot use LOB locators selected from  
remote tables
```

Tal como se ha comentado anteriormente, existe un segundo escenario, a partir de Oracle RDBMS 10gR2, donde esta limitación se matiza al ser posible vincular el resultado de la consulta a una variable de tipo PL/SQL, de tipo CLOB o tipo BLOB, según convenga.

Veámos un ejemplo de este escenario para campos de tipo CLOB,

```
declare  
var1 varchar(5000);  
BEGIN  
SELECT col2 INTO var1 FROM tab1@link_to_A  
      where col2=101;  
dbms_output.put_line(my_ad);  
END;  
/
```

Y en caso de campos de tipo BLOB,

```
declare  
var2 raw(50);  
BEGIN  
SELECT col2 INTO var2 FROM tab2@link_to_A
```

```
        where id=101;  
    END;  
/
```

Inserción de campos LOB a través de database link

En el caso de inserciones de campos o variables de tipo LOB, tanto CLOB como BLOB, en una base de remota a través de un database link, deberemos tener en cuenta que no es posible realizar de forma directa.

Por ejemplo, si usamos el siguiente PL/SQL para realizarlo:

```
declare  
    var1:= 'aaaa';  
begin  
    INSERT INTO table@DBLINK VALUES (var1);  
end;  
/
```

El resultado será bien el error ORA-600 [kokegPinLob1], ORA-600 [16622] o el error ORA-22922 "nonexistent LOB value" a partir de Oracle RDBMS 10gR2.

En este caso, la recomendación sería realizar la misma inserción usando un tabla temporal local, como se muestra a continuación.

```
insert into local_table values (...);  
insert into table@DBLINK select ... from local_table;
```