



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**



FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN

**“ORACLE REAL APPLICATION CLUSTER,
UNA TECNOLOGIA ACTUAL DE BASE DE DATOS”**

T E S I N A

**QUE PARA OBTENER EL TÍTULO DE :
LICENCIADA EN MATEMÁTICAS
APLICADAS Y COMPUTACIÓN
P R E S E N T A :
JESSICA PADILLA SANABRIA**

ASESORA:

LIC. TERESA CARRILLO RAMÍREZ

OCTUBRE DEL 2005

17349220



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



DEDICATORIAS.

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Jessica Padilla

FECHA: 21-10-2005

A mi mamá: FIRMA: [Firma]

A ti te doy gracias por todos tus cuidados Eres la mejor mujer que conozco. Te dedico este trabajo porque es algo que sin tus desvelos no hubiera podido ser. Te quiero mucho mamá.

A mi papá:

Qué puedo decirte, sino mil gracias por ser el mejor padre del mundo. No puedo expresar todo lo que significas para mi, así es que solamente digo GRACIAS. Este trabajo es tuyo.

A mi hermana:

Por ser siempre un excelente ejemplo a seguir, además de contar siempre con su apoyo y comprensión.

A mi cuñado:

Por estar incondicionalmente a nuestro lado en los momentos mas difíciles y felices de nuestra vida.

A Cuauhtémoc:

Por su apoyo incondicional y calidad humana envidiable, que me han enseñado vivir feliz.

A mis profesores:

Por ser parte de este logro y brindarme todo el soporte y dirección durante mi estancia en la Universidad.

A mi UNAM:

Por brindarme la oportunidad de estudiar una carrera profesional y darme las armas para poder luchar por un lugar digno en esta sociedad.

ÍNDICE.

DEDICATORIAS.

INTRODUCCIÓN. 1

CAPÍTULO 1: CONCEPTOS GENERALES. 3

1.1 Base de Datos.	3
1.1.1 Componentes de la Base de Datos.	3
1.1.2 Ventajas de Implementar una Base de Datos.	9
1.2 Tipos de Sistemas Manejadores de Bases De Datos.	10
1.2.1 Sistemas Relacionales.	10
1.2.2 Sistemas Jerárquicos.	11
1.2.3 Sistemas de Red.	12
1.3 Arquitectura de la Aplicación.	13
1.3.1 Arquitectura Cliente / Servidor.	13
1.3.2 Arquitectura Multicapas.	15
1.4 Sistema de Alta Disponibilidad.	17
1.5 Clusters.	17

CAPÍTULO 2: SISTEMA MANEJADOR DE BASE DE DATOS ORACLE. 22

2.1 Introducción al Servidor de Base de Datos Oracle.	22
2.2 Base de Datos Oracle.	23
2.2.1 Estructura Lógica.	23
2.2.2 Estructura Física.	25
2.3 Instancia Oracle.	26
2.3.1 Arquitectura de la Memoria.	26
2.3.2 Arquitectura de los Procesos.	30
2.4 Configuraciones de un Servidor de Base Datos Oracle.	39
2.4.1 Servidor Dedicado.	39
2.4.2 Servidor Compartido.	41

CAPÍTULO 3. ORACLE REAL APPLICATION CLUSTER.	45
3.1 Arquitectura de Hardware para Real Application Cluster.	46
3.1.1 Nodo.	47
3.1.2 Interconnect.	47
3.1.3 Subsistema de Discos Compartidos.	48
3.2 Arquitectura de Software para Real Application Cluster.	50
3.2.1 Software de Cluster: Operating System Dependent (Osd).	50
3.2.2 Componentes de la Base de Datos en Rac.	51
3.2.3 Procesos Especificos de una Instanciapara Real Application Clusters.	52
3.2.4 Los Servicios: Global Cache Service (Gcs) y Global Enqueue Service (Ges).	54
3.3 Cache Fusion.	56
CAPÍTULO 4. BENEFICIOS DE ORACLE REAL APPLICATION CLUSTER.	59
4.1 Ventajas Adquiridas en una Configuración de Rac.	59
4.1.1 Redundancia.	59
4.1.2 Failover.	60
4.1.3 Balanceo de Cargas.	62
4.1.4 Escalabilidad.	64
4.2 Comparación de RAC con otras Bases de Datos.	65
4.2.1 Rac (Shared Database) vs. Microsoft Sql Server (Federated Database).	65
4.2.2 Rac (Shared Database) vs. Ibm Db2 Udb Eee V8.1 (Shared Nothing Database).	70
CONCLUSIÓN.	77
GLOSARIO.	79
FUENTES DE INFORMACIÓN.	82

INTRODUCCIÓN.

La información se ha convertido en lo más importante e indispensable para el funcionamiento óptimo de las empresas, es por eso que se requiere tener disponibilidad y escalabilidad en las aplicaciones críticas que se encuentran en producción. Por ejemplo, para una empresa podría ser desastroso que sus aplicaciones principales dejaran de proporcionar servicio a los usuarios finales, por una falla en el aplicativo o en el hardware, lo que ocasionaría que el sistema quedara inhabilitado para poder acceder a los datos o a la información, y como consecuencia tener pérdidas tanto funcionales como monetarias.

Es normal que en las empresas que tienen sistemas de misión crítica, es decir que los sistemas no pueden dejar de dar servicio, se necesite implementar un ambiente de alta disponibilidad y escalabilidad, la solución más común para proteger al sistema de alguna falla se conoce como failover, este concepto se refiere a implementar un cluster por hardware, es decir, cuando se tiene un servidor en producción, existe otro servidor de las mismas características pero en standby o en espera, o sea, un ambiente de cluster activo - pasivo, para tener alta disponibilidad, pero si se dejara de funcionar el servidor primario, el servidor en standby tardaría en tomar el control del sistema entre 15 y 45 minutos, lo cual no es viable para cierta tipo de empresas, además de que se esta desperdiciando el poder de computo del segundo servidor, y ¿si llegara a fallar el segundo servidor?, sería catastrófico. Otro punto importante a considerar es el mantenimiento de estos servidores, ya que por lo general estas maquinas son grandes en capacidad, con el tiempo el mantenimiento de estos equipos se vuelve más caro incluso que el mismo servidor como hardware, además que se vuelven obsoletos.

Real Application Cluster (RAC) es una alternativa de solución a los problemas mencionados, ya que se puede tener un ambiente de alta disponibilidad, escalabilidad, mucho más económico y eficiente, porque no es un cluster de hardware sino de software, lo cual se explicará con detalle en el capítulo 3, esto es, cuando se implementa un ambiente de alta disponibilidad con Real Application Cluster se hace en cualquier máquina (PC's, servidores, frames) incluso con características diferentes, además se puede aprovechar el poder de computo de todos los componentes del cluster, aumentando el performance, es decir, el funcionamiento de las aplicaciones al doble o al triple según los nodos del cluster, estas ventajas se aprovechan en un ambiente de cluster activo - activo, en el cual se basa Real Application Cluster, por ejemplo si un servidor se llegara a caer, no detiene el trabajo de los usuarios porque basta con que un servidor del cluster quede funcionando correctamente para que la información siga disponible, de esta manera, la disponibilidad para los usuarios es transparente.

Cuando se cuenta con una base de datos ORACLE, como repositorio de datos, Real Application Cluster es una solución viable para tener disponibilidad y escalabilidad en aplicaciones, y por consecuencia en la información.

En el capítulo 1 presentare los conceptos básicos necesarios para entender la tecnología de las base de datos.

En seguida en el capítulo 2, me enfocare en las características del manejador de base de datos Oracle, en sus componentes, arquitectura y funcionamiento.

Una vez que se tenga claro el concepto de la base de datos y sus componentes, entrare completamente al concepto de Real Application Cluster en capitulo 3, donde mencionare los componentes, es decir su arquitectura, así como el funcionamiento del mismo.

Finalmente después de conocer el concepto y arquitectura de Real Application Cluster, expondré en capitulo 4 los beneficios de tener esta tecnología de base de datos en una empresa donde necesite un ambiente de alta disponibilidad y escalabilidad. También comparo RAC con otras tecnologías de cluster de base de datos proporcionadas por otros proveedores.

Teniendo como objetivo dar mas herramientas a los egresados de MAC para conocer la tecnología que se aplica actualmente en el mercado laboral y se facilite la integración al ámbito laboral, aplicando las habilidades aprendidas durante las diferentes asignaturas de la carrera, las cuales fueron transmitidos por el personal docente del la carrera de Matemáticas Aplicadas y Computación.

CAPÍTULO 1: CONCEPTOS GENERALES.

En este capítulo se tocarán los conceptos generales necesarios para poder abordar con más solidez los siguientes capítulos de esta tesina, los cuales se enfocan por completo a la tecnología oracle, más específicamente en la base de datos, y como una solución de alta disponibilidad de la base de datos en Real Application Cluster (RAC).

1.1 Base de Datos.

Un sistema de base de datos es fundamentalmente un sistema computarizado para administrar registros, es decir, una base de datos es un depósito o contenedor de una colección de datos computarizados.

Una base de datos es un conjunto de datos que son utilizados por los sistemas de aplicación de una empresa.

Toda empresa debe mantener el control de una gran cantidad de datos acerca de su operación funcional, es por eso que estos datos son almacenados y ordenados en una base de datos para que puedan ser explotados y utilizados para satisfacer las necesidades de dicha empresa.

Cuando existe una gran cantidad de información, es difícil ordenarla, administrarla de manera manual, entonces para consultarla de una manera eficiente, es cuando la implementación de un sistema de base de datos se convierte en una parte esencial para una empresa, en la actualidad la mayoría de las empresas tienen implementado un sistema de base de datos.

1.1.1 Componentes de la Base de Datos.

Una base de datos tiene 4 componentes principales: hardware, datos, software y usuarios.

Hardware.

Son los recursos físicos en los cuales estará funcionando una base de datos, por ejemplo, CPU (central process unit), unidades de almacenamiento, procesadores, memoria, y todos los componentes físicos que conforman una computadora.

Datos.

Son los hechos que describen sucesos y entidades."Datos" es una palabra en plural que se refiere a más de un hecho. A un hecho simple se le denomina elemento de dato.

Los datos son comunicados por varios tipos de símbolos tales como las letras del alfabeto, números, puntos y rayas, señales con la mano, dibujos, etc. Estos símbolos se pueden ordenar y reordenar de forma utilizable y se les denomina información.

Los datos son símbolos que describen condiciones, hechos, situaciones o valores.

Los datos se caracterizan por no contener ninguna información. Un dato puede significar un número, una letra, un signo ortográfico o cualquier símbolo que represente una cantidad, una medida, una palabra o una descripción.

La importancia de los datos está en su capacidad de asociarse dentro de un contexto para convertirse en información. Por sí mismos los datos no tienen capacidad de comunicar un significado y por tanto no pueden afectar el comportamiento de quien los recibe. Para ser útiles, los datos deben convertirse en información para ofrecer un significado, conocimiento, ideas o conclusiones.

Información es un conjunto de datos significativos y pertinentes que describan sucesos o entidades.

La información no es un conjunto de datos cualquiera. Es más bien una colección de hechos significativos y pertinentes, para el organismo u organización que los percibe.

Ahora bien, el término datos en una base de datos, se refiere a la información que para la empresa es importante y que se necesita optimizar para un mejor uso de está, que además, se base en las políticas dicha empresa.

Como por ejemplo, toda la información referente a un empleado como nombre, puesto, dirección, teléfono, etc. Es importante para cualquier empresa a la que pertenezca

Los datos en la base de datos deben de ser integrados y compartidos:

Integración: la base de datos es una unificación de datos distintos, en los cuales no debería de existir redundancia

Compartida: los datos pueden ser compartidos entre los diferentes usuarios y cada uno de ellos puede tener acceso a la misma pieza de datos, y probablemente con fines diferentes.

Software.

Al software también se le conoce como aplicación informática, es la parte lógica de la computadora, esto es, el conjunto de instrucciones (programas) que puede ejecutar el hardware para la realización de las tareas de cómputo a las que se destina.

La definición más formal de software es la atribuida a la IEEE en su estándar 729, que a continuación cito:

La suma total de los programas de cómputo, procedimientos, reglas [,] documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo (Ver: IEEE Std 729-1993, IEEE Software Engineering Standard 729-1993: Glossary of Software Engineering Terminology. IEEE Computer Society Press, 1993).

El principal componente de una base de datos es el software que permite a los usuarios del sistema interrelacionarse con los datos, conocido de manera indistinta como el sistema manejador de base de datos o servidor de base de datos.

Todas las solicitudes de acceso a la base de datos son manejadas por el DBMS (Data Base Management System). En otras palabras el DBMS ofrece a los usuarios una percepción de la base de datos que esta por encima del nivel del hardware y que maneja las operaciones del usuario expresadas en términos de este nivel más alto de percepción.

Si bien, el DBMS es el componente de software más importante del sistema no es el único, otros componentes comprenden las utilerías, herramientas de desarrollo de aplicaciones, generadores de reportes y el administrador de transacciones.

El término DBMS se usa también para referirse en forma genérica a un producto de base datos determinado de algún proveedor en específico como Oracle, Progress, DB2/Informix (estos dos últimos pertenecen a IBM), que actualmente están en el mercado, también este término se utiliza para referirse a una copia de dicho manejador de base de datos o producto que opera en alguna instalación determinada en un equipo determinado, es decir los discos de software que se le entregan al cliente cuando compra el producto.

El Sistema De Administración De Base De Datos.

El DBMS (Sistema De Administración De Base De Datos), es el software que maneja el acceso a la base de datos la siguiente manera:

- Un usuario emite una petición de acceso, utilizando algún sublenguaje de datos específico (ejemplo SQL).
- El DBMS intercepta esa petición y la analiza.
- El DBMS ejecuta las operaciones necesarias sobre la base de datos.
- El DBMS regresa los resultados solicitados por el usuario.

Las principales funciones del DBMS son las siguientes.

- **Definición de datos.** El manejador debe ser capaz de aceptar definiciones de datos en la forma fuente y convertir las a la forma objeto correspondiente.

- **Manipulación de datos.** El DBMS debe ser capaz de manejar peticiones para recuperar, actualizar o eliminar datos existentes en la base de datos o agregar nuevos datos a esta.
- **Optimización y ejecución:** Las peticiones deben ser procesadas por el componente optimizador, cuya finalidad es determinar una forma eficiente de implementar la petición.
- **Seguridad e integridad de los datos:** Se deben vigilar las peticiones del usuario y rechazar todo intento de violar las restricciones de seguridad e integridad definidas por el Administrador de Base de Datos (DBA).
- **Recuperación de datos y concurrencia:** El DBMS debe proporcionar una función de diccionario de datos. Este diccionario puede ser visto como una base de datos por derecho propio. El diccionario de datos contiene datos acerca de datos, es decir definiciones de otros objetos del sistema.
- **Rendimiento,** sobra decir que el DBMS debe realizar todas las tareas antes identificadas de la manera más eficiente posible.

Usuarios.

Un usuario es la persona, computadora o sistema que explota la información en un sistema de información para poder atender un requerimiento o necesidad.

Hay 3 tipos de usuarios que pueden acceder a una base de datos los cuales se describen a continuación:

- **Programadores de aplicación** son responsables de escribir los programas de aplicación de base de datos. Estos programas acceden a la base de datos emitiendo la solicitud apropiada al DBMS.
- **Usuarios finales** interactúan con el sistema desde las estaciones de trabajo o terminales en línea, aunque los usuarios finales pueden acceder a la base de datos a través de las aplicaciones en línea mencionadas anteriormente, o por medio de interfaz proporcionadas por el fabricante, es decir, estas no son escritas por el usuario.
- **El administrador de base de datos (DBA)** es responsable del control general del sistema a nivel técnico.

Las tareas de un DBA se resumen en los siguientes puntos:

- **Definir el esquema conceptual:** una vez que el usuario funcional o administrador del sistema, decidió el contenido de la base de datos a un nivel abstracto, entonces el DBA creará el esquema conceptual correspondiente.
- **Definir el esquema interno:** el DBA debe decidir la forma en qué van a ser representados los datos en la base de datos. A este proceso se le conoce como diseño físico de la base de datos. Una vez realizado el

diseño físico, el DBA deberá crear la definición de la estructura de almacenamiento.

- **Establecer un enlace con los usuarios:** el DBA debe de realizar el enlace con los usuarios para asegurarse que los datos necesarios estén disponibles para su uso.
- **Definir las restricciones de seguridad y de integridad.**
- **Supervisar el rendimiento y responder a los requerimientos cambiantes,** el DBA es el responsable de organizar el sistema de tal manera que se obtenga el rendimiento “ideal para la empresa”, y de hacer ajustes necesarios, es decir, afinar el rendimiento conforme las necesidades cambien.

La figura 1.1 muestra la interrelación entre las partes principales de las bases de datos

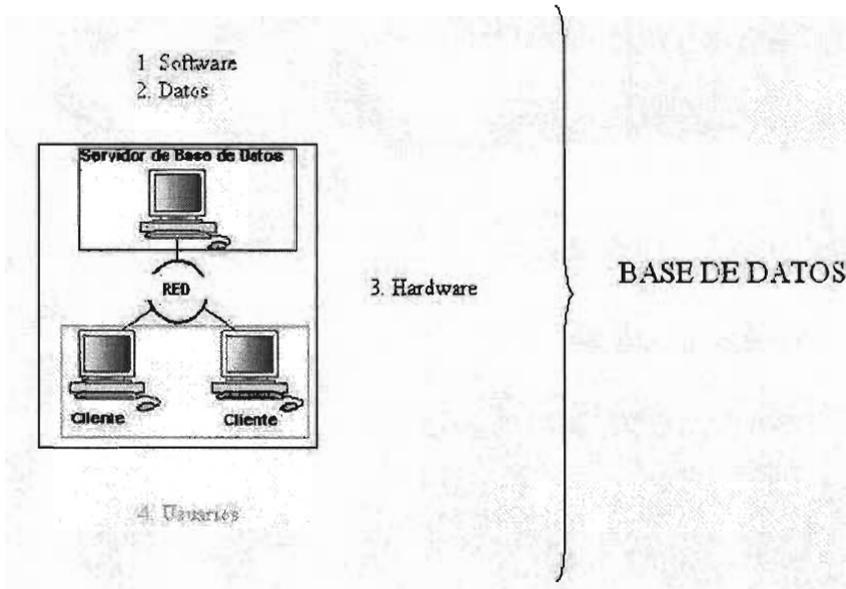


Figura 1.1 Componentes de base de datos

Niveles de Arquitectura de Base de Datos.

Una base de datos consta de diferentes niveles a continuación explicare cada uno de ellos:

- El nivel interno o nivel físico.
- El nivel externo o nivel lógico de usuario.
- El nivel conceptual o nivel lógico.

a. Nivel Interno.

La vista interna es una representación de bajo nivel de toda la base de datos y consiste en muchas ocurrencias de cada uno de los diversos tipos de registros internos. Esta vista todavía está distante del nivel físico y da por hecho un espacio de direcciones lineal infinito; los detalles de cómo el espacio de direcciones se asocia en el almacenamiento físico, son específicos del sistema operativo.

La vista interna se describe por medio del esquema interno, el cual no solo define los diversos tipos de registro almacenados sino que especifica también qué índices existen, cómo están representados los campos almacenados y en qué secuencia están dichos registros.

b. Nivel Externo.

El nivel externo es a nivel del usuario individual, un usuario puede ser un programador de aplicaciones, un usuario final con cualquier grado de conocimientos, desde un usuario funcional hasta un dba.

Cada usuario tiene a su disposición un lenguaje:

- Para el programador de aplicaciones, será un lenguaje de programación convencional (como PL/I, C++, Java) o bien un lenguaje de tipo propietario.
- Para el usuario final, el lenguaje será un lenguaje de consulta o un lenguaje de finalidad específica que puede estar controlado por formularios o menús.

Lo importante acerca de dichos lenguajes es que se incluirán en sublenguaje de datos, es decir, un subconjunto del lenguaje total que se ocupe específicamente de los objetos y operaciones de la base de datos. El sublenguaje de datos está incrustado dentro de su lenguaje anfitrión correspondiente. El lenguaje anfitrión es el responsable de proporcionar diversas propiedades que no son específicas de la base de datos como variables locales u operaciones de cálculo. El sublenguaje de datos específico soportado por casi todos los sistemas actuales es el lenguaje SQL.

Cualquier sublenguaje de datos es, en realidad, una combinación de por lo menos dos lenguajes subordinados:

- Un DDL (lenguaje de definición de datos), que permite la definición o declaración de objetos de base de datos. Consiste de aquellas construcciones declarativas de PL (Program Language) para declarar objetos en la base de datos.
- Un DML (lenguaje de manipulación de datos), que permite la manipulación o procesamiento de dichos objetos. Consisten en aquellas instrucciones ejecutables de PL que transfieren información desde y hacia la base de datos.

c. Nivel Conceptual.

La vista conceptual es una representación de todo el contenido de la información de la base de datos. Es una vista de los datos tal como son, en vez de tal como los usuarios están obligados a verlos debido a las limitaciones del lenguaje o el hardware.

Esta vista consiste en muchas ocurrencias de varios tipos de registro conceptual, y está definida por medio del esquema conceptual, el cual comprende definiciones de cada uno de los diversos tipos de registros conceptuales. Dicho esquema está escrito con otro lenguaje de definición de datos, y no debe haber ninguna referencia para la representación de campos almacenados, la secuencia de registros almacenados, los índices o cualquier otro detalle de almacenamiento y acceso.

En el momento que ya se tenga implementado un sistema de base de datos es necesaria una administración, de este concepto nace el administrador de datos (AD), que es indispensable en el sistema de base de datos. Esta persona es la encargada del control centralizado de los datos, es decir, debe entender los datos y las necesidades de la empresa con respecto a éstos, por lo tanto la tarea del AD es decidir, en primer lugar, qué datos deben ser almacenados en la base de datos y establecer políticas para mantenerlos y manejarlos una vez almacenados.

Ahora bien, ya que se tiene bien delimitado qué es lo que se quiere de un sistema de base de datos, entra en juego otra persona capaz de poder implementar estas ideas funcionales. El técnico responsable de implementar las decisiones del administrador de datos es el DBA, el trabajo del DBA consiste en crear la base de datos real implementar los controles técnicos necesarios para cumplir las diversas decisiones de las políticas hechas por el administrador de datos.

1.1.2 Ventajas de Implementar una Base de Datos.

Una vez que conocemos los componentes de una base de datos analizaremos algunas de las ventajas de contar con una de ellas, se listan a continuación:

- Compactación.
- Velocidad.
- El trabajo se optimiza.
- Datos Actuales.
- Datos Compartidos.
- Reducir la redundancia.
- Evitar inconsistencia.
- Manejo automatizado de transacciones.
- Mantener la integridad de la información.
- Mantener la seguridad de la información.

1.2 Tipos de Sistemas Manejadores de Bases de Datos.

1.2.1 Sistemas Relacionales.

Los productos DBMS (Data Base Management System) que se basan en el modelo de datos relacional (los “sistemas relacionales”) se les denomina RDBMS (Relational Data Base Management System) en la actualidad están dominando el mercado de base de datos, de hecho, la mayor parte de la investigación sobre bases de datos en los últimos 30 años, se ha basado en este modelo, incluso se puede decir que la presentación de este modelo relacional en los años 70’s por Edgar F. Codd fue el evento más importante en toda la historia de las bases de datos, puesto que este modelo esta sólidamente fundamentado en la lógica y en las matemáticas.

Un sistema relacional es aquel en el que:

- Los datos son percibidos por el usuario como tablas (relaciones).
- Los operadores disponibles para el usuario, son operadores que generan nuevas tablas a partir de las anteriores.

La razón por las que dichos sistemas se denominan relacionales es que el término relación es básicamente el término matemático para tabla.

Un sistema relacional se distingue de uno que no lo es porque, el usuario de un sistema relacional ve solamente tablas, en cambio, un usuario de un sistema no relacional ve otras estructuras de datos, las cuales requieren de otros operadores para manipularlas.

Los primeros productos relacionales comenzaron a aparecer a finales de los años setenta y principios de los ochenta. Hasta este momento la gran mayoría de los sistemas de bases de datos son relacionales y operan prácticamente en todo tipo de plataforma de hardware y de software disponible. Algunos ejemplos de estos manejadores son:

- DB2.
- Microsoft SQL.
- Oracle.
- Sybase.

Un ejemplo de una entidad relación se muestra en seguida.

Una entidad llamada PRODUCTO tiene los siguientes atributos:

- Id (Primary key).
- Nombre (Obligatorio).

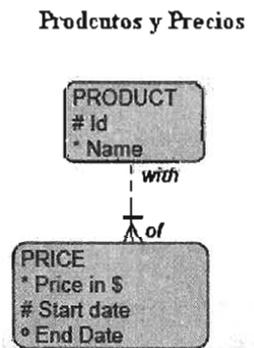
Una entidad NOMBRE tiene las siguientes atributos:

- Precio en pesos (Obligatorio).
- Fecha de inicio (Primary key).
- Fecha de terminación (Opcional).

Y tienen la siguiente relación: Un producto puede tener 1 o varios precios dependiendo a quien se les va a vender:

- Socios.
- Clientes.
- Vendedores.

En la figura 1.2 se muestra el modelo entidad relación del ejemplo anterior:



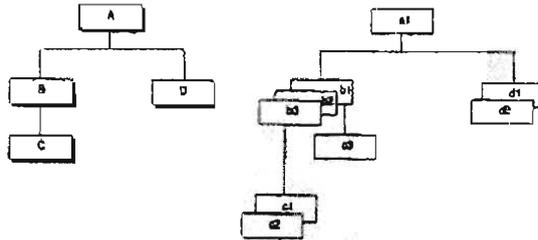
1.2 Modelo Entidad Relación

1.2.2 Sistemas Jerárquicos.

Entre los primeros modelos de datos que surgieron se encuentran las estructuras de árbol, propias de los productos jerárquicos.

Es un modelo muy rígido en el que las diferentes entidades de las que está compuesta una determinada situación, se organizan en niveles múltiples de acuerdo a una estricta relación PADRE/HIJO, de manera que un padre puede tener más de un hijo, todos ellos localizados en el mismo nivel, y un hijo únicamente puede tener un padre situado en el nivel inmediatamente superior al suyo.

En la figura 1.3 se muestra un modelo jerárquico un nivel y uno con 3 niveles.



1.3 Modelo Jerárquico.

Los árboles, como instrumentos para la representación de estructuras de datos, presentan problemas por su poca flexibilidad, lo que da origen a una falta de adaptación a muchas organizaciones reales.

No se ha llegado a una formalización matemática del modelo y de sus lenguajes, como ha ocurrido en el caso del relacional; ni tampoco se ha intentado su estandarización, a pesar de lo cual los productos jerárquicos (debemos considerar el IMS y el DL/I de IBM como máximos exponentes de estos sistemas) consiguieron altas ventas de mercado, sin embargo la actual difusión de la tecnología relacional los han llevado a convertirse en sistemas superados, lo cual no quiere decir que ya no existan, todavía hay importantes aplicaciones soportadas en estos productos las cuales están trabajando aceptablemente, por su eficiente respuesta y la satisfacción de sus usuarios.

1.2.3 Sistemas de Red.

El modelo de datos en red representa las entidades en forma de nodos de una gráfica, y las interrelaciones entre éstas mediante arcos que unen dichos nodos. En principio, esta representación no impone restricción alguna acerca del tipo y el número de arcos que puede haber, con lo que se pueden modelar estructuras de datos tan complejas como sea necesario.

Este modelo, consta de un componente estático y otro dinámico. El estático estaría compuesto por los objetos (entidades o nodos y atributos), las interrelaciones o arcos y las restricciones, que a su vez pueden ser inherentes y de usuario (pueden ser reconocidas por el modelo de datos o de responsabilidad exclusiva del usuario). Dentro del componente estático podemos citar un elemento más que atendería a la representación mediante gráficas.

Por otro lado, el componente dinámico estaría compuesto por el aspecto navegacional.

El esquema representa los aspectos estáticos, es decir, la estructura de los datos, que comprende los tipos de entidades, interrelaciones. Una ocurrencia del esquema son los valores que toman los elementos del esquema en un determinado momento. Estos valores

irán variando a lo largo del tiempo debido a la aplicación de los operadores de manipulación de datos a una ocurrencia del esquema.

El modelo es muy flexible debido a la inexistencia de restricciones inherentes. Esto implica dificultad a la hora de implementarlo físicamente y a la larga poco eficiente. Es por esto que el modelo sea tan solo teórico, y que a la hora de llevarlo a la práctica se introduzcan restricciones.

En la figura 1.4 se muestra la relación entre los nodos de la red.

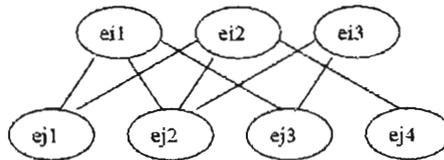


Figura 1.4 Modelo de Red

1.3 Arquitectura de la Aplicación.

Existen básicamente dos arquitecturas de aplicaciones para acceder a la base de datos:

- Cliente/Servidor.
- Multicapas.

1.3.1 Arquitectura Cliente / Servidor.

En el ambiente del sistema de base de datos, la aplicación de base de datos y la base de datos están separadas en dos partes: un front - end o porción del cliente y el back- end o porción del servidor. Los clientes corren la aplicación que accesa a la información de la base de datos e interactúa con el usuario a través del teclado, pantalla, mouse. El DBMS corre en el servidor y soporta las funciones requeridas.

Servidor.

Son máquinas que actúan como "almacenes" de información, esta información es solicitada por los clientes, de forma que el servidor responde a peticiones de información devolviendo los datos solicitados.

Clientes.

Un cliente inicia un requerimiento para que una operación sea realizada en un servidor de base de datos. El cliente puede ser un navegador de Web o cualquier otro proceso de usuario final. En una arquitectura de multicapas.

Aunque la aplicación cliente y el RDBMS puedan correr sobre la misma computadora, es recomendable que la aplicación cliente y el servidor corran en diferentes computadoras conectadas a través de la red, para un mejor funcionamiento, ya que la carga de trabajo se divide, es decir, los usuarios finales abrirán su aplicación desde su estación de trabajo utilizando memoria y cpu localmente, mientras que en el servidor solo habrá conexiones remotas.

Ejemplos:

- En la figura 1.5, el cliente y el servidor se encuentran en diferentes máquinas y están conectadas a través de la red. El servidor de base de datos y los clientes son comunicados a través de los servicios de red.
- En la figura 1.6, una computadora con más capacidad de cómputo se divide la ejecución de la aplicación del cliente y del servidor de una manera óptima.

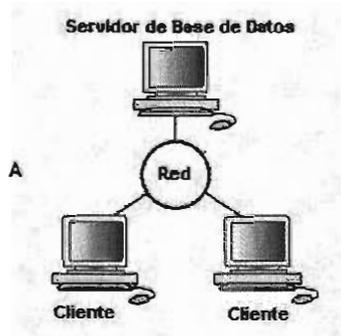


Figura 1.5

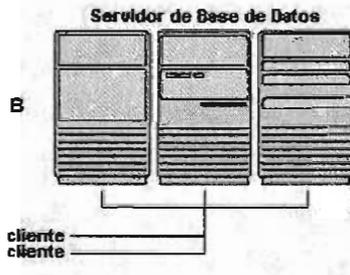


Figura 1.6

La arquitectura cliente/servidor en diferentes máquinas proporciona los siguientes beneficios:

- Las aplicaciones cliente no son responsables por el rendimiento de ningún procesamiento de datos.
- Las aplicaciones cliente no son dependientes de donde se encuentren físicamente, es decir, pueden estar en diferentes lugares los clientes y acceder a los mismos datos, y todos los clientes tendrán la misma funcionalidad, aunque los datos sean movidos o distribuidos en otros servidores de base de datos, la aplicación debe de continuar trabajando de igual manera, es por eso que los clientes no dependen de la configuración del servidor para poder tener la misma funcionalidad requerida.
- El RDBMS explota las facilidades de multitarea y de memoria compartida de los sistemas operativos. Dando como resultado, la ejecución de un grado alto de concurrencia, integridad de datos y el performance de las aplicaciones cliente.
- La estaciones de trabajo cliente o las terminales pueden ser optimizadas por las presentaciones de datos, por ejemplo, en despliegue de gráficas; y el servidor puede ser optimizado por el procesamiento y el almacenamiento de datos, por ejemplo, teniendo grandes cantidades de memoria y espacio en discos.
- En los ambientes de redes, se puede utilizar estaciones de trabajo no muy caras para acceder a los datos remotos de servidor eficientemente.
- Si es necesario, el RDBMS puede ser escalable. Se pueden agregar múltiples servidores para distribuir la carga de procesamiento a través de la red (escalar horizontalmente), o también se puede mover el RDBMS a minicomputadoras o mainframe, para tomar ventajas de un sistema grande (escalar verticalmente). En otro caso, todos los datos y aplicaciones son mantenidas con pequeñas modificaciones o sin ninguna.
- En un ambiente de red, los datos compartidos son almacenados en los servidores en lugar de almacenarlos en todas las computadoras del sistema. Esto hace más fácil y más eficiente el manejo de acceso concurrente.
- En los ambientes de red, las aplicaciones clientes dan de alta los requerimientos al servidor usando sentencias SQL. Después de que sean recibidas, las sentencias SQL son procesadas por el servidor, y los resultados son regresados a la aplicación cliente. El tráfico de red es utilizando al mínimo, porque solo los requerimientos y los resultados navegan a través de la red.

1.3.2 Arquitectura Multicapas.

En un ambiente de arquitectura multicapas, un servidor de aplicación proporciona datos para clientes y servidores como una interfaz entre clientes y servidores de base de datos. El cliente se conecta al servidor de base de datos a través de uno o más servidores de aplicaciones. Esta arquitectura es importante por el uso de internet.

Servidor de Aplicaciones.

Un servidor de aplicaciones proporciona acceso a los datos para el cliente. Este servidor actúa como una interfase entre el cliente y uno o más servidores de base de datos, el cual proporciona un nivel de seguridad. También se puede mejorar el procesamiento de algunos queries para el cliente, removiendo algunas cargas del servidor de base de datos.

El servidor de aplicaciones asume la identidad del cliente cuando la operación se esta realizando en el servidor de base datos para ese cliente. Los privilegios del servidor de aplicaciones son restringidos para prevenir que se realicen operaciones innecesarias o indeseables durante la operación del cliente.

Esta arquitectura habilita el uso de un servidor de aplicaciones:

- Valida la autenticación del cliente, como un navegador de Web.
- Conecta un servidor de base de datos.
- Mejora la operación de requerimientos.

En la figura 1.7 se muestra como se interactúan las tres capas de la arquitectura (multicapas), es decir, los clientes se conectan a los servidores de aplicaciones correspondientes, los cuales se dividen la carga de usuarios y direccionan a su vez a los usuarios al manejador de base de datos correspondientes.

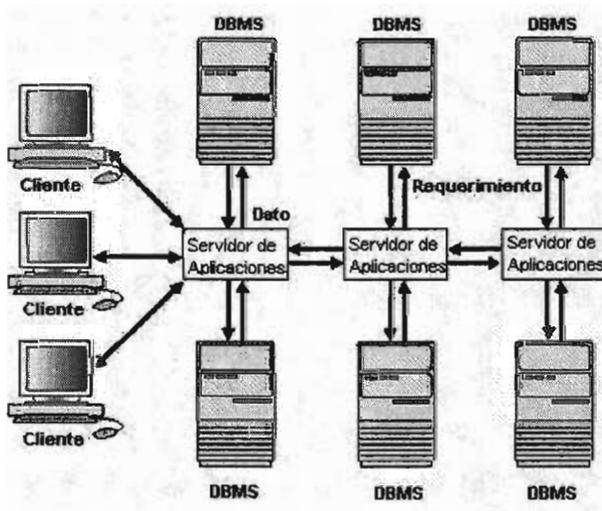


Figura 1.7

1.4 Sistema de Alta Disponibilidad.

Los ambientes computacionales que deben proporcionar disponibilidad por tiempo completo, es decir que deben estar dando servicio las 24 horas de horas del día, son conocidos como sistemas de alta disponibilidad. Estos sistemas típicamente tienen redundancia en hardware y software que hace que el sistema siga disponible aunque se presenten algunas fallas o interrupciones.

Debemos diferenciar dos tipos de interrupciones en los sistemas.

- **Las interrupciones previstas** son Las que se realizan cuando el sistema se tiene que detener para realizar cambios o mejoras en nuestro hardware o software.
- **Las interrupciones imprevistas** son las que suceden por acontecimientos imprevistos (como un apagón, un error del hardware o del software, problemas de seguridad, un desastre natural, virus, accidentes, caídas involuntarias del sistema).

En el momento en que una falla ocurre, el proceso de failover realiza el cambio del componente que fallo al componente de respaldo. Este proceso cubre los recursos del sistema, recupera transacciones parciales o fallidas, y regresa el sistema a la normalidad en unos cuantos segundos. El failover es transparente para los usuarios, pues ellos no deben percibir cuando un sistema de alta disponibilidad falla.

Cuando se tiene una rápida recuperación se reducen los efectos de la falla en el sistema que está el línea, es decir entre menos tiempo este abajo el sistema, es menos la pérdida de transacciones, además son menores la pérdidas funcionales y monetarias.

En el mercado existen diferentes soluciones de alta disponibilidad enfocadas a la base de datos, las cuales son ofrecidas por diferentes proveedores como por ejemplo IBM, Microsoft y Oracle principalmente.

1.5 Clusters.

Un cluster consiste en el uso de múltiples computadoras de cualquier tipo, es decir, PC's, servidores, estaciones de trabajo, dominios, los cuales puedes tener múltiples dispositivos de almacenamiento, conexiones redundantes, para crear un sistema de alta disponibilidad para que el acceso a la aplicación no sea interrumpido por la falla de uno de los componentes del sistema.

Actualmente un gran numero de proveedores proporcionan soluciones de clusters, como por ejemplo alta disponibilidad en equipo de hardware, el cual es un esquema de pasivo – activo, es decir que mientras un nodo este trabajando, existe otro que está esperando a que ocurra alguna falla para tomar el control.

Otra característica común del cluster es que haga parecer ante las aplicaciones que se esta trabajando con un solo servidor, además de que la administración de los servidores debe de ser lo más parecido posible a la administración de un solo servidor.

Ahora bien, es importante distinguir entre **arquitecturas de hardware en cluster** y **arquitecturas de base de datos en cluster**, son conceptos diferentes pero ambos se complementan para tener una solución de alta disponibilidad completa y eficiente hacia los usuarios que utilizan los servicios. A continuación se explicara cada una de estas arquitecturas:

Sistema de hardware en cluster.

Cuando se tiene un sistema de hardware en cluster, la alta disponibilidad de las aplicaciones depende únicamente de la configuración del hardware por medio del sistema operativo, es decir las aplicaciones no tienen ninguna configuración especial para que sigan funcionando después de una falla inesperada.

Hay tres principales implementaciones de hardware en cluster.

- **Shared disk clusters:** todos los nodos tienen acceso directo a todos los discos de almacenamiento. Para lograr este tipo de conexión compartida se necesita un esquema especial para la conectividad de discos de almacenamiento, para la cual se utiliza una SAN (Storage Area Network), esta tecnología de almacenamiento evita las limitaciones de los discos internos normales, por ejemplo en estos discos no se pueden escribir, acceder y leer dos usuarios al mismo tiempo, los discos compartidos permiten a cada nodo del cluster conectarse a un gran número de dispositivos al mismo tiempo.
- **Shared nothing clusters:** cada nodo tiene acceso a solo un conjunto específico de discos de almacenamiento, y solo a un conjunto de información.
- **Fail – Over:** Se tiene dos servidores, un servidor primario que esta en producción funcionando normal y un servidor secundario, el cual esta en espera a que el otro falle.

En ambos servidores se encuentran los mismos componentes, el mismo software para que cuando ocurra una falla inesperada que deje al servidor primario fuera de servicio, el segundo servidor reemplace al primario dando el servicio sin ningún problema.

Arquitecturas de Base de Datos en Cluster.

En esta configuración las aplicaciones o el software se involucran para tener alta funcionalidad en el servicio, en este caso el software que se configuraría en cluster sería la base de datos.

El concepto de sistema de base de datos en cluster se basa en dos tecnologías: la de hardware en cluster y la de base de datos, estas dos tecnologías se complementan, por una

parte tomando todo el potencial de tener un cluster en el hardware para poder montar una base de datos en cluster, dando como resultado una base de datos en alta disponibilidad.

Un sistema de base de datos en cluster necesita un sistema de hardware en cluster para funcionar pero cada uno de estas tecnologías tiene su propia arquitectura.

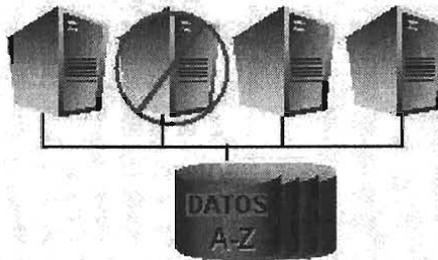
La arquitectura de un sistema de base de datos en cluster se basa en los siguientes componentes primarios de una de base de datos en cluster son:

- Servidores o nodos.
- Interconnect.
- Un subsistema de discos.
- Base de datos.

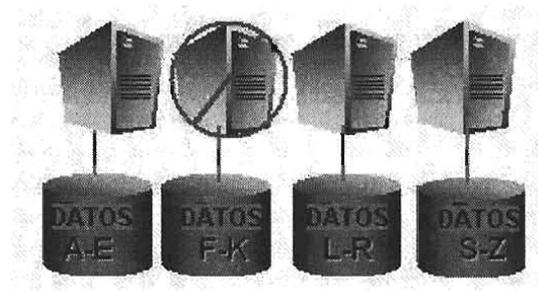
Mas adelante en el capitulo 3 comentaremos más a detalle estos componentes.

Las arquitecturas de bases de datos en cluster que existen en el mercado, son lo siguientes:

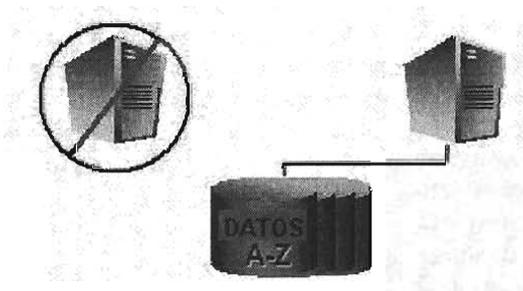
- **Shared-disk cluster:** Todas las instancias tienen acceso directo a todos los discos de almacenamiento en donde se encuentran los archivos de base de datos, si un servidor llegara a fallar no habría una perdida de servicio, pues los demás servidores (donde se encuentran las instancias) seguirían operando y accedendo a los mismo archivos de datos (que se encuentra en los discos compartidos), y el usuario no se dará cuenta de que se ha perdido este servidor.



- **Shared nothing cluster:** Cada instancia tiene acceso a solo un conjunto específico de información de los archivos de base de datos. Esto ocasiona que cuando una instancia llegue a fallar los datos a los cuales accesa quedan inoperables, por lo menos el tiempo que se tarde otro nodo que pertenezca al cluster en reconectarse a esos discos.



- **Failover cluster:** Se tiene dos servidores de base de datos, una base de datos esta en producción funcionando normal, existe un segunda base de datos la cual esta en espera de haya una falla, en ambos servidores se encuentran la misma base de datos, para que cuando sea necesario que el segundo servidor reemplaza el primario lo haga sin ningún problema. Estas operaciones de recuperación, por supuesto que se llevan tiempo de producción inoperable, dependiendo de cuanto se tarde en levantar el servidor, hacer la reconexión de los clientes al nuevo servidor, pueden ser minutos u horas.



En estos momentos ya explique cada una de las tecnologías de cluster, el hardware y el software, la principal diferencia entre estas dos tecnologías, es que un cluster de hardware es una solución de alta disponibilidad para cualquier aplicación que resida en el servidor, mas sin en cambio un cluster de base de datos se enfoca al software de base de datos la cual depende la tecnología de cluster de software para su funcionamiento.

Cada solución es correcta dependiendo de las necesidades del cliente, por ejemplo si una aplicación no puede estar fuera de servicio ni un minuto, es recomendable que no se use el fail – over ya que esta solución necesita tiempo para volver a poner el servicio disponible, la solución correcta seria el shared - disk cluster ya que maneja una tecnología donde todos los nodos están activos y dando servicio al mismo tiempo, es decir, esto lo decide el tipo de negocio con que se este tratando.

CAPÍTULO 2: SISTEMA MANEJADOR DE BASE DE DATOS ORACLE.

Una vez que se explico los conceptos generales, es importante conocer la estructura y funcionamiento del sistema manejador de bases de datos Oracle, el DBMS de Oracle corresponde al enfoque relacional establecido por Edgar F. Codd, mencionado en el capítulo 1.

La tecnología de alta disponibilidad de bases de datos Real Application Cluster (RAC) se basa totalmente en arquitectura del RDBMS (Relational Data Base Magament System), incluso, el RAC es una extensión del RDBMS de Oracle para proporcionar alta disponibilidad y un mejor desempeño de la base de datos.

Por esta razón primero debemos de entender bien cómo funciona y cómo está estructurada la arquitectura del RDBMS de Oracle de una manera normal, es decir standalone, como si no estuviéramos considerando ningún esquema de alta disponibilidad. Simplemente como un servidor de base de datos común y corriente.

2.1 Introducción al Servidor de Base de Datos Oracle.

El servidor Oracle es un sistema manejador de base de datos que provee un enfoque acerca del manejo de información. Un servidor Oracle esta formado por una base de datos y una instancia. Como se muestra en la Figura 2.1.

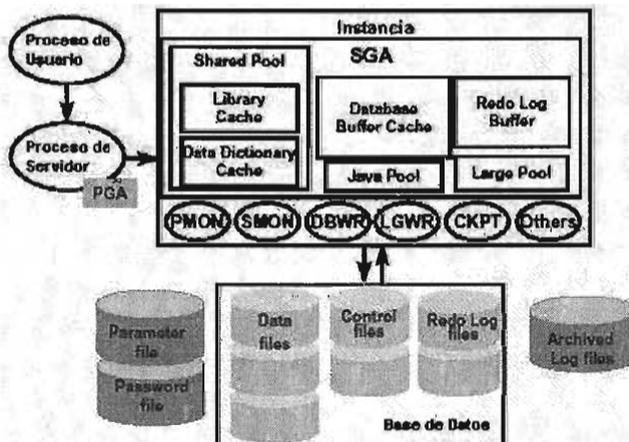


Figura 2.1 Manejador de Base de Datos Oracle

A continuación describiremos cada uno de los componentes:

2.2 Base de Datos Oracle.

Una base de datos Oracle esta conformada de una estructura lógica y una estructura física, a continuación se explican estos conceptos

2.2.1 Estructura Lógica.

Es la organización lógica de los objetos de la base de datos, es decir que ocupa dicha base de datos. Esta conformada por: Esquemas, datablocks, segmentos, extents y tablespaces, cada uno de los cuales explicaremos en los siguientes apartados.

Esquemas.

Un esquema es una colección de objetos que es utilizado por un usuario de la base de datos y tiene el mismo nombre que el usuario. Un esquema de usuario son las estructuras lógicas que se refieren directamente a los datos de la base de datos, por ejemplo un objeto de la base de datos podrían ser tablas, vistas o índices,

Data blocks, Extents, Segments, Tablespaces.

Estos componentes habilitan al servidor de base de datos para tener control detallado del uso del espacio del disco.

Oracle asigna espacio lógico para todos los datos que son utilizados en la base de datos. Las unidades de almacenamiento asignadas son data blocks, extents, segmentos:

- **Data blocks:** es el nivel más fino de granularidad (granulo: es una unidad de almacenamiento de memoria virtual contigua), la base de datos Oracle es almacenada en data blocks. Un data block corresponde a un numero específico de bytes de espacio físico en disco.
- **Extents:** El siguiente nivel del espacio lógico es un extent el cual es un número específico de data blocks contiguos, cuando un extent es asignado es usado para almacenar información.
- **Segmentos:** Después de los extents, el siguiente nivel lógico de almacenamiento es un segmento. Un segmento es un conjunto de extents asignados para una cierta estructura lógica.

La figura 2.2 muestra la relación entre estas estructuras lógicas de datos:

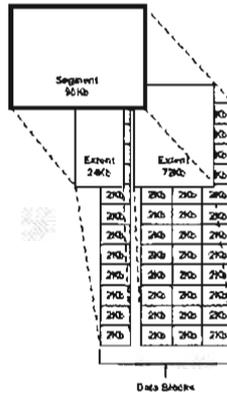


Figura 2.2 Relación entre las estructuras lógicas de la base de datos

- **Tablespaces:** Una base de datos está formada por una o varias unidades lógicas llamadas tablespaces en donde se almacenan lógicamente los objetos de la base de datos como tablas, índices, procedimientos, el cual a su vez están compuestos por uno o más segmentos. Además, cada uno de estos tablespaces contiene uno o varios archivos físicos que son los datafiles, es decir, la información utilizada. Un datafile solamente puede pertenecer a un tablespace. Por lo tanto, los datafiles de una base de datos son todos los datafiles que forman parte de todos los tablespaces de la base.

La figura 2.3 describe los componentes que forma parte de la estructura lógica de la base de datos:

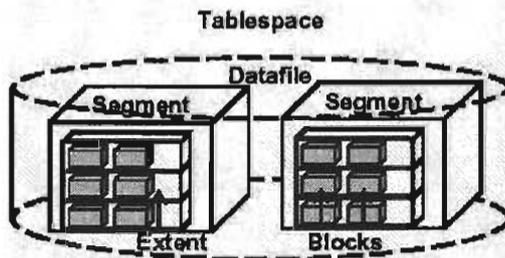


Figura 2.3 Relación entre estructuras lógicas

2.2.2 Estructura Física.

Consiste en el conjunto de archivos del sistema operativo, conocidos como data files, control files y redo log files, lo cuales proporcionan el almacenamiento físico actual para la información de la base de datos.

- **Data files:** todas las bases de datos oracle tienen uno o mas data files, los cuales contienen todos los datos. Los datos de las estructura lógicas como tablas, índices, son físicamente almacenados en los data files que son asignados a un tablespace. Las características de los data files son las siguientes:
 - Un data file solo puede asociarse con una base de datos.
 - Los data files pueden estar configurados para permitir su crecimiento automático es cuando se le termine el espacio inicial asignado.
 - Uno o más data files forman una unidad lógica del almacenamiento llamada tablespace.

Los datos en los data files son leídos, como se necesitan durante la operación normal de la base de datos.

- **Redo log files:** Toda base de datos oracle tiene un conjunto de dos o más grupos redo log. El conjunto de redo log files es colectivamente conocido como los redo log para la base de datos. La principal función del redo log es grabar todos los cambios hechos a los datos. Como una prevención a una a falla por ejemplo: que deje de funcionar el servidor, que se corrompan los archivos, o si se hicieron cambios a los datos pero antes de darle un commit (es decir que se graben permanentemente los cambios realizados a la información contenida a los datafiles). Los cambios permanentes a los datos son almacenados en los data files, y por consecuencia los cambios sin guardar pueden ser obtenidos de los redo log recuperando los datos sin perderlos. Como otra protección a fallas, Oracle permite multiplexar a los redo log files, es decir hacer copias de estos archivos y almacenarlas físicamente en otro dispositivo de almacenamiento. La información en los redo log file es usada solo para la recuperación de la base de datos en caso de falla.
- **Control files:** Un control file contiene información que especifica la estructura física de la base de datos, por ejemplo contiene la siguiente información: Nombre de la base de datos, nombre y lugar de almacenamiento de los data files y redo log files, tiempo de creación, etc. Se utiliza cada vez que una instancia de la base de Oracle es levantada, su control file identifica la base de datos y los redo log files que deben de ser abiertos para que la operación de la base de datos sea procesada. También el control file es usado para la recuperación.

El servidor de Oracle también utiliza otros archivos que no son parte de la base de datos pero que si son muy importante para su funcionamiento:

Parameter file: define las características de la instancia de Oracle. Es decir para que una instancia inicie el servidor Oracle debe leer un parameter file de inicialización, Por ejemplo, este archivo contiene parámetros del tamaño de algunas estructuras de memoria en la System Global Área.

Password file: identifica los privilegios de administrador de los usuarios.

Archive redo log files: son copias offline de los redo logs files que están online, los cuales son necesarios para la recuperación de la base de datos en caso de una de falla.

2.3 Instancia Oracle.

Toda base de datos Oracle que este corriendo esta asociada a una instancia Oracle, para que los datos puedan ser accedados, cuando una base de datos esta funcionando sobre un servidor de base de datos, Oracle levanta un conjunto de procesos de background Oracle y una área de memoria llamada System Global Area (SGA), esta combinación de procesos y memoria es lo que conforma una Instancia Oracle. La memoria y los procesos de una instancia manejan de una manera eficiente los datos de la base de datos. Una instancia puede abrir y usar una sola base de datos a la vez, es decir, un solo conjunto de archivos de base de datos como datafiles, redo log files, control file.

En la figura 2.4 se muestra los componentes de una instancia.

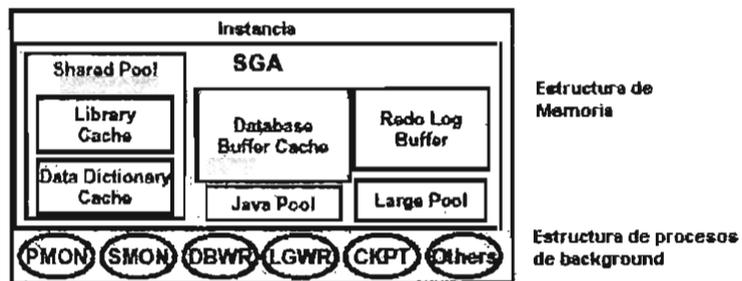


Figura 2.4 Instancia de base de datos oracle

2.3.1 Arquitectura de la Memoria.

Oracle utiliza memoria para almacenar información como:

- Código.
- Información acerca de sesiones, aunque no estén actualmente activas.

- Información que se necesita durante la ejecución del programa (por ejemplo, el estado actual del query que ya ha sido ejecutado).
- Información que es compartida y comunicada entre los procesos de Oracle.
- Almacenar datos en el cache.

La estructura básica de memoria asociada con Oracle incluye:

- System global área (SGA).
- Program Global Area (PGA).

System Global Area (SGA).

Es usada para almacenar información de la base de datos que es compartida por todos los servidores y los procesos de background, contiene datos e información de control para el servidor de oracle que se encuentra en la memoria virtual de la máquina en donde está Oracle corriendo.

La SGA es un grupo de estructura de memoria compartida que contiene datos e información de control para una instancia Oracle. Si múltiples usuarios están conectados a la misma instancia, entonces los datos en la memoria SGA de la instancia son compartidos entre los usuarios. La memoria SGA también se conoce como Shared Global Area.

La SGA y Procesos Oracle conforman una instancia, entonces cuando se inicia una instancia Oracle inmediatamente asigna memoria para la SGA, cada instancia tiene su propia SGA.

La SGA es de lectura y escritura. Todos los usuarios conectados a la instancia pueden leer información contenida dentro de la memoria SGA de la instancia y los procesos escriben en la SGA durante la ejecución de Oracle.

La SGA esta conformada de los siguientes elementos:

- Database buffer cache.
- Redo log buffer.
- Shared pool.
- Java pool (es configurable).
- Large pool (es configurable).

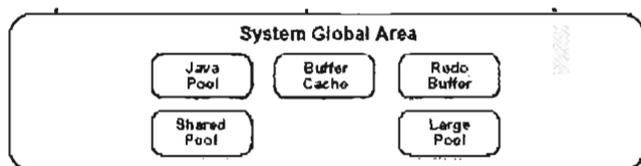


Figura 2.5 Componentes de la SGA

Database buffer cache: Es el elemento de la SGA que mantiene copias de los data blocks que son leídos de los data files. Todos los procesos de usuarios concurrentes conectados a la instancia comparten el acceso al database buffer cache.

Los buffers en el cache son organizados en dos listas:

- La lista de escritura.
- La lista más reciente usada (LRU).

La lista de escritura contiene los buffers sucios, que contienen datos que han sido modificados pero que no han sido escritos a disco. La LRU contiene los buffers libres que no contienen ningún dato útil.

La primera vez que un proceso de usuario requiere un dato en particular, busca el dato en el database buffer cache. Si el proceso encuentra el dato en la cache, puede leerlo directamente de memoria, pero si el proceso no encuentra el dato en la cache, el proceso debe de ir a buscar el dato a disco, haciendo una copia de los datablocks del datafile en disco dentro del buffer en la cache antes de acceder a los datos. Obviamente si los datos los encuentra en memoria el procedimiento será más rápido que si fuera a buscar hasta el disco.

Redo log buffer: Redo log es un buffer circular en la SGA que contiene información de los cambios hechos en la base de datos. Esta información es almacenada en las entradas redo. Las entradas redo contienen la información necesaria para reconstruir o leer, los cambios hechos a la base de datos como INSERT, UPDATE, DELETE, CREATE, ALTER, o DROP. Las entradas redo son usadas para la recuperación de la base de datos si es necesario.

Shared pool: Se utiliza para almacenar las sentencias SQL más recientemente ejecutadas y las definiciones de datos más recientemente utilizadas.

La shared pool de la SGA contiene dos principales áreas:

- Library cache.
- Data dictionary cache.

La library cache Almacena información acerca de las sentencias SQL y PL/SQL utilizadas recientemente.

Si el tamaño del shared pool es muy chico las sentencias son continuamente releídas en el library cache, pero esto afecta el desempeño. Si las sentencias ya no se han utilizado son borradas del library cache.

El library cache tiene la siguiente estructura:

- La shared SQL: almacena y comparte el plan de ejecución y analiza que las sentencias SQL se ejecutan de nuevo en la base de datos. La segunda vez que una sentencia idéntica esta corriendo, se puede tomar ventaja y ser ejecutada desde la memoria. Para asegurarse de que un sentencia SQL utilice el Shared SQL área siempre debe ser idéntica en estructura a la que se encuentra almacenada en memoria, por ejemplo se comparan elementos como texto, esquema, además de que las variables tienen que ser las mismas.
- El área shared PL/SQL almacena y comparte las sentencias PL/SQL recientemente ejecutadas; analiza y compila programas y procedimientos (funciones, paquetes y triggers), que son almacenados en esta área.

Data dictionary cache: también conocido como dictionary cache o raw cache. La información de la base de datos es almacenada en las tablas del diccionario de datos, como cuentas de usuario, nombres de data files, posiciones de los extents, descripciones de tablas y privilegios de usuarios.

Cuando esta información es requerida por el servidor de base de datos, las tablas del diccionario de datos son leídas, y los datos que son regresados son almacenados en el data dictionary cache. Oracle consulta el diccionario de datos frecuentemente durante el análisis de las sentencias SQL. Este acceso es esencial para que la operación de Oracle continúe.

Large pool: El administrador de base de datos puede configurar un área de memoria opcional llamada large pool para proveer asignaciones de memoria para:

- Procesos de entrada / salida.
- Operaciones de Backup.
- Sesiones de memoria para servidores compartidos.
- Ejecuciones en paralelo.

La large pool es mejor para satisfacer requerimientos de memoria muchos más grande que el shared pool.

Java Pool: Analiza los requerimientos de los programas de java, es requerido si se ha instalado Java y está en uso, además maneja el uso de memoria java utilizada por un programa java cuando esta en ejecución como un instalador, un asistente o alguna herramienta que se base en java.

SGA Dinámica.

La SGA Dinámica surge con la versión de Oracle 9i, una SGA dinámica implementa una infraestructura que permite que la configuración de la SGA pueda cambiar sin bajar la instancia, esto permite que el tamaño de database buffer cache, shared pool, large pool cambien sin interrumpir el servicio del RDBMS.

Program Global Area (PGA).

PGA es una región de memoria que contiene datos e información para un proceso de servidor o para un proceso de background, la cual es útil para el control de los procesos. La PGA es una memoria no compartida por Oracle cuando un proceso es iniciado. El acceso a esta memoria es exclusivo para cada proceso y es solo de lectura para el servidor de Oracle.

En contraste con la SGA que es compartida por muchos procesos, la PGA es asignada a un solo proceso, cuando este es creado y desaparece cuando el proceso termina.

La figura 2.6 muestra el papel que juegan las estructuras de memoria mencionadas anteriormente, en una instancia:

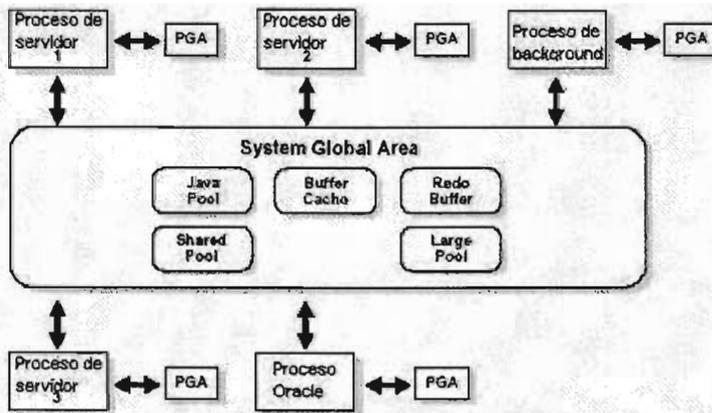


Figura 2.6 Relación entre estructuras de memoria

2.3.2 Arquitectura de los Procesos.

Todos los usuarios de oracle deben correr dos módulos de código para acceder a una instancia de bases de datos oracle:

- Aplicación o herramientas oracle: un usuario de la base de datos corre un aplicación de base de datos, como un programa pre-compilador o una herramienta oracle, como SQL*PLUS.

- Código del servidor de oracle: cada usuario tiene algún código del servidor de base de datos ejecutándose que procesa e interpreta la aplicación que utiliza sentencias SQL para utilizar la base de datos.

Estos módulos son ejecutados por proceso. Un proceso es un mecanismo en un sistema operativo que puede correr una serie de pasos. Un proceso normalmente tiene su propia área de memoria privada en donde corre el proceso.

Multi-Procesos en los Sistemas Oracle

Los multi-procesos usan muchos procesos para correr partes diferentes del código de Oracle y procesos adicionales para los usuarios, como por ejemplo un proceso para cada usuario conectado o muchos procesos compartidos para múltiples usuarios.

Muchos otros sistemas manejadores de bases de datos son multiusuarios, porque uno de los principales beneficios de este sistema es que se manejen datos para múltiples usuarios al mismo tiempo. Cada proceso en una instancia de Oracle se encarga de un trabajo específico. Dividiendo el trabajo del servidor de Oracle y las aplicaciones de base de datos dentro de los procesos, pueden conectarse al mismo tiempo varios usuarios y aplicaciones mientras que el sistema mantiene un desempeño aceptable.

Los procesos en los sistemas Oracle pueden dividirse en dos grandes grupos:

- Procesos de usuarios que corren la aplicación o el código de las herramientas Oracle.
- Procesos Oracle que corren el código del Servidor Oracle. Incluyen procesos del servidor y procesos de background.

La estructura de los procesos varía para diferentes configuraciones de Oracle dependiendo del sistema operativo y de la elección de las opciones de Oracle por ejemplo, el código para la conexión de los usuarios puede ser configurado como servidor dedicado o como servidor compartido, más adelante en el tema 2.4 CONFIGURACIONES DE UN SERVIDOR DE BASE DE DATOS ORACLE, se explicara como los procesos de usuario y servidor son diferentes para cada configuración de la base de datos

Procesos de Usuario.

Cuando un usuario corre un programa de aplicación como un Pro*C o una herramienta de oracle como el Enterprise Manager o SQL*PLUS, Oracle crea un proceso de usuario para correr la aplicación del usuario.

Conexiones y Sesiones.

Son conceptos muy relacionados con los procesos del usuario pero tienen un significado muy diferente.

Una conexión es una ruta de comunicación entre los procesos de usuario y la instancia Oracle, es decir, cuando hay una conexión una ruta de comunicación se establece usando mecanismos de procesos disponibles, es decir en los siguientes casos:

- Cuando sobre una computadora se ejecutan todos los componentes: los procesos de usuario, el servidor de Oracle, es decir, cliente y servidor en la misma computadora, los componentes se comunican localmente.
- Cuando una aplicación de base de datos y el servidor de Oracle se ejecutan en diferentes computadoras, es decir, cliente en una máquina diferente del servidor, los cuales se pueden comunicar a través de la red.

Un usuario de base de datos se puede conectar al servidor de base de datos en tres formas diferentes:

- El usuario se identifica en sistema operativo corriendo una instancia de Oracle e inicializa una aplicación o herramienta para acceder la base de datos en este sistema.
- El usuario abre una aplicación o herramienta en la máquina local y se conecta a la instancia de Oracle a través de la red. Esta configuración es cliente – servidor y se utiliza un software de red para la comunicación entre el usuario y la base de datos.
- Una conexión de tres capas, la computadora del usuario se comunica a través de la red con una aplicación o a un servidor de red, el cual es conectado a través de la red con el servidor que tiene la instancia de Oracle corriendo.

Una sesión es una conexión específica de un usuario a una instancia Oracle a través del proceso de usuario. Una sesión se crea cuando un usuario se valida al servidor de Oracle y finaliza cuando este usuario termina la sesión normalmente o cuando la sesión termina anormalmente por alguna falla del sistema.

Por ejemplo, cuando un usuario abre el SQL*Plus, debe tener un nombre de usuario y un password, entonces se establece una sesión para ese usuario. Una sesión finaliza cuando el usuario se desconecta.

Se pueden crear varias sesiones para un solo usuario, usando el mismo nombre de usuario, por ejemplo, un usuario con el nombre de usuario/password SCOTT/TIGER, puede conectarse a la misma instancia Oracle muchas veces, utilizando herramientas, aplicaciones o terminales al mismo tiempo. Oracle crea un

proceso por cada sesión de usuario en una configuración de servidor dedicado, pero con el servidor compartido, muchas sesiones de usuario pueden compartir un solo proceso.

Listener.

Cuando una instancia inicializa el proceso de listener establece la ruta de comunicación con oracle. Cuando los procesos de usuarios hacen un requerimiento de conexión, el listener determina a qué instancia se debe conectar y establece una conexión apropiada.

El listener también establece una ruta de comunicación entre bases de datos. Cuando diferentes bases de datos o instancias corren en una computadora, como real application cluster, los servicios de nombres habilitan instancias para registrarse automáticamente con otros listeners en la misma máquina. Un servicio de nombre puede identificar múltiples instancias, y las instancias pueden pertenecer a múltiples servicios. Los clientes conectados al servicio no tienen que especificar a qué instancia se quieren conectar.

Registro de Información del Servicio.

Un registro dinámico de servicio reduce la administración de múltiples de base de datos o instancias. La información acerca de servicios requeridos por el cliente son registrados con el listener. Dicha información puede ser registrado dinámicamente por el listener a través de la característica llamada registro de servicio o configuración estática en el archivo listener.ora.

El registro de servicio se basa sobre en el proceso pmon (un proceso de background de la instancia) para registrar la información de la instancia con el listener, así como el estado actual y la carga de la instancia y los dispatchers del servidor compartido. La información registrada habilita al listener para regresar los requerimientos del cliente para proporcionar el servicio. El registro de información no requiere una configuración en el archivo listener.ora.

Cuando se inicia una instancia se identifica con el listener de otras instancias a las que pertenecen al mismo servicio. Durante el funcionamiento de la base de datos, las instancias de servicio pasan información a través del CPU y una conexión actual cuenta todos los listeners en el mismo servicio. Esto habilita el balanceo de cargas y la conexión failover, los cuales son dos conceptos muy importantes para el funcionamiento de Real Application Cluster.

Procesos de Oracle.

El servidor de Oracle corre dos tipos de procesos: procesos del servidor y procesos de background.

Procesos de Servidor.

En la configuración más básica de un servidor de Oracle, cuando un usuario se identifica en el servidor de Oracle un proceso es creado en la computadora donde se encuentra el servidor Oracle, este proceso es llamado "proceso de Servidor", este servidor se comunica con la instancia detrás de los proceso de usuarios que corren en el cliente.

Oracle crea procesos de servidor para soportar los requerimientos de los procesos de usuarios conectados a la instancia. En algunas situaciones cuando las aplicaciones y oracle opera en la misma máquina, es posible que se combinen los procesos de usuarios y los correspondientes procesos de servidor en un solo proceso para reducir la carga de procesamiento al sistema. Pero si la aplicación y oracle operan en máquinas diferentes, los procesos de usuario siempre se comunican con Oracle a través de procesos separados de servidor.

Los procesos de servidor creados para cada usuario de aplicación pueden mejorar el funcionamiento de uno o más de los siguientes:

- Correr sentencias SQL a través de la aplicación.
- Leer los necesarios datablocks de los datafiles sobre disco dentro de los buffers database compartidos de la SGA, si los data blocks no existen aún en la SGA.
- Regresar resultados en cuanto la aplicación pueda procesar la información.

La figura 2.7 muestra los pasos que sigue un requerimiento de conexión a la base de datos mediante la comunicación entre el proceso de usuario y el proceso de servidor, el cual se comunica con la instancia.

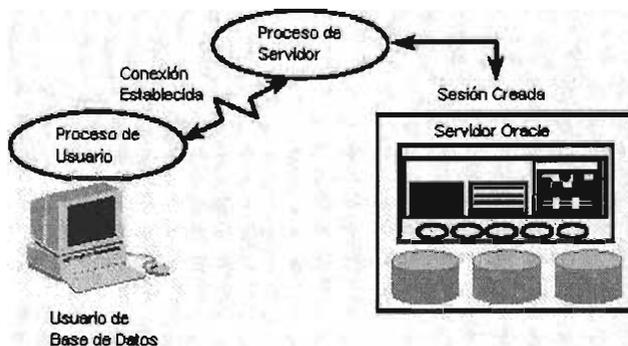


Figura 2.7 Sesión y Conexión al RDBMS

Procesos de Background.

Para mejorar el rendimiento y gestionar los recursos de la base de datos, el sistema multiprocesamiento Oracle utiliza algunos procesos adicionales, llamados procesos de background. Una instancia Oracle puede tener muchos procesos de background aunque no siempre todos están presentes, los procesos de background en una instancia Oracle son los siguientes:

- Database Writer (DBW0 o DBWn).
- Log Writer (LGWR).
- Checkpoint (CKPT).
- System Monitor (SMON).
- Process Monitor (PMON).
- Archiver (ARCn).
- Recoverer (RECO).
- Lock Manager Server (LMS), solo Real Application Clusters.
- Queue Monitor (QMNn).
- Dispatcher (Dnnn).
- Server (Snnn).

En muchos sistemas operativos, los procesos de background son creados automáticamente cuando una instancia se inicia.

A continuación explicaremos el funcionamiento de los procesos de background:

Database Writer Process (Dbwn).

El proceso DBWn escribe el contenido de los buffers hacia a los datafiles. El proceso DBWn es responsable de escribir buffers contenidos en la database buffer cache hacia disco, esto asegura que haya un número suficiente de buffers libres disponibles en el DataBase Buffer Cache. Aunque un solo proceso DBW0 es suficiente para la mayoría de los sistemas, esto depende del número de información y transacción que se maneje en la base de datos, se pueden configurar procesos adicionales (DBW1 al DBW9) para mejorar el rendimiento en el sistema. Cuando un buffer en la database buffer cache es modificado, se le llama usados.

El proceso DBWn escribe buffers usados cuando se cumplen las siguientes situaciones:

- Cuando un proceso de servidor no puede encontrar un buffer limpio para volver a utilizarlo.
- DBWn escribe periódicamente a los buffers cuando ocurre un checkpoint.

El proceso ocurre cuando:

- Ocurre un checkpoint.
- El número de buffers ocupados sobrepasa el valor establecido.

- Ya no hay buffers limpios.
- Ocurre un timeout.
- Se borra un tablespace.
- Los tablespace están fuera de línea.

Log Writer Process (LGWR).

El proceso LGWR es responsable del manejo del redo log buffer, escribiendo el redo log buffer al archivo de redo log en disco. LGWR escribe todas las entradas redo que han sido copiadas en el buffer desde la última vez que se escribió.

El redo log buffer es circular. Cuando LGWR escribe las entradas redo del redo log buffer hacia el archivo de redo log, el proceso de servidor puede entonces copiar nuevas entradas al final de las entradas en el redo log buffer que han sido escritas en disco. LGWR normalmente escribe lo suficientemente rápido para asegurarse que el espacio esté siempre disponible en el buffer para nuevas entradas.

LGWR escribe en una porción contigua del buffer al disco. El proceso LGWR escribe a los archivos de redo log cuando:

- Sucede un commit.
- Cuando sucede un checkpoint.
- Cada 3 segundos.
- El redo log buffer está lleno.
- Un proceso DBWn escribe buffers modificados a disco si es necesario.

Checkpoint Process (CKPT).

Cada tres segundos el proceso de CKPT almacena datos en archivo de control para identificar el lugar en los archivos online redo log, donde se iniciará la recuperación. El propósito de un checkpoint es asegurar que todos los buffers en el DataBase Buffer Cache que hayan sido modificados antes del punto en tiempo sean escritos en los archivos de datos. Este punto en tiempo es cuando la recuperación de la base de datos empieza cuando una falla ocurra.

Un checkpoint es iniciado en las siguientes situaciones:

- Para asegurarse que los data blocks modificados en memoria sean escritos a disco regularmente y que estos datos no se pierdan en caso de que el sistema falle.
- Para reducir el tiempo requerido para la recuperación de la instancia.
- Asegurarse de que todos los datos con commit hayan sido escritos a los archivos de datos cuando se está bajando la base de datos.
- Cuando un checkpoint ocurre, Oracle debe actualizar las cabeceras de los datafiles para grabar los detalles del checkpoint. Esto es hecho por el proceso CKPT. EL proceso CKPT no escribe bloques al disco, el DBWn siempre realiza este trabajo.

La información que un checkpoint incluye son: la posición del checkpoint, número de cambio del sistema, posición en los archivos online redo log en el cual se empezará la recuperación, información acerca de los logs.

System Monitor Process (SMON).

El proceso SMON hace la recuperación si es necesaria, cuando la instancia inicia. SMON también es responsable de limpiar los segmentos temporales y de juntar los extents libres de una manera contigua dentro del manejo de tablespaces.

Si alguna transacción terminada fue saltada durante la recuperación de la instancia por alguna circunstancia, SMON la recupera cuando el tablespace o archivo es llevado de vuelta en línea.

El proceso de recuperación actúa de la siguiente manera:

- Cuando se detiene una instancia por causa de alguna falla, los datos que no fueron grabados en los data files, pero si fueron grabados en los archivos online redo log, Porque todas las transacciones de commit deben de haberse grabado en los archivos de redo log. En el momento que se inicia la instancia nuevamente el proceso SMON lee los archivos online redo log y aplica los cambios grabados en los archivos online redo log a los data blocks.
- SMON checa la regularidad cuando es necesario.
- Otro proceso puede llamar a SMON si se detecta que lo necesitan.
- En Real Application Cluster, el proceso de SMON de una instancia puede funcionar como instancia de recuperación por falla de CPU o falla de instancia.

Process Monitor Process (PMON).

El proceso PMON funciona como proceso de recuperación cuando un proceso de usuarios falla. PMON es responsable de limpiar el database buffer cache y de la liberación de recursos que los procesos de usuarios estaban usando. PMON checa periódicamente el status del dispatcher y los procesos del servidor. PMON también registra información acerca de la instancia y el proceso dispatcher con el listener de red.

Recoverer Process (RECO).

Es un proceso RECO es usado para la configuración distribuida de la base de datos que resuelve automáticamente fallas que tengan que ver con las transacciones distribuidas en duda. El proceso RECO de un nodo conecta automáticamente a otras bases de datos involucradas en las transacciones distribuidas. Cuando el proceso de RECO reestablece la conexión entre las bases de datos involucradas, automáticamente resuelve todas las transacciones en duda, removiendo de cada base de datos cualquier renglón que corresponda a la resolución de transacciones en duda de la tabla de transacciones pendientes.

Si el proceso RECO falla en la conexión con el servidor remoto, RECO automáticamente trata de conectarse a través después de un determinado intervalo de tiempo. De cualquier forma RECO espera un tiempo antes de lanzar otra conexión. El proceso RECO esta presente, solo si la instancia permite transacciones distribuidas. El número de transacciones distribuidas concurrentes no tiene límite.

Archiver Processes (ARCn).

El proceso ARCn es opcional, aunque de todas formas es crucial para la recuperación de la base de datos después de una pérdida de disco. Cuando un archivo de online redo log está lleno, el servidor de Oracle empieza a escribir en el otro redo log, este proceso de cambio de un redo log a otro se llama log switch.

El proceso ARCn copia los archivos de redo en línea en un dispositivo de almacenamiento designado después de que un switch de log ocurra. Los procesos ARCn están presentes solo cuando la base de datos está en modo ARCHIVELOG, y cuando el automatic archiving es habilitado.

La base de datos puede estar en modo NOARCHIVELOG o ARCHIVELOG.

Modo NOARCHIVELOG: los archivos de redo log son sobre escritos cada vez que un log switch ocurre. El proceso de LGWR no sobre escribe los archivos de redo log en línea hasta que un checkpoint ocurra para que ese grupo sea completado. Esto asegura que los datos guardados no se pierdan si la instancia falla. Si la instancia falla se pierde solo la SGA, es decir solo hay pérdida de memoria y no disco.

Modo ARCHIVELOG: si la base de datos esta en modo archive log los grupos de redo log que estén inactivos y llenos deben ser archivados antes de que puedan ser utilizados otra vez. Porque los cambios a la base de datos son grabados en los archivos de redo log en línea. El administrador de la base de datos puede utilizar un backup físico de los archivos de datos y los archivos de redo log que hayan sido respaldados, para recuperar la base de datos sin ninguna pérdida de datos.

Una instancia puede tener hasta 10 procesos de ARCn (de ARC0 hasta ARC9). El proceso LGWR inicializa un proceso nuevo de ARCn siempre que el número actual de proceso ARCn sea insuficiente para aguantar la carga de datos. El archivo de alerta guarda un registro de cuando LGWR inicializa un nuevo proceso de ARCn.

Lock Manager Server Process (LMS).

Proporciona un manejo de recursos entre instancias, más adelante en el capítulo 3 tocaremos este punto.

Queue Monitor Processes (QMn).

El proceso QMn es opcional para Oracle Advanced Queuing, que monitorea los mensajes de cola, se puede configurar hasta 10 procesos de monitoreo. Estos procesos son diferentes de otro procesos de background, si este proceso falla no provoca que la instancia se caiga.

La figura 2.8 muestra la interrelación de los componentes que forman un sistema manejador de bases de datos Oracle.

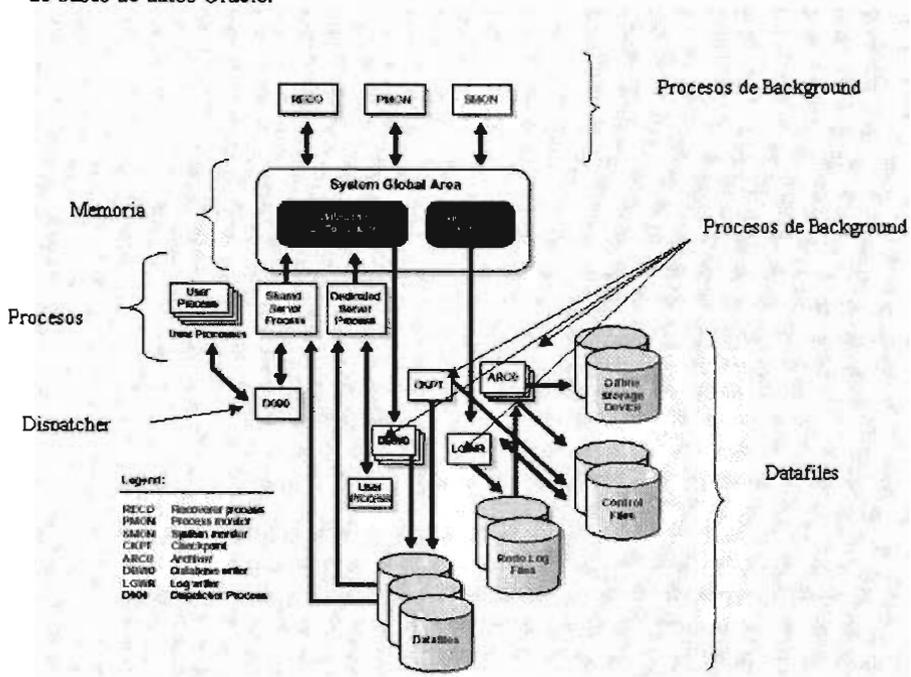


Figura 2.8 RDBMS ORACLE

2.4 Configuraciones de un Servidor de Base de Datos Oracle.

2.4.1 Servidor Dedicado.

El servidor Oracle se puede configurar de modo dedicado o compartido para atender requerimientos como conexiones de los procesos de usuario. A continuación explicare cada configuración.

Cuando los usuarios y procesos de servidor están separados se dice que existe una configuración de servidor dedicado. Los procesos de servidor separados creados detrás de cada proceso de usuario son llamados procesos de servidor dedicado, porque los actos de este proceso de servidor solo están sobre los procesos asociados a los usuarios.

Esta configuración mantiene una relación de uno a uno entre el número de procesos de usuario y los procesos de servidores. Aunque cuando el usuario no esta activamente haciendo un requerimiento a la base de datos, los procesos de servidor dedicado lo mantiene como activo.

En una configuración de un servidor dedicado, el usuario y los procesos de servidor se comunican usando diferentes mecanismos:

- Si el sistema es configurado entonces los procesos de usuarios y los procesos de servidor dedicado corren en la misma maquina, la interfaz de programa usa los procesos internos del sistema operativo como un mecanismo de comunicación para realizar su trabajo.
- Si los procesos de usuarios y los procesos del servidor dedicado corren en diferentes computadoras, la interfaz de programa proporciona los mecanismos de comunicación (como el software de red y Oracle Net Services) entre los programas.

La figura 2.9 muestra como en una configuración de servidor dedicado un proceso de usuario se comunica solo con un proceso de servidor, es decir, la relación es uno a uno, una vez que el proceso de servidor tenga el requerimiento del usuario este se comunica con la instancia, el cual espera la respuesta de dicha instancia.

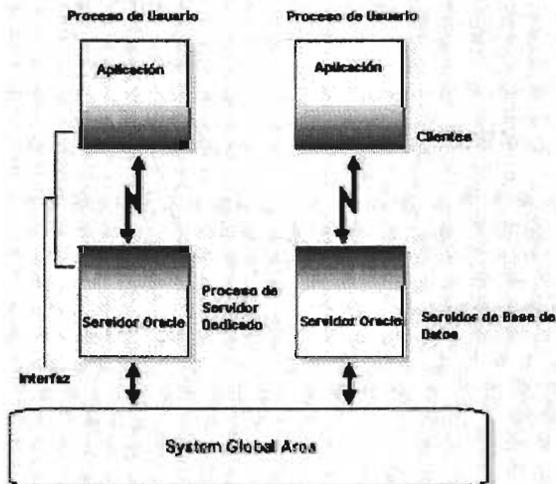


Figura 2.9 Proceso Usuario – Proceso de Servidor

La figura 2.10 muestra la configuración de una instancia en servidor dedicado. Cada usuario conectado tiene un proceso de usuario.

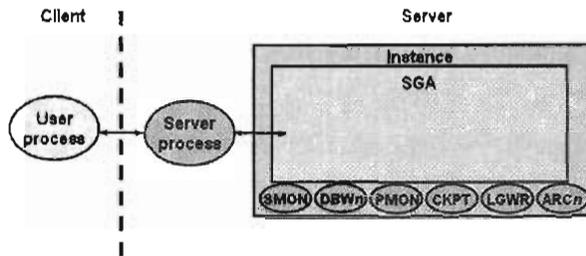


Figura 2.10 Servidor Dedicado

2.4.2 Servidor Compartido

La arquitectura de un servidor compartido elimina la necesidad de tener un proceso de servidor para cada conexión. Un dispatcher direcciona las sesiones múltiples de red a un pool de procesos del servidor.

Un solo proceso de servidor compartido puede recoger los requerimientos de varios usuarios. Entonces, un número pequeño de servidores compartidos funcionan igual a muchos servidores dedicados, además de que se ahorra recursos porque solo se requiere la memoria por cada usuario conectado ya que un solo un proceso de servidor atenderá el número de usuarios que se le configure, por esto requiere menos memoria y manejo de procesos en una configuración de servidor compartido que en uno dedicado, además de que pueden ser soportados más usuarios.

El número de procesos que son necesarios en un sistema de servidor compartido:

- Dispatcher (Dnnn): Este proceso soporta configuración de servidor compartido permitiendo a los procesos de usuarios compartir un número limitado de proceso de servidor. Con el servidor compartido, pocos procesos de servidor compartido son requeridos para el mismo número de usuarios. Por lo tanto, el servidor compartido puede soportar un gran número de usuarios, particularmente en ambientes de cliente / servidor donde la aplicación del cliente y el servidor operan sobre diferentes máquinas.

Se pueden crear múltiples procesos de dispatcher para una sola instancia. El último dispatcher debe ser creado para cada protocolo de red usado por Oracle. El administrador de la base de datos inicializa una cantidad óptima de procesos de dispatcher dependiendo de la limitación del sistema operativo en el número de

conexiones para cada proceso, y el poder agregar o eliminar procesos de dispatcher mientras la instancia esta corriendo.

- **Listener:** En una configuración de servidor compartido, un proceso que se llama listener de red espera por los requerimientos de conexión de las aplicaciones del cliente y las rutas de cada proceso de dispatcher. Si este no puede conectar a la aplicación del cliente al dispatcher, el proceso de listener inicializa un proceso de servidor dedicado, y conecta a la aplicación del cliente al servidor compartido. El proceso de listener no es parte de la instancia de Oracle, pero si es parte del proceso de red que trabaja con Oracle.
- **Shared Server (Snnn):** Cada proceso de servidor compartido sirve a múltiples requerimientos de clientes en la configuración de servidor compartido. Los procesos de servidor compartido y los procesos de servidor dedicado proporcionan la misma funcionalidad, exceptuando que los procesos de servidor compartido no están asociados con un proceso de usuario específico. Los procesos de servidor compartido le dan servicio a cualquier requerimiento del cliente en una configuración de servidor compartido.

Cuando una instancia inicializa, el listener de red se abre y establece una ruta de comunicación a través de los usuarios conectados a oracle. Entonces cada proceso de dispatcher provee al proceso de listener una dirección en la cual el proceso de dispatcher escucha los requerimientos de conexión.

Cada uno de los procesos de dispatcher debe ser configurado e inicializado para cada protocolo de red que los usuarios de la base de datos utilizarán.

Cuando un proceso de usuario hace un requerimiento de conexión, el listener examina el requerimiento y determina si el proceso de usuario puede usar un proceso de servidor compartido.

El listener regresa la dirección del proceso de dispatcher que tiene la carga más pesada, y el proceso del usuario que conecta directamente al dispatcher que tenga menos procesos de usuario asignados.

Si algunos de los procesos de los usuarios no se pueden comunicar con el dispatcher, entonces el proceso de red listener no puede conectarlos al dispatcher. En este caso, o si el requerimiento de proceso de usuario es de servidor dedicado, el listener crea un servidor dedicado y establece una conexión apropiada.

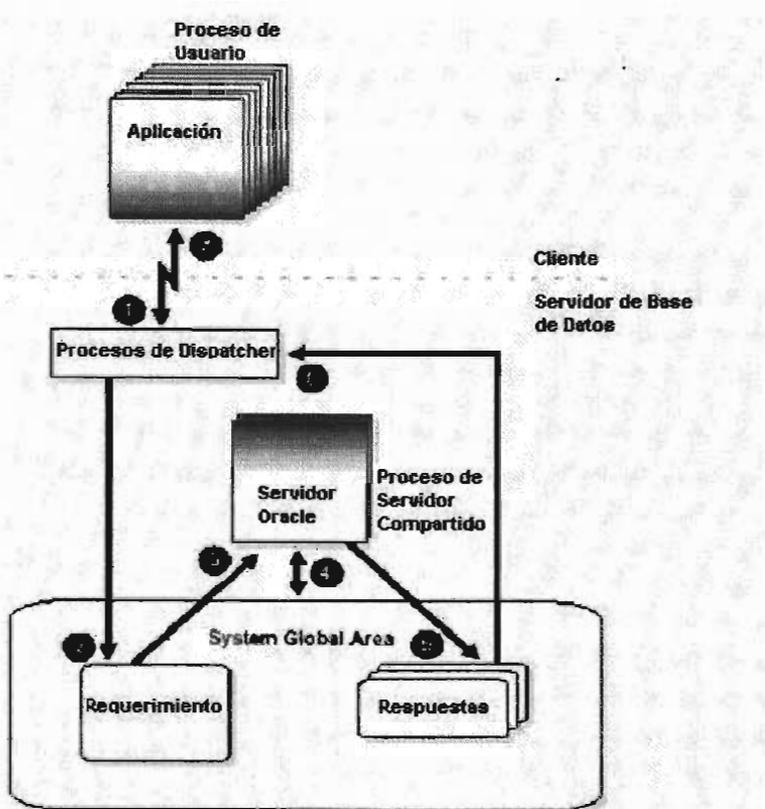


Figura 2.12

CAPÍTULO 3: ORACLE REAL APPLICATION CLUSTER.

Ahora que ya explique la arquitectura y el funcionamiento de la base de datos Oracle como una instancia, entrare por completo a los conceptos de RAC. La configuración de una base de datos Oracle en RAC es diferente a la configuración de una base de datos standalone, sin embargo se utilizan los mismos conceptos, por consiguiente es indispensable entender primero la arquitectura standalone.

Los componentes de la arquitectura para una base de datos en RAC, no cambian por completo, como lo mencione anteriormente solo se generalizan para tener un esquema de alta disponibilidad.

El software de RAC junto con el hardware de un cluster crean el cluster de Oracle.

RAC es una opción de Oracle que permite tomar ventajas del hardware de cluster corriendo múltiples instancias con una sola base de datos, es decir RAC se basa en la arquitectura de shared-disk cluster.

Los archivos físicos de la base de datos son almacenados en discos que están física o lógicamente conectados a cada nodo, es decir compartidos, para que cada instancia activa pueda leer o escribir en ellos.

El software de RAC maneja el acceso de datos, coordinándolos entre las instancias y cada instancia ve una imagen consistente (es decir todas las instancias ven la misma información) de la base de datos.

RAC proporciona el poder de procesamiento de varias computadoras interconectadas. El software de RAC junto con un cluster de hardware certificado o autorizado por la compañía del software para dar soporte a sus productos, es decir, que crean un "cluster de base de datos". Esta combinación de software y hardware unifica el poder de procesamiento de cada computadora en la configuración del cluster creando una plataforma robusta y escalable.

En un ambiente de RAC todas las instancias activas pueden ejecutar transacciones concurrentes usando una base de datos compartida. RAC coordina cada instancia para tener acceso y compartir los datos para proporcionar consistencia e integridad de los mismos.

Una tarea grande dividida en varias sub tareas y distribuidas entre múltiples servidores es completada más rápido y más eficientemente que si la tarea fuera ejecutada en un solo servidor.

Con RAC es posible escalar aplicaciones para incrementar el procesamiento de datos sin cambiar el código de aplicación. Conforme se agreguen recursos como servidores o contenedores de discos RAC aumenta el poder de procesamiento de estos recursos.

3.1 Arquitectura de Hardware: Shared-Disk Cluster para Real Application Cluster.

En esta siguiente sección describire a detalle los componentes de hardware que el RAC, (cluster de base de datos) necesita para funcionar de una manera adecuada.

El cluster de base de datos comprende dos o más nodos, que son PC'S, servidores, mainframe o cualquier máquina en donde pueda correr una instancia de la base de datos, estos nodos se tienen que comunicar de alguna forma, para esta comunicación existe un dispositivo llamado interconnect, el cual establece una ruta entre cada nodo de la base de datos en cluster, físicamente este interconnect es un cable, en caso de que solo sean dos nodos y puede ser de fibra óptica o cable cruzado. Cuando hay más de dos nodos en el cluster, se establece una red privada donde la comunicación debe ser de alta velocidad, para que sea rápida y eficiente.

Técnicamente el interconnect es empleado por cada instancia para la sincronización de datos entre el cluster, que utilizan los recursos compartidos. Oracle también utiliza el interconnect para transmitir data blocks que comparten las distintas instancias, por ejemplo, el primer tipo de recursos compartidos son los data files a los cuales tienen acceso todos los nodos al mismo tiempo.

En la siguiente figura se muestran los componentes que un cluster de base de datos Oracle debe tener. Como ya había mencionado, si se tienen más de dos nodos es necesario tener una red privada, por lo que se requiere un hub o un switch; todos estos nodos deben de tener acceso a un sistema de almacenamiento compartido, porque todos deben ver el mismo arreglo de discos, como lo muestra la figura 3.1:

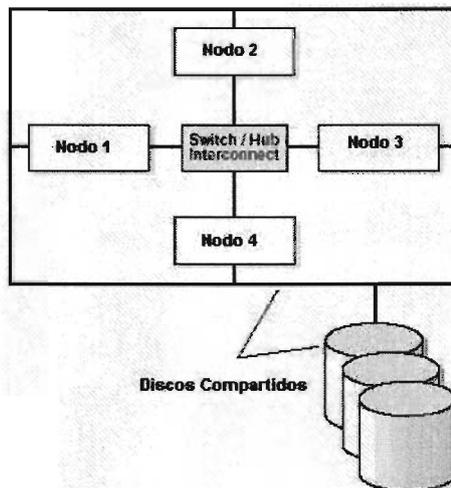


Figura 3.1 Discos Compartidos.

3.1.1 Nodo.

Cada nodo en el cluster consiste en una o más unidades de procesamiento central (CPU), memoria y posiblemente uno o más dispositivos de almacenamiento locales. Los nodos deben de estar habilitados para conectarse con cada nodo que forme parte del cluster, y con el subsistema de discos compartidos. Los nodos que participen en el cluster pueden funcionar como servidores independientes usando su propio dispositivo de almacenamiento y conexiones estándar de red para ser accedados por aplicaciones cliente.

Los nodos en el cluster no tienen que tener las mismas características de hardware pero si deben de tener el mismo sistema operativo. El número total de nodos en el cluster soportado por la base de datos en Oracle Real Application Clusters depende de la combinación de hardware, algunos sistemas restringen al número de nodos a 2 como máximo y otros permiten hasta 16, es variable.

3.1.2 Interconnect Cluster Interconnect (IC) e Interprocess Communication (IPC).

El IPC es un componente de alta velocidad que utiliza RAC para la comunicación entre los nodos. El IPC define el protocolo y la interfaz requerida para la comunicación, con el cual se puede transferir información entre las instancias que forman el cluster.

Cada nodo en cluster necesita guardar información de su estado y de la configuración de otros nodos, esto se hace periódicamente por una transmisión de un mensaje de red, llamado heartbeat, a través de la red. La señal heartbeat es comúnmente enviado a través de una red privada, el cluster interconnect, que es usado para la comunicación entre nodos.

El cluster interconnect es construido instalando tarjetas de red en cada nodo y conectándolas con un cable adecuado de red y configurando un protocolo que corra a través del cable. Se puede utilizar como interconnect una Ethernet, Fiber Distributed Data Interface (FDDI), u otro hardware para la interconexión.

Se debe considerar instalar un backup del interconnect en caso de que el interconnect primario falle, reduciendo la probabilidad de que el interconnect se convierta en un punto de falla.

La figura 3.2 muestra el interconnect entre dos maquinas.

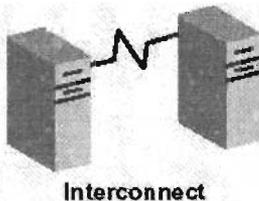


Figura 3.2 Interconnect.

3.1.3 Subsistema de Discos Compartidos.

Una base de datos en RAC debe ser accesible para todas las instancias que están corriendo en diferentes nodos. La mayoría de los file systems (sistemas de archivos) no permiten el acceso a múltiples servidores para que abran y escriban archivos.

Una base de datos en RAC requiere que todos los nodos tengan acceso simultáneo a los discos de almacenamiento compartidos para dar a las instancias acceso concurrente a la base de datos. Esto requiere de un formato especial de sistema de archivos.

La implementación del subsistema de discos compartidos, está basado en el sistema operativo donde será instalado el RAC.

En la mayoría de las plataformas Real Application Clusters se requiere que cada instancia pueda acceder a un conjunto de discos sin formato en un subsistema de discos compartidos. Estos discos compartidos pueden hacer referencia a los raw devices o cluster file system.

Si la plataforma soporta un Cluster File System (CFS) que esté certificado por Oracle se pueden almacenar los archivos que Real Application Clusters requiere directamente en el CFS.

Un CFS es un sistema de archivos que pueden ser accedados (lectura y escritura) por todos los miembros del cluster al mismo tiempo. Esto implica que todos los miembros del cluster puedan ver los mismos datos. Ejemplos de CFS en el mercado son RMS para el Sistema Operativo HP OpenVMS, Veritas Cluster File System para plataforma Unix, Oracle Cluster File System para plataformas Linux y Windows. Para plataforma IBM General Parallel File System (GPFS) con su sistema operativo AIX.

Las instancias de Oracle en Real Application Clusters utilizan raw devices o CFS para almacenar los archivos de la base de datos como controlfile, datafile, redologfile, archivo de parámetros, archivo de password, todas las instancias en el cluster tienen que compartir estos archivos. Los raw devices deben ser creados antes de que la base de datos sea creada, usando comandos específicos del sistema operativo.

Las tecnologías de almacenamiento compartido en las cuales se han implementado Real Application Clusters son las siguientes:

SCSI: Los dispositivos de discos son conectados individualmente a la máquina utilizando una pequeña interfaz de sistema de computadora a través de los controladores.

SAN (Storage Área Network): Proporciona un almacenamiento compartido a través de la red sirviendo como un respaldo eficiente para los servidores de la misma, además de que utiliza una conexión de alta velocidad.

NAS (Network Attached Storage): Es un servidor con propósito especial con su propio software que permite compartir archivos a través de la red.

Acceso A Discos.

La diferencia entre una base de datos en un cluster y una normal es el requerimiento de acceso a los discos compartidos. En una base de datos normal solo una instancia necesita tener acceso a los archivos de la base de datos al mismo tiempo así que estos archivos pueden estar almacenados en un disco al cual la base de datos tenga un acceso rápido, este disco puede estar en el mismo servidor o en un arreglo de discos externos. En RAC diferentes instancias necesitan leer y escribir en los archivos de la base de datos (control file, redo log file, data file, server parameter file), es por eso que estos archivos deben ser almacenados en un sistema de discos compartidos que esté disponible para todos los nodos.

Los archivos de base de datos incluyen uno o más control files, un conjunto de online redo log files, los archive files son opcionales y data files. Se puede utilizar un archivo para facilitar la administración de parámetros, Sever parameter file, el cual es almacenado en los discos compartidos. Por lo tanto, se debe proporcionar acceso a los discos compartidos a todos los archivos físicos de la base de datos.

El tener acceso al disco permite utilizar ciertas características opcionales con Real Application Clusters.

Algo muy importante de mencionar es que la configuración de Oracle y herramientas administrativas requiere acceso a la configuración del cluster donde los datos son almacenados en discos compartidos, es decir para que se pueda instalar una base de datos Oracle en cluster es estrictamente necesario tener una configuración de cluster por hardware previamente.

La información de los nodos, la base de datos y la instancia es necesaria para las herramientas de administración. DataBase Configuration Assistant, la utilería SRVCTL y el Oracle Enterprise Manager deben estar disponibles a través del cluster. Asimismo, se requiere un sistema de discos compartidos para almacenar la información de la configuración y así poder utilizar estas herramientas.

En la figura 3.3 se muestra como todos los servidores tienen acceso directos a los discos compartidos que en donde se encuentra los archivos físicos de la base de datos.

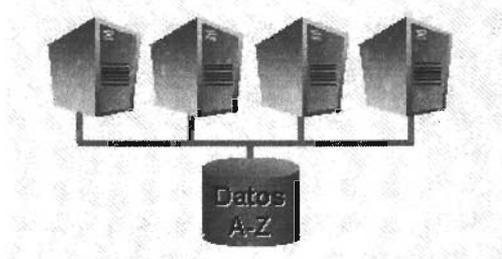


Figura 3.3 Discos Compartidos.

3.2 Arquitectura de Software para Real Application Cluster.

Ya que hemos definido qué hardware se necesita para la implementación del RAC ahora definiremos el software necesario para la instalación del cluster de base de datos. En primer lugar tenemos un software al que se le llama Clusterware.

3.2.1 Software de Cluster : Operating System Dependent (OSD).

RAC utiliza todos los componentes de una instancia normal de Oracle más el software de cluster que facilita la comunicación entre los nodos. El software de cluster del sistema operativo contiene el Operating System Dependent (OSD) el cual controla los servicios del sistema operativo y del clusterware requeridos para Real Application Clusters.

RAC utiliza el OSD para acceder el sistema operativo y para procesar los servicios de cluster como comunicar la información acerca de que la instancia esta iniciada o esta abajo. Los proveedores proporcionan el OSD para sistemas operativos Unix y Oracle proporciona el clusterware OSD para Linux Advanced Server 2.1, SuSe y para Windows NT y Windows 2000.

OSD tiene los siguientes componentes:

- El Cluster Manager.
- El Nodo Monitor.
- IPC (Interprocess Communication).

Cluster Manager (CM)

CM supervisa los mensajes que viajan a través del interconnect para coordinar operaciones entre nodos. Proporciona una vista global del cluster, por ejemplo de los nodos y de las instancias que son parte del cluster y controla los miembros del cluster. El CM usualmente lo proporciona el proveedor del sistema operativo, Oracle provee este software para Windows y Linux. RAC interactúa con el CM para obtener la información y requerimientos que necesita del cluster.

CM informa a los clientes y al servidor de Oracle cuando el status del cluster cambia. Esto es porque RAC maneja miembros de cluster para reconfigurar el cluster de la base de datos, por ejemplo cuando una instancia se agrega, se registra con el CM o cuando existe una instancia desconectada del cluster.

Nodo Monitor.

CM incluye un subconjunto de funcionalidad como el Nodo Monitor, el cual pregunta el status de cada recurso en el cluster incluyendo los nodos, el hardware de interconnect, el software y los discos compartidos.

El nodo monitor sirve al CM en los siguientes aspectos:

- Proporcionando comunicación entre nodos.
- Descubriendo y rastreando el estado de los miembros para proporcionar una vista actualizada de los nodos miembros que pertenecen al cluster.
- Detectando y modificando cambios en el estado de los nodos activos y comunicando esta información acerca de los cambios.

IPC (Interprocess Communication).

El software IPC es otro componente clave de OSD. El IPC controla los mensajes a través de los nodos. RAC también utiliza IPC para transferir data blocks entre las instancias, es decir permite la comunicación entre nodos que conformen el cluster.

En la figura 3.4 se muestra el software que es utilizado para una configuración de la base de datos en cluster.

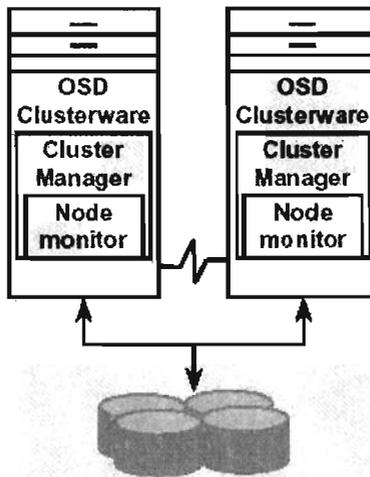


Figura 3.4 Componentes OSD

3.2.2 Componentes de la Base de Datos en RAC.

Una base de datos en RAC tiene los siguientes componentes que son casi idénticos a la de una base de datos normal:

- Una o más copias de archivos de control (control files).
- Un conjunto de archivos de redo log en línea (redo log file).
- Un conjunto de archivos de recuperación (archive log).
- Un conjunto de archivos de datos (data file).

- Un server parameter file es opcional (SPFILE).
- Una SGA para cada instancia la cual consiste de áreas configurables.
- Un conjunto de procesos de background como DBWR, PMON, SMON, LGWR.

3.2.3 Procesos de Instancia Específicos para Real Application Clusters.

En el capítulo 2, explicamos los conceptos de instancia, procesos, configuración del servidor de base de datos entre otros, para una base de datos que no está en cluster. Estos conceptos eran necesarios para poder entender que Real Application Cluster es una extensión de la base de datos Oracle. En la siguiente sección se explicarán los procesos de background extras que tiene una instancia de base de datos en RAC.

Global Services Deamon (GSD).

El proceso GSD corre en cada uno de los nodos con un proceso GSD, el cual coordina tareas de administración con el cluster manager para recibir requerimientos de clientes como DBCA (Data Base Configuration Assistant), EM (Enterprise Manager) y la utilería SRVCTL la cual se utiliza para ejecutar tareas como por ejemplo levantar y bajar una instancia.

GSD no es un proceso de background de una instancia de Oracle y por lo tanto no se inicializa con la instancia de Oracle.

Una base de datos en RAC tiene los mismos procesos que una instancia simple de Oracle, además de los procesos específicos que a continuación se listan:

1. Global Cache Service Processes (LMSn). Controla el flujo de mensajes a las instancias remotas, maneja los accesos globales a los data blocas y transmite imágenes de bloques entre el buffer cache de diferentes instancias. Este procesamiento es parte de la característica de Cache Fusion.

Maneja los requerimientos de los accesos de usuarios de datos a través del cluster, esto asegura que la misma imagen de los data blocas o datos puedan aparecer en el buffer cache para dos instancias diferentes si el contenido del bloque es válido para cada instancia.

Este proceso coordina el acceso a bloques mandando mensajes entre la instancia que intenta entrar a un bloque específico y la instancia que mantiene la imagen de ese mismo bloque.

Coordina la actualización de bloques, permitiendo que solo una instancia haga cambios en los bloques al mismo tiempo, y asegurándose que estos cambios han sido hechos para la versión más actualizada del bloque.

Mientras que una instancia mantiene la imagen del bloque en su buffer cache, el proceso LMSn es responsable de mantener el registro de esta imagen, incluyendo la actualización de la bandera de status si el bloque cambia. Puede haber hasta 10 procesos LMSn, dependiendo del número de mensajes entre los nodos del cluster.

2. Global Enqueue Service Monitor (LMON) es responsable de monitorear el cluster para manejar los global enqueues y los recursos a través del cluster, además de mejorar la ejecución de las operaciones de recuperación. Enqueues son estructuras de memoria compartidas que serializan los accesos a los recursos de la base de datos. Este proceso asegura que el proceso LMD y el almacenamiento para global enqueues funcionen correctamente.
3. Global Enqueue Service Deamon (LMD) administra en cada instancia los requerimientos remotos de recursos.

Es el proceso responsable de administrar requerimientos de los global enqueues y actualizar el status de los enqueues cuando estos son aceptados o rechazados, en una instancia. El proceso LMD de cada instancia soporta un registro de enqueue específico y recibe los requerimientos que necesita otra instancia, si este recurso está disponible el proceso LMD actualiza el estatus del enqueue y notifica al proceso LMD de la instancia remota. Si el recurso está siendo utilizado por otra instancia y no está disponible en ese momento, el proceso LMD se manda así mismo un requerimiento para que cuando esté disponible este recurso, el proceso LMD coordine la actividad, cambiando el status en su área de memoria, e informe a la otra instancia que el recurso está disponible.

El status actual de cada global enqueue es mantenido en un área de memoria compartida de cada una de las instancias activas. El status indica qué instancia, si es necesario, tiene que utilizar el recurso.

4. Proceso LCK: Asiste al proceso LMSn con los requerimientos de recursos entre las instancias y soporta los requerimientos de cache. Administra los requerimientos de recursos de las instancias y llamadas de operaciones entre las instancias. Estas llamadas son asociadas con la coordinación de accesos al diccionario y a los objetos de raw cache. Estos recursos de cache y los procesos de LCK en las instancias no involucran la transferencia de bloques de Cache Fusion entre las instancias.
5. Diagnosability Deamon (DIAG) captura diagnósticos de datos acerca de procesos fallidos de las instancias. Esta información puede ser analizada para resolver problemas con la base de datos y las instancias. La operación de este demonio es automatizada y actualiza un archivo de alerta para grabar la actividad que éste realiza.

El proceso DIAG escribe la información que contiene el diagnóstico en archivos en un directorio especificado. Este proceso se inicia automáticamente, y no debería ser

deshabilitado o borrado, y éste podría ser restaurado por otro proceso de background si es necesario.

En las figura 3.5 se muestra la correlación entre cada componente de una base de datos en RAC, cada instancia corresponde a un nodo del cluster que están comunicadas por medio de interconnect, cada instancia con sus respectivos procesos de background y redo log files, además de que todas las instancias accesan a los mismo archivos de la base de datos.

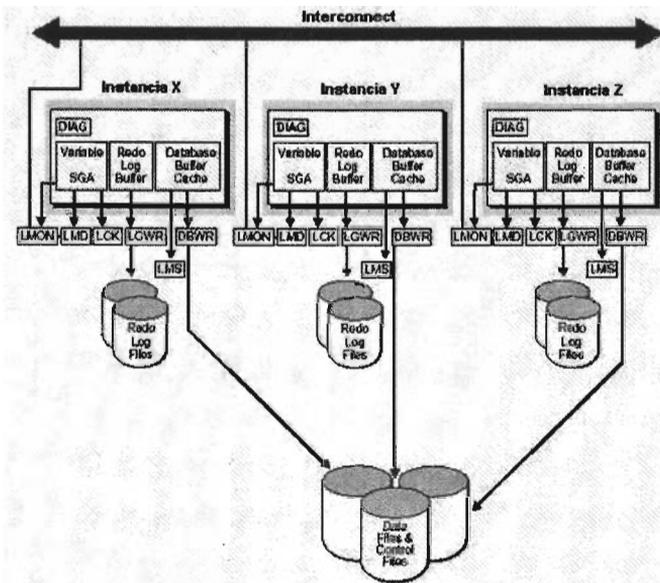


Figura 3.5 Cluster Base de Base Datos.

3.2.4 Los Servicios: Global Cache Service (Gcs) y Global Enqueue Service (Ges).

El GCS y GES son componentes integrados de RAC que coordinan simultáneamente el acceso a la base de datos compartida y los recursos compartidos de la base de datos. Estos servicios mantienen la consistencia y la integridad de los datos. El GCS y GES en cada instancia, así como el Cluster Manager, utiliza el IPC para comunicarse entre las instancias y dentro del cluster.

Cuando una instancia hace la petición de un recurso, como un data block, RAC maneja localmente la adquisición del bloque de esta instancia. Si este bloque es modificado por una o más instancias, entonces Oracle realiza la sincronización sobre el nivel global para permitir el acceso compartido a este bloque a través del cluster. Para que se realice esta sincronización se requiere de un mensaje entre los nodos así como las versiones

consistentes del bloque y la transmisión de las copias de los bloques entre las memorias cache de las instancias que pertenecen al cluster.

Global Cache Service (GCS): Localiza, prepara y transmite la imagen más actual del bloque cuando una o más instancias requieren un data block que esta siendo actualizado por otra instancia, para coordinar el procesamiento GCS, el GES transmite mensajes entre los nodos a través del interconnect.

El GCS maneja la ubicación, modos y roles de los data blocks. También maneja los privilegios de acceso de las instancias en relación con los recursos. Oracle utiliza el GCS para la coherencia del cache, es decir, cuando la versión actual del data block esta en el buffer cache de una instancia y otra instancia requiere ese bloque para modificarlo. Por el control de sincronización de GCS solo una instancia puede actualizar el bloque, la otra instancia debe esperar.

Global Enqueue Service (GES): Maneja todo lo que no tiene que ver con el Cache fusión, es decir, todas las operaciones de recursos entre las instancias y el estatus de todos los mecanismos de enqueue, el primer recurso que el GES controla es el Dictionary Cache y Library Cache. El GES también maneja la comunicación de los recursos entre las instancias.

Las características de GCS y GES son:

- **Transparencia de aplicación:** La coordinación de acceso a recursos de la instancia y del hardware es realizado por el GCS y GES es transparente para las aplicaciones. Las aplicaciones en RAC utilizan los mismos mecanismos de concurrencia en una instancia normal de base de datos Oracle.
- **Global resource directory con arquitectura distribuida:** El GSM y GES mantienen al global resource directory para grabar información acerca de recursos. El global resource directory reside en memoria, es distribuido a través del cluster, y esta disponible para todas las instancias activas. En esta arquitectura distribuida cada nodo participa en el manejo de la información del directorio. Este esquema distribuido proporciona fault tolerance y un mejor performance. GCS Y GES aseguran la integridad del Global Resource Directory aunque uno de los nodos falle, la base de datos compartida siempre está disponible, con que una de las instancias este activa después de que la recuperación este completa. El fault tolerance de resource directory también habilita a las instancias de RAC a poder inicializarlas y bajarlas en cualquier orden.
- **Recurso maestro y afinidad:** El GCS y GES mantienen información acerca de cada recurso en el cluster. El GCS y GES nominan una instancia para manejar toda la información acerca de un recurso en particular. Esta instancia es llamada Resource Master. El GCS evalúa los recursos periódicamente y cambia los recursos basado en el acceso de datos. Esto reduce el tráfico de red así como el tiempo de adquisición del recurso.

- **Interacción de GCS y GES con cluster manager:** El GCS y el GES operan independiente del cluster manager. De cualquier forma estos servicios confían en el cluster manager para corregir la información acerca del estatus de las instancias en el cluster. Si estos servicios no pueden obtener la información que necesitan de una instancia en particular, Oracle baja la instancia, para asegurar la integridad de la base de datos en RAC, esto sucede porque cada instancia debe de estar enterada de todos las instancia activas para coordinar el acceso a los discos compartidos.

3.3 Cache Fusion.

Cache fusion es un componente que da a RAC una escalabilidad lineal, es decir, que permite que el numero de componentes en el cluster aumente. Es un mecanismo de cache en RAC que proporciona las copias de los bloques directamente de la memoria cache de la instancia que le pertenece el dato a otra memoria cache de la instancia que lo esta requiriendo, haciendo más rápido el acceso a los datos.

Cache fusion puede crecer exponencialmente su poder de procesamiento en una base de datos en RAC, dependiendo de los nodos en integren el cluster.

Por default, un recurso es alojado para cada data block que reside en el cache de una instancia. Con el cache fusion y la eliminación de escritura de discos que ocurren cuando una instancia requiere un bloque para hacerle modificaciones, el funcionamiento máximo de procesamiento que se necesita para manejar datos compartidos entre instancias es considerablemente disminuido.

El cache fusion tiene diferentes tipos de concurrencia que se describen en seguida:

- **Lecturas concurrentes en nodos múltiples:** Ocurre cuando dos instancias necesitan leer el mismo data block. RAC resuelve esta situación sin la sincronización porque múltiples instancias pueden compartir los data blocks para lectura sin tener conflictos de coherencia de cache.
- **Lecturas concurrentes y escritura en nodos diferentes:** Ocurre cuando hay un requerimiento de lectura de una instancia, pero ese mismo bloque ha sido modificado por otra instancia y todavía no se ha escrito a disco, el resultado puede ser la versión más actual del bloque o por la versión de consistencia de lectura en cualquier caso, el proceso Global Cache Service (GCS) transfiere el bloque del cache de la instancia que modifiko el bloque a la instancia que lo esta solicitando a través del interconnect.
- **Escrituras concurrentes en nodos diferentes:** Ocurre cuando el mismo data block es modificado frecuentemente por diferentes instancias. En este caso la instancia que estaba modificando completa su trabajo sobre el data block después recibe el requerimiento para el bloque. El GCS cubre el recurso sobre el bloque para que sea globalmente manejado y el proceso LMSn transfiere la copia del block al cache de la instancia que esta requiriendo el dato.

Las principales características de este procesamiento son:

- El GCS graba cada versión del data block, y cada versión es referida como una imagen pasada.
- La transferencia de datos de cache a cache es realizada través del interconnect, lo cual elimina escritura y lectura de disco.

Los siguientes escenarios ilustran los puntos más importantes de Cache Fusion.

Escenario 1: Requerimiento de un bloque cambiado para una operación de modificación, en la figura 3.6 se ilustran los siguientes pasos:

1. La instancia 1 es la que quiere el bloque para modificarlo, lanza un requerimiento al GCS.
2. El GCS transmite el requerimiento a la instancia 2 que esta adueñada del bloque.
3. La instancia 2 recibe el mensaje y el proceso de LMS manda el bloque a la instancia.
4. Cuando la instancia 1 ha recibido el bloque, la instancia informa al GCS que ahora ella es dueña del bloque.

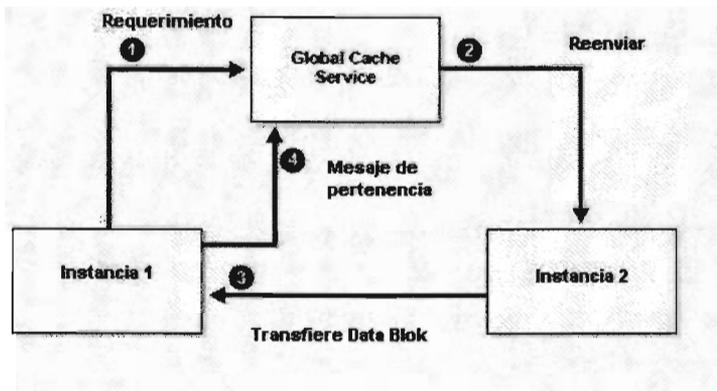


Figura 3.6

Escenario 2: Escribir bloques a disco.

Este escenario muestra como una instancia puede realizar un checkpoint en cualquier tiempo o reemplazar buffers en el cache. Esto es porque las múltiples versiones de los data block con cambios pueden existir en las caches de las instancias en el cluster, un protocolo de escritura es manejado por el GCS asegurando que solo en la versión más reciente de los datos es escrita al disco. También debe asegurar que todas la versiones previas son purgadas de las otras caches. Un requerimiento de un data block puede originar que

cualquier instancia tenga la actual o imagen pasada del bloque. En la figura 3.7 se ilustran los siguientes pasos.

1. La instancia 2 manda el requerimiento de escritura al GCS.
2. El GCS envía el requerimiento a la instancia 1, que es la que actualmente mantiene el bloque en su poder.
3. La instancia 1 recibe le requerimiento de escritura y escribe el bloque a disco.
4. La instancia 1 graba la terminación de la operación de escritura con el GCS.
5. Después que el GCS recibe la notificación, ordena que todas las imágenes pasadas sean descartadas y que ya no son necesarias para recuperación. El buffer esta libre en este momento.

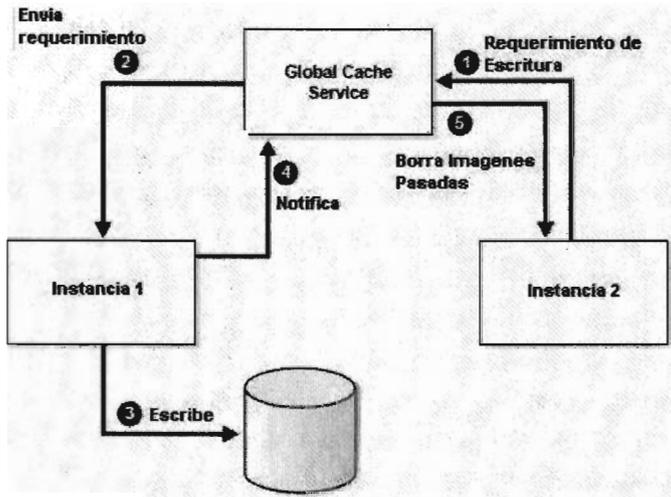


Figura 3.7

Con esto concluimos la arquitectura de RAC, así como su funcionamiento a detalle, basados en los conocimientos de los capítulos anteriores, el siguiente capítulo me enfocare a las ventajas de RAC y a la comparación con otras tecnologías de cluster de base de datos.

CAPÍTULO 4: BENEFICIOS DE ORACLE REAL APPLICATION CLUSTER.

En este capítulo explicaremos las ventajas que proporciona una configuración de RAC a un sistema transaccional que se necesita que este disponible todo el tiempo, además de comparar el producto de Oracle con otros proveedores que ofrecen una solución de cluster de base de datos como Microsoft e IBM.

4.1 Ventajas Adquiridas en una Configuración de RAC.

Con una configuración de RAC, se tiene un sistema de alta disponibilidad en el servicio de base de datos, porque se explota la redundancia proporcionada por el cluster de base de datos, para dar una disponibilidad de $n - 1$ nodos sobre n nodos, en otras palabras, todos los usuarios tiene acceso a todos los datos aun que haya un solo nodo disponible en el cluster.

4.1.1 Redundancia.

La redundancia en una infraestructura de computo, es contar con 2 o mas componentes de una pieza de dicha infraestructura.

Los componentes de RAC en los cuales se tiene alta disponibilidad son los siguientes:

- **Nodos del cluster:** Todos los nodos accesan a la base de datos, cuando har dos o mas nodos ya se cuenta con redundancia de servidores. Si un nodo llegara a fallar no afecta la disponibilidad de los nodos sobrevivientes ni las transacciones que estén corriendo en ese momento en los demás nodos que pertenecen al cluster. Aunque haya un solo nodo vivo en el cluster los clientes pueden procesar todas las transacciones, claro que están sujetos a la capacidad de procesamiento del nodo sobreviviente.
- **Cluster interconnect:** La redundancia del interconnect es regularmente pasada por alto, porque el tiempo promedio de falla en un interconnect es de años, además la redundancia del interconnect no debe de ser de alta prioridad, también depende del sistema y del nivel de sofisticación, tener una redundancia del interconnect en el cluster puede ser muy costoso, ya que pueden ser piezas con alto valor económico. De cualquier forma un interconnect redundante en el cluster es un aspecto importante de un sistema de alta disponibilidad en una configuración de RAC.
- **Software de la base de datos:** en RAC, los ejecutables de Oracle son instalados en CFS (Cluster File System) o en los discos locales de cada nodo. Y en cada nodo se tiene una instancia corriendo, todas las instancias tendrán acceso igual a

todos los datos y pueden realizar cualquier transacción. De esta forma RAC asegura que haya una absoluta redundancia en el software de la base de datos.

4.1.2 Failover.

Describiremos las características del failover y cómo el RAC lo implementa para proporcionar un sistema de alta disponibilidad.

El failover requiere que los sistemas de alta disponibilidad tengan una instancia de monitoreo o mecanismos de heartbeat, aparte de tener esta funcionalidad para una operación normal, el sistema debe de poder sincronizar los recursos durante el failover.

El proceso de sincronización requiere que el sistema que este fuera de servicio por un determinado tiempo se identifique como el sistema que tuvo la falla. Para que se defina el control de los recursos que tomaran el sistema remoto.

Por esta razón en RAC los sistemas graban la información de los recursos de los nodos remotos así como del local. Esta información es requerida por el failover cuando se presente una falla del sistema. La duración de un failover incluye el tiempo que el sistema requiere para tomar el control de los recursos del sistema, y el tiempo para recuperarse de las fallas.

Failover En El Cliente.

Es importante esconder las fallas del sistema de las conexiones de clientes de base de datos, estas conexiones pueden incluir usuarios de aplicación en un ambiente cliente/servidor o clientes de base de datos en una capa intermedia en ambientes de aplicaciones multicapas. Cuando se configura un mecanismo de failover transparente la sesión del cliente que esta accedendo al servidor que ha fallado en ese momento se direcciona a un nodo disponible en el cluster, esta capacidad de redireccionamiento en las bases de datos Oracle se conoce como TAF (Transparent Application Failover).

TAF (Transparent Application Failover).

TAF habilita al usuario para que se pueda reconectar automáticamente a la base de datos si la conexión falla. Las transacciones actuales se deshacen en ese momento, dependiendo de la operación que se esta haciendo, por ejemplo si la transacción es solo de consulta esta continuara sin que haya interrupción de servicio, pero si se actualizando los datos, la transacción se pierde y no finaliza; pero el servicio no se pierde, el usuario tendrá que hacer una reconexión, e inmediatamente seguirá trabajando.

Las conexiones que están asociadas con el TAF son las siguientes:

- Conexiones a la base de datos en un ambiente de cliente/servidor.
- Sesiones de usuarios de base de datos.
- Transacciones activas.

TAF automáticamente restaura estos elementos, por ejemplo, durante una operación normal de una base de datos en cliente/servidor, el cliente mantiene conexión a la base de datos, entonces el cliente y el servidor se pueden comunicar, si el servidor falla, la conexión se pierde, la próxima vez que el cliente trate de reconectarse, habrá un error y el usuario tendrá que validarse a la base de datos otra vez.

Con TAF el software de Oracle obtiene una nueva conexión a la base de datos, esto permite a los usuarios seguir trabajando como si la conexión original no hubiera fallado, ni siquiera se presenta un mensaje de error de que la conexión se ha perdido, mientras que halla una instancia disponible para servir a la aplicación.

La figura 4.1 muestra el funcionamiento del TAF, es decir en el momento que ocurre una falla en algún nodo del cluster, los clientes que estaban conectados a este nodo son reconectados al nodo sobreviviente, sin perder el servicio definitivamente, depende de la transacción que se este ejecutando.



Figura 4.1 Transparent Application Fail-Over.

La disponibilidad del failover para las sesiones de clientes es un beneficio importante del TAF para proporcionar alta disponibilidad, sin embargo, no es el único pues existen otros escenarios en donde el TAF es muy útil, los cuales se mencionan a continuación:

- **Shutdown transaccionales:** algunas veces es necesario sacar algunos nodos fuera de servicio para mantenimiento o reparación, por ejemplo, si se desea aplicar un parche ya sea de sistema operativo o del software de Oracle sin interrumpir el servicio a los usuarios, el shutdown transaccional facilita dar de baja los nodos seleccionados en vez de bajar toda la base de datos. Después de realizar un shutdown transaccional, Oracle redirecciona las nuevas transacciones al nuevo nodo e inmediatamente se da de baja el nodo cuando las transacciones existentes sean completadas.

- **Failover durante el procesamiento del cliente de base de datos:** cuando se realiza un failover por alguna falla y hay una transacción corriendo ya sea un query de lectura o de manipulación de datos, el funcionamiento del failover es diferente para cada una de ellos. Pero lo importante durante la operación de un failover es que la falla es cubierta por la conexión existente tan pronto como sea posible. A continuación explicaremos el funcionamiento del failover en los dos tipos de query's.
 - Query de lectura: En el momento que ocurre una falla, como por ejemplo que se caiga una instancia, y hay un query de lectura ejecutándose de un cliente x, para este cliente debe de ser transparente la reconexión a algún nodo que este disponible, es decir no se debe dar cuenta de que el sistema tuvo una falla, es en este momento cuando el TAF entra en acción, y a través de la configuración de los archivos de red de Oracle (tnsnames.ora, listener.ora) en los cuales se configuran para el failover, se realiza el cambio de conexión a un nodo activo en menos de 5 segundos, la experiencia para el usuario final es solo una pausa en el query, es decir, se congela la pantalla por unos momentos, y el query continua hasta finalizar.
 - Query de manipulación de datos: cuando un cliente esta ejecutando una sentencia de DML (Database Manipulation Language) como por ejemplo INSERT; UPDATE y DELETE, Oracle desconecta al cliente y da un roll back a este tipo de transacciones, el punto es que en el momento que el cliente se reconecte habrá servicio disponible de inmediato; no se perderá el servicio definitivamente, bastara que se vuelva a conectar para seguir trabajando.

Failover del Servidor.

RAC proporciona un rápido failover de servidor. Esto se puede realizar porque la arquitectura de RAC es tener nodo activo-activo, o sea, múltiples instancias Oracle están activas en múltiples nodos y estas instancias sincronizan acceso a la misma base de datos.

Todos los nodos tienen acceso a todos los discos y cuando un nodo falle, todos los otros nodos en el cluster mantienen acceso a los discos. Dependiendo del tamaño de la base de datos, la duración del failover puede variar.

4.1.3 Balanceo de Cargas.

Una base de datos esta disponible cuando una transacción termina de manera oportuna. Cuando la carga excede la capacidad del nodo, las respuesta del nodo hacia el cliente se disminuye considerablemente y la disponibilidad y funcionalidad de la base de datos se compromete, en esta situación es cuando se da la distribución de clientes entre los nodos con el objetivo de que se tenga menos carga sobre un

solo nodo, esta operación es útil para mantener un tiempo de respuesta aceptable y disposición de la aplicación, ha esto se le llama balanceo de cargas, este balanceo puede ser aleatorio, por round robin o cuando se detecte que existe un nodo sobrecargado en usuarios. En RAC el TNS (Transport Network Services) es el que proporciona un balanceo de cargas automático a través de los nodos que forman parte del cluster.

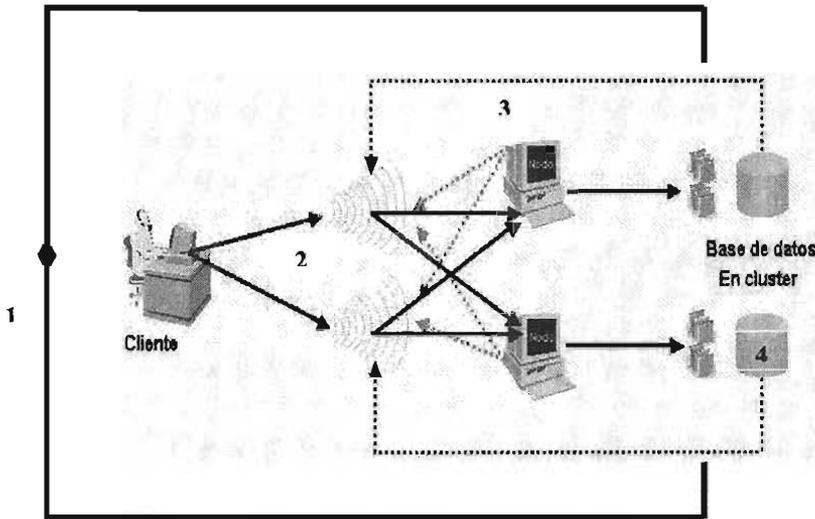


Figura 4.2 Balanceo de Cargas.

En la figura anterior se muestra como es el procedimiento de balanceo de cargas:

1. El cliente hace una petición de conexión.
2. Los listener recibe las peticiones de conexión y están configurados para balancear las conexiones.
3. Los nodos del cluster reciben esta petición y realizan la operación para devolver el resultado de las operaciones solicitadas.
4. Los manejadores de base de datos mandan sus respuestas del requerimiento solicitado al cliente.

Tanto el TAF como el balanceo de cargas se implementan a través de los archivos de configuración propios de Oracle listener.ora y tnsnames.ora que pertenecen al TNS.

4.1.4 Escalabilidad.

Se refiere a la posibilidad de agregar nodos al cluster para incrementar las capacidades de procesamiento sin tener que redistribuir los datos o modificar las aplicaciones.

En este sentido, RAC automáticamente aumenta el poder de procesamiento de los nodos adicionales que han sido sumados al cluster para proporcionar escalabilidad sin ser necesario una baja del servicio, con Cache Fusion no es necesario reparticionar datos o modificar la aplicación usuaria para tomar ventajas del poder adicional de procesamiento del CPU de los nodos sumados.

La figura ejemplifica como el cache fusion es un elemento importante en la escalabilidad de memoria y procesamiento, porque al agregar un nodo más al cluster la arquitectura de este componente inmediatamente utiliza los recurso del CPU y de memoria del nuevo nodo, RAC automáticamente reorganiza los recursos de la base de datos de una manera óptima utilizando los recursos del cluster con un mínimo de I/O en disco y una disminución de comunicación entre los nodos, esto permite que se disminuya el tiempo de respuesta de la aplicación, ya que la memoria de todos los servidores agregados al cluster se suma para procesar mas rápido las peticiones de los usuarios.

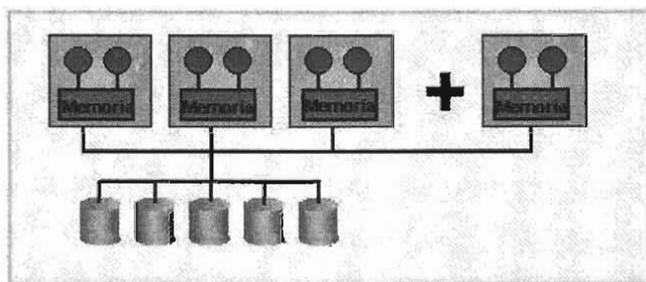


Figura 4.3 Escalabilidad.

La figura 4.3 muestra la escalabilidad de procesamiento y memoria al ir agregando nodos al cluster.

Los sistemas transaccionales son caracterizados por hacer actualizaciones a los datos de una manera constante, este tipos de sistemas se desempeñan muy bien en RAC, porque proporciona escalabilidad tanto en el procesamiento, como en memoria, es decir, en cualquier momento en que el sistema necesite crecer, ya sea en performance o en procesamiento, se pueden agregar nodos para darle este plus al sistema.

4.2 Comparación de RAC con otras Bases de Datos.

En este apartado compararemos RAC con otras soluciones de cluster de base de datos proporcionadas por otros proveedores: Microsoft e IBM, cada solución con sus propias características de comparación en base a RAC, ya que no se pueden tomar los mismos aspectos por la diferencia de arquitectura. A continuación se hará una comparación de IBM y Microsoft con Oracle para exponer las ventajas y desventajas de las diferentes soluciones ofrecidas por cada proveedor.

4.2.1 RAC (Shared Database) vs. Microsoft SQL Server (Federated Database).

Microsoft SQL Server se basa en la tecnología de base de datos federada que a continuación se explica con más detalle.

Una base de datos federada es una unificación lógica de distintas bases de datos corriendo en servidores independientes, no comparten recursos y se conectan a través de una LAN (Local Area Network). Los datos son horizontalmente particionados a través de los servidores participantes.

Tanto para el DBA como para el desarrollador de la aplicación, hay una clara distinción entre los datos locales y los datos remotos, las aplicaciones ven una vista simple de los datos a través de las vistas UNION ALL y SQL Distribuidos. Microsoft llama a esta tecnología Vistas Particionadas Distribuidas (DPV: Distributed Partitioned Views).

Las DPV son construidas de manera diferente en cada nodo, se debe considerar qué particiones son locales y cuáles son remotas.

En la figura se 4.4 muestra la arquitectura de una base de datos federada.

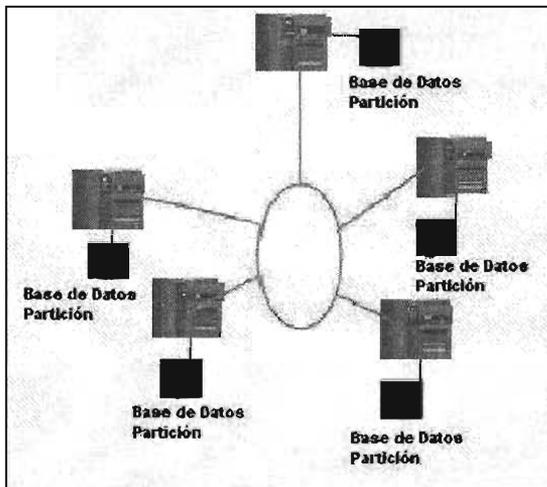


Figura 4.4 Base de Datos Federada.

Entonces una base de datos federada es una base distribuida lógicamente basada en la tecnología de DPV, esa decir, vistas y SQL distribuidas en todos los servidores para poder acceder a los datos de todos los miembros de la base de datos federada.

Puntos de Comparación RAC y Microsoft SQL Server.

Ahora bien ya que se tiene los conceptos de una base de datos federada en la cual se basa Microsoft con su producto SQL Server 2000 y una base de datos en cluster con discos compartidos como Oracle Real Application Clusters de Oracle, se hará una comparación de ambos productos en los siguientes aspectos:

1. Desarrollo de Aplicación.
2. Escalabilidad.
3. Disponibilidad.
4. Administración.
5. Costo.
6. Implementación de la solución.

I. Desarrollo de Aplicación.

La aplicación es un elemento muy importante dentro de un sistema computacional, pues es la interfaz que interactúa con el usuario final, dependiendo las necesidades de la empresa la aplicación es elaborada a la medida o proporcionada por proveedores.

Una aplicación tiene que acceder a una base de datos, en este caso mencionaremos qué diferencias hay entre un ambiente multinodo de una base federada y discos compartidos, respecto al desarrollo de la aplicación:

a. Oracle Real Application Cluster.

Oracle Real Application Clusters (RAC) no necesita una restricción para el desarrollador de la aplicación. Una aplicación OLTP (Online Transaction Process) como SAP o Oracle e-Business Suite, utilizando la base de datos Oracle en una plataforma SMP (Symmetric MultiProcessing) corre sin ninguna modificación en un cluster de discos compartidos y RAC. La imagen de una sola base de datos que proporciona RAC, sin particiones de datos, hace que no tenga que haber ninguna modificación de código en la aplicación.

b. Microsoft SQL Server 2000.

Microsoft SQL Server parte la imagen de una sola base de datos en múltiples instancias. Los datos deben ser distribuidos a través de las bases particionadas o replicados. Esto implica hacer cambios en el código de las

aplicaciones por ejemplo dividir los datos a través de la bases de datos requiere crear Distributed Partition Views que son distintos en ambos nodos y tener tablas replicadas necesita mantener en sincronización las bases de datos utilizando triggers customizados.

Hay aplicaciones hechas por terceros como SAP, PeopleSoft y Oracle Applications, que tienen miles de tablas e índices únicos que no pueden ser distribuidos en múltiples nodos, pues traen un código ya elaborado y no se permite su modificación es aquí cuando una base de datos federada ya no es aplicable con estas aplicaciones.

Cuando hay aplicaciones elaboradas por el equipo de desarrollo de la empresa o por despachos dedicados a elaborar software es más fácil que las aplicaciones se personalicen a esta tecnología, pero aun así, es un esfuerzo extra modificarlas.

2. Escalabilidad.

La escalabilidad es una de las razones para que un cliente se mueva a una plataforma SMP (Symmetric MultiProcessing) con un ambiente multinodo, como la arquitectura federada o de discos compartidos, ya que proporcionan una mejora considerable en el performance de las aplicaciones.

La escalabilidad en la base de datos es un recurso que se utiliza cuando el negocio crece, y se requiere una mejor respuesta por parte del sistema de información conforme a las necesidades del cliente.

a. Oracle Real Application Cluster.

Un cluster de base de datos basado en cluster de discos compartidos siempre ha escalado bien para tener un mejor funcionamiento de performance, pues se cuenta con la implementación de la tecnología de shared-cache o cache compartida en la que esta basada el cache fusión de RAC. El agregar más nodos al cluster significa agregar más recursos para que cuando haya una carga de trabajo más pesadas se reparta en todos los nodos del cluster.

b. Microsoft SQL Server 2000.

Hay algunas barreras inherentes para que una base de datos federada sea escalable, cada que se agrega un nodo a este tipo de arquitectura significa que se tiene que agregar mas DPV con una clave de particionamiento, es decir se necesita agregar más código en todos los nodos existentes para que lo reconozcan como miembro de la federación. Además mientras más nodos se agreguen menor performance se tendrá ya que de una sola consulta tendrá que acceder a todos los nodos para ver si se encuentra en uno de ellos.

3. Disponibilidad.

La disponibilidad es un factor crítico en un sistema de información, si no hay información no se puede continuar con el funcionamiento del negocio, hay clientes que no pueden dejar de contar con su información porque esto significaría pérdidas millonarias para la empresa, ahora bien se hará un comparativo de lo que ofrece Microsoft y Oracle para poder tener la disponibilidad de nuestra información siempre:

a. Oracle Real Application Cluster.

Cuando un nodo en cluster de discos compartidos falla, todos los datos permanecen accesibles para los otros nodos. Y las transacciones que se estaban ejecutando se les da un roll-back.

En Oracle Real Application Clusters, después de que se detecta que un nodo ha fallado, el cluster es automáticamente reconfigurado y el mismo proceso de recuperación (roll-forward/roll-back) de los ambientes SMP son aplicados.

Para aplicaciones que están configuradas con el TAF (Transparent Application Failover) proporcionada por OCI(Oracle Call Interface), los usuarios que estaban conectados al nodo que fallo, serán reconectados automáticamente y podrán seguir trabajando sin interrupción.

b. Microsoft SQL Server 2000.

La información en las bases de datos federadas es dividida entre todas las instancias de bases de datos que pertenecen a la federación, es decir, por cada nodo que pertenece al cluster existe una instancia en donde se almacenara una porción de datos.

La única manera de acceder a los datos que pertenecen a otro nodo es hacer un requerimiento del nodo del cual se quiere la información y esperar la respuesta.

Entonces, cuando un nodo falla, los datos que pertenecen a ese nodo quedan inaccesibles. Y cualquier transacción distribuida controlada por el nodo bloqueara los datos para los otros nodos, además de que la recuperación del nodo que fallo requiere un esfuerzo adicional para resolver las transacciones que se estaban ejecutando, ya que la recuperación necesita un serie de pasos manuales que requiere que el sistema sea puesto fuera de línea, lo que recomienda Microsoft es tener un nodo de failover para cada partición, lo que sería duplicar el hardware y el costo de la configuración.

4. Administración.

La administración en un RDBMS es proporcional al número de entidades definidas y por mantener. Cuando se mueve de una instancia a una arquitectura multi-instancia como federada o de discos compartidos, se incurre en la administración de parámetros específicos de configuración, de inicialización y de manejo. Se comparan las tareas de administración y mantenimiento que se tienen que realizar en los dos ambientes en cuestión:

a. Oracle Real Application Cluster.

Moverse de una instancia de Oracle a un ambiente multi-nodo como RAC requiere mantener:

- Configuración específica de cada instancia y parámetros de tuning. Muchos de estos parámetros son parecidos en los nodos.
- Recuperar los redo log por instancia.

Deben mantenerse especificaciones de instancia similares en cada nodo de la base de datos federada. Sin embargo, RAC no requiere ninguna otra tarea de mantenimiento que crezca con el número de nodos en el cluster.

Las tareas de administración que se requieren cuando se agrega un nodo en RAC son las siguientes:

- Agregar hardware.
- Configurar la nueva instancia.

b. Microsoft SQL Server 2000.

En la configuración federada cada base de datos requeriría de un respaldo y recuperación por separado, además de hacer tareas de mantenimiento, seguridad, administración de usuarios, administración de espacio, además de que la miles de tablas e índices que son típicos en un complejo mundo real de las aplicaciones OLTP se tendrán que dividir a través de los nodos participantes, y el manejo de cada objeto requeriría un trabajo extra por nodo muy complicado.

Las DVP lógicamente unifican los datos distribuidos a través de cada nodo de base de datos SQL Server, agregar un nodo a las bases de datos federadas cambian la distribución de los datos y por lo tanto la implementación de los DVP.

Estas son las tareas que un DBA debe de realizar cuando un nodo es agregado a una base de datos federada:

- Agregar hardware.
- Configurar una nueva instancia.
- Crear una nueva base de datos.

- Desconectar a todos los usuarios.
- Descargar los datos de las tablas existentes.
- Redefinir el particionamiento de las tablas e índices.
- Redefinir los trigger sobre tablas particionadas o replicadas.
- Redefinir los DVP.
- Cargar los datos a las diferentes particiones.
- Reconectar a todos los usuarios.

Una de las ventajas más notorias de Oracle sobre Microsoft es la de administración y mantenimiento de los nodos en el cluster, pues con Microsoft se dificulta demasiado cada vez que se agrega un nodo al cluster.

5. Costo e Implementación.

El tamaño de la implementación depende del tamaño de la empresa, implementar una solución de alta disponibilidad en la base de datos, requiere tener el hardware y el software necesario, lo que significa invertir en la solución.

En el caso de Oracle es necesario contar con una tecnología de cluster proporcionada por un tercero en algunas plataformas como HP, Sun, IBM que consta de hardware y software de cluster, las cuales suelen ser costosas en el precio y en la implementación, y por ejemplo en el caso de Linux y Windows Oracle proporciona el software de cluster y no es necesario contar con un hardware adicional.

Además de contar con las licencias del producto, tanto de Oracle (base de datos) como para Oracle 9i Real Application Clusters (solución de alta disponibilidad de Oracle).

Es por esto que para las empresas pequeñas es difícil adquirir una solución de alta disponibilidad en base datos, así como en sistema operativo, de hecho esta solución de Oracle esta dirigida para empresas medianas y grandes, para las que sea indispensable con una solución de alta disponibilidad.

Para implementar y poner a punto la solución de Oracle 9i Real Application se necesita de especialistas que conozcan el producto lo que ocasionaría un costo más para el cliente.

Cualquiera de las soluciones expuestas en este apartado será la correcta dependiendo de las necesidades y capacidad del cliente.

4.2.2 RAC (Shared Database) vs. IBM DB2 UDB EEE v8.1 (Shared Nothing Database).

A continuación hablare del producto de base de datos desarrollado por IBM: DB2, tocare un poco de su arquitectura y su funcionamiento, y después comparar la solución de escalabilidad de DB2 con la de Oracle.

Es conveniente pensar que una partición en DB2 es equivalente a una instancia de Oracle excepto porque la partición accesa a un subconjunto de datos directamente.

Los requerimientos de software y hardware son similares. La abstracción para la aplicación es la misma con ambas soluciones, da la ilusión de que es una base de datos y no se necesita modificar el código. Las diferencias se encuentran en la facilidad de desarrollar, el performance y el manejo o administración.

Desde una vista conceptual, los requerimientos para RAC y para DB2 son similares, especialmente cuando la alta disponibilidad es un requerimiento básico. Un cluster, CPU, cluster interconnect y memoria son similares para ambas soluciones.

Una de las más grandes diferencias entre RAC y DB2, es que DB2 utiliza shared nothing como medio de almacenamiento, mientras de RAC utiliza shared-disk, la tecnología de shared disk fue explicada a detalle en los capítulos anteriores, ahora explicaremos la tecnología de shared-nothing para poder hacer una comparación entre las dos tecnologías.

En arquitecturas basadas en share nothing, los archivos de la base de datos están particionados entre las instancias que corren en los nodos de un sistema multi-computadoras. Cada instancia o nodo es dueño de un subconjunto de datos y todos los accesos a estos datos son realizados exclusivamente mediante la instancia que es dueña. En otras palabras un sistema shared nothing utiliza un acceso particionado o restringido para dividir el trabajo entre los múltiples nodos. Esto funciona bien en ambientes donde los datos cambian muy poco, las razones más frecuentes de cambios considerables en la base de datos es cuando hay reorganizaciones de bases de datos o cuando hay fallas en los nodos.

La ejecución en paralelo en un sistema de shared nothing esta directamente basado en un esquema de particionamiento de datos. Cuando los datos son exactamente particionados, el sistema es escalable linealmente. En los sistemas de base de datos basados en la tecnología de shared nothing, puede haber un subsistema dual de discos, entonces cada conjunto de discos esta físicamente conectado a dos nodos en el cluster con la noción de “dueño primario y secundario de discos”. Esta configuración protege al sistema de que un nodo falle, entonces, requiere que dos nodos fallen para que el sistema sea inaccesible, como esto es muy difícil que suceda, la falla de uno de los nodos puede causar degradación de performance.

Un nivel superficial, en un sistema de shared nothing es similar a una base de datos distribuida. Una transacción que se ejecuta en un nodo cualquiera debe de mandar mensajes a los otros nodos que es dueña de los datos que esta accesando.

Se debe coordinar el trabajo hecho en otros nodos para realizar las actividades requeridas de lectura/escritura.

De cualquier modo, una base de datos shared nothig es diferente de una base de datos distribuida porque opera con una sola base de datos física utilizando un diccionario de datos.

La figura 4.5 muestra como se encontraría distribuida una base de datos en una arquitectura shared nothing.

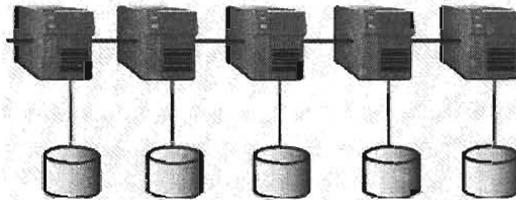


Figura 4.5 Base de datos en cluster utilizando la tecnología de shared nothig.

DB2 EEE Shared-Nothing.

Aunque DB2 EEE (Universal Database Enterprise Extended Edition) es considerada una base de datos shared nothing, la base debe ser creada en una tecnología shared-disk para proporcionar alta disponibilidad de los datos. Shared nothing solo se refiere a la pertenencia de datos mientras que la base esta corriendo, no a la conexión física.

En DB2 EEE es posible que los discos sean conectados solo a un subconjunto de maquinas como servidores propietarios secundarios de los datos en la partición, para tener un respaldo en cada partición de datos.

De cualquier forma, si solo se usa un subconjunto de datos entonces algunos nodos van a tener más cargas de trabajo que otros, además de reducir el procesamiento central del sistema así como el performance. RAC, en contraste, requiere de una conectividad total de los discos a todas las maquinas por lo que no sufre de esta situación.

RAC e IBM Parallel Sysplex Shared-Data.

IBM's Parallel Sysplex es la solución de disponibilidad de base de datos que ofrece IBM al mercado, la cual compararemos con Oracle.

RAC e IBM's Parallel Sysplex utilizan la tecnología de "shared data" el enfoque donde cada servidor tiene acceso a todos los datos, y cualquier transacción puede correr en cualquier servidor. El balanceo de cargas asegura una distribución de trabajo.

Los usuarios pueden continuar utilizando la aplicación aunque un servidor este fuera de línea y la combinación del poder de todos los servidores es útil para que una aplicación crezca linealmente.

Para poder utilizar de la tecnología de cluster de IBM Parallel Sysplex, se necesita de un hardware propietario y elementos de software de IBM, lo que con Oracle no sucede, porque se tiene la opción de utilizar múltiples sistemas operativos, es decir el software de Oracle pueden ejecutarse sobre Unix (Linux, AIX, Solaris, HP), Windows y arquitectura de hardware con múltiples proveedores.

Puntos de Comparación RAC e IBM.

Requerimientos de Software.

RAC y DB2 EEE dependen de un cluster manager para proporcionar al cluster servicios de miembros del cluster. En algunas plataformas como en Linux y Windows Oracle proporciona el software de cluster manager.

Por ejemplo en Windows 2000, DB2 EEE está limitada a 4 u 8 nodos en un cluster, dependiendo de la edición de Windows 2000, además no puede usar raw devices con Microsoft Cluster Service (MSCS) para los archivos de la base de datos.

Es posible, de cualquier forma que una base de datos Stand Alone corra en más de 8 máquinas sobre Windows 2000. La limitación es que una partición no puede estar fallando a través de las máquinas en los grupos diferentes.

RAC, por otro lado, puede correr hasta 64 bits hoy en día, esto permite que se utilicen nodos menos caros y más pequeños para realizar el mismo trabajo.

Y con el CFS y el cluster alias, se puede administrar el cluster entero como una máquina SMP.

Desarrollo.

Hemos visto que la configuración e instalación del software y hardware es muy similar. Aquí nos enfocaremos en los pasos adicionales requeridos para desarrollar una base de datos y una aplicación en un cluster vs. Desarrollando en una simple máquina.

RAC y DB2 EE permite a los usuarios conectarse a cualquier máquina en el cluster. En algunos casos, se puede considerar direccionar la conexión del cliente a una determinada máquina para improvisar localidad.

De hecho, no existe una diferencia significativa en la arquitectura respecto a las conexiones. Desarrollar una base de datos en RAC de una instancia exclusiva solo requiere que el DBA cree tanto redo y undo tablespace como nodos en el cluster, no hay modificaciones a los archivos de base de datos.

En contraste con DB2 EE debe de migrar forzosamente a EEE, los datos deben ser físicamente migrados y reorganizados, porque DB2 EEE necesita particionar los datos entre

los discos que son dueños cada nodo o partición en el orden en el que serán usados por los recursos extras de hardware. Particionar los datos en esta manera es una tarea muy compleja.

El Particionamiento de Datos.

En un ambiente de shared nothing, el particionamiento de datos es la forma de dividir los datos a través de varios servidores de acuerdo a las necesidades del cliente.

En resumen, las bases de datos en shared nothing por lo regular requieren que se particionen manualmente se los datos para un performance optimo. Cuando la base de datos esta en un sistema de shared disk no requiriere de un esquema de particionamiento de datos para escalar.

Con un cluster basado en el sistema de shared nothing, escoger un esquema apropiado de particionamiento es muy dificil ya que se debe conservar el balanceo de cargas a través de todos los nodos y evitar el costo del volver a particionar los datos. El esfuerzo de particionar los datos es una labor intensa y podría conducir al error humano (lo que ocasionaría un numero de razones para que el sistema este abajo). Esto es necesario de cualquier forma cuando un sistema de cluster escala integrando más nodos. En el mundo real aplicar esta técnica a una base de datos que contiene datos de escritura/lectura no es trivial.

Escalabilidad y Performance.

Hay muchos puntos que se deben tomar en cuenta para la escalabilidad y performance lo cual depende del tipo de aplicación (OLTP Online Transaction Process, DSS Decisión Support System) y de la carga de trabajo que estas corran. Veremos un poco de las características de cada una de este tipo de aplicaciones y como las dos arquitectura influyen en el performance cuando se hace una carga de datos.

Performance en Aplicaciones OLTP.

En los ambientes OLTP, una transacción típica es muy corta, porque actualiza, borra inserta datos, comparada con las transacciones en ambiente DSS (decision support system) las cuales son querys que solo consulta la información, el problema es que estos sistemas tienen cantidades grandes de información. El performance es medido por el rendimiento del procesamiento (numero de transacciones procesadas en una unidad de tiempo). Como ya lo hemos mencionado un esquema de particionamiento le agrega complejidad al desarrollo de la aplicación.

Performance en Aplicaciones DSS (Decisión Support System).

Usualmente a carga de datos en un sistema que soporta decisiones consiste de complejas y grandes consultas. Entonces el enfoque del diseño de la base de datos es asegurar que cada

consulta sea ejecutada en paralelo a través de todos los nodos de la base de datos en el cluster para reducir que el tiempo de ejecución de una consulta se colapse.

La arquitectura de consultas paralelas de RAC es única en su habilidad de determinar el grado de paralelismo para cada consulta. De lo contrario con DB2 EEE, el grado de paralelismo es determinado por el esquema de particionamiento de tablas underlying. El paralelismo de una consulta en Oracle es determinado inteligentemente basado en el número de CPU's, el número de archivos accesados y otras variables. RAC implementa un algoritmo adaptado para determinar el grado necesario de paralelismo. Este algoritmo avanzado considera la carga actual sobre un datawarehouse cuando escoges el grado de paralelismo.

Esta característica mejora el rendimiento de procesamiento de una datawarehouse ejecutando consultas en paralelo.

Disponibilidad.

Tanto RAC y DB2 EEE soporta todos los métodos convencionales para mantener alta disponibilidad, respaldo y recuperación.

De cualquier manera, cuando se conocen los detalles para cada una de estas características en oracle database se entienden las ventajas que oracle proporciona para minimizar el tiempo que este abajo el sistema.

Enseguida veremos como las diferentes arquitecturas de cluster de base de datos pueden afectar a la disponibilidad y costo de un sistema.

Recuperación De Ambientes.

Cuando sucede algo que provoque que un nodo falle, RAC y DB2 pueden recuperar la base de datos transparentemente. Aunque podemos esperar que los tiempos de recuperación sea más rápido con RAC por las siguientes razones:

- DB2 EEE confía en el cluster manager para reiniciar la partición en un nodo de que este vivo, esto requiere que un proceso DB2 este inicializado, compartir memoria que sea inicializado y los archivos de la base de datos sean abiertos.
- En algunos casos, RAC aplicara la generación de redo por el nodo que fallo en bloques que están presente en la cache de nodos vivos. En estos casos los bloques no tienes que ser leídos directamente del disco.
- Después de que la base de datos ha sido recuperada, las aplicaciones pueden esperar que obtengan los tiempos de respuesta más rápida en RAC porque los datos y los paquetes que necesitan por la aplicación pueden estar en la cache de los nodos vivos.

RAC permite que las conexiones que fallaron sean eventualmente distribuidas a través de los nodos sobrevivientes. Con DB2 EEE el cluster manager determina cual de los nodos sobrevivientes tomaran los discos pertenecientes a la partición que fallo. Para evitar cargas

saturadas, DB2 EEE debe de ser configurado para que cada nodo sobreviviente se adueñe de la misma cantidad de datos, esto es posible creando múltiples particiones en cada nodo.

En una configuración de alta disponibilidad, el número de particiones crece rápidamente con el número de nodos. Esto crea problemas graves:

- Es más complicado administrar el cluster. Cada partición tiene su propia configuración de parámetros, los archivos de la base de datos y los redo logs necesitan ser administrados.
- Cada recurso físico de cada nodo puede ser inutilizado. Aunque varias particiones son dueñas del mismo nodo físico, las particiones no pueden compartir memoria del buffer pool, el cache de paquetes, etc.
- La posibilidad de cargas o datos sesgados se incrementa con el numero de particiones.

Es este trabajo me enfoque solo al análisis a fondo de la alta disponibilidad de base de datos oracle, las bases de datos de Microsoft e IBM solo son un punto de referencia para oracle en cuanto a las características mencionadas en la tabla.

La siguiente tabla resume las características de cada solución de alta disponibilidad de base de datos.

	Oracle	Sql Server	IBM
Costo	✗	✓	✗
Disponibilidad	✓	✗	✓
Escalabilidad	✓	✗	✓
Administración	✓	✗	✓
Implementación de la solución	✓	✓	✗
Multiplataformas	✓	✗	✗
Desarrollo	✓	✗	✓
Performance	✓	✗	✓
Seguridad	✓	✗	✓

Cada una de las soluciones que IBM, Oracle y Microsoft ofrecen será correcta si cubre las necesidades del cliente, ya que siempre existen diferentes ambientes en los cuales cada una de ellas sería lo ideal para solucionar el problema o el requerimiento.

Todas las soluciones tienen sus ventajas y sus desventajas, pero cada una de ellas puede ser perfecta en un determinado ambiente que el cliente haya creado. Y cualquiera podrá ser la mejor si satisface las necesidades del cliente, más sin embargo, si la solución no encaja en el ambiente del cliente, no será la correcta y no se podrá explotar al máximo.

CONCLUSIÓN.

Real Application Clusters (RAC) es una nueva característica en la base de datos Oracle a partir de la versión 9i, ésta solución proporciona escalabilidad y disponibilidad en las aplicaciones de base de datos. También provee redundancia en el sistema para hacer a la aplicación más disponible y para dar consistencia de datos y servicio ininterrumpido durante las fallas.

RAC es una base de datos Oracle que tiene dos o mas instancias accedando a una base de datos compartida por medio de una tecnología de cluster. Un cluster es un grupo de maquinas o nodos que trabajan de manera conjunta para realizar una misma tarea. Esta arquitectura beneficiada la aplicación en el poder de procesamiento de la múltiples maquinas. La arquitectura de RAC también proporciona redundancia en el caso de que una se presente una falla en el sistema el operativo o algún otro componente de la infraestructura, el objetivo es que las aplicaciones pueden seguir accedando a la base de datos en cualquier instancia que siga funcionando. Para soportar esta arquitectura, en donde dos o en más maquinas se encuentran instancias activas, estas máquinas son comunicadas por un interconnect de alta velocidad para formar el cluster. El interconnect es una red física utilizada para la comunicación entre cada nodo del cluster.

Una parte esencial para el funcionamiento y sincronización de datos del cluster y la base de datos es la cache fusion, que es un mecanismo de memoria que proporciona una escalabilidad lineal.

El cache fusion elimina escrituras a disco, las cuales ocurren cuando otras instancias requieren datos para modificarlos, además de que ayuda a mejorar el performance que se utiliza para manejar los datos compartidos entre instancias.

Si bien, RAC es la solución de alta disponibilidad de base de datos proporcionada por Oracle, puede ser o no ser la mejor opción para la solución de los sistemas de alta disponibilidad, esto depende de los requerimientos que el cliente tenga en el momento de implementar la solución.

El RAC marca la pauta de una nueva tecnología en alta disponibilidad en la base de datos, pues en el pasado todos los clientes que requerían que sus aplicaciones estuvieran protegidas ante falle del sistema, del hardware o de la aplicación, tenían un cluster de hardware solamente, que se convertía en caro y obsoleto al paso del tiempo. A pesar de esto en la actualidad esta solución sigue siendo muy confiable para los clientes con base de datos grandes, el tener un cluster de hardware representa seguridad para ellos.

RAC tiene un cluster de software, en donde se aprovechan todos los beneficios de procesamiento de todos los servidores que conformen el cluster, claro que el hardware es una parte esencial para su funcionamiento del RAC, aunque los clientes aun no le tienen la

suficiente confianza por lo nuevo del producto y la tecnología, para dejar las soluciones de alta disponibilidad como fail over, poco a poco el producto ha ido ganando terreno y confianza del mercado de bases de datos por los casos de éxito que han implementado con RAC, además de que Oracle esta enfocándose con la nuevas versiones de software, y no solo a las empresas grandes sino también a las empresas pequeñas o medianas que necesitan una base de datos para almacenar su información.

Esta solución sigue en evolución constante con las nuevas versiones de software de Oracle, como en todo software siempre seguirá desarrollándose y ajustándose en base a las necesidades del hombre y del mercado laboral.

Para los profesionistas de MAC es importante conocer las tecnologías de bases de datos que existen en el mercado laboral, antes de incorporarse a él, pues es una rama muy importante en el área de sistemas, en la cual nos podemos desempeñar perfectamente, ya que entre mas preparados nos encontremos será mas fácil incorporarnos y desarrollarnos en el mundo de la tecnología, además conocer la tecnología de punta nos permite poner en practica conceptos que solo fueron manejados teóricamente durante la carrera.

Dado que actualmente me desempeño en el área de tecnología de Oracle, lo cual me ha servido para darme cuenta que las bases que te da la carrera de MAC como profesionista son magnificas para desempeñarse en el campo laboral competitivamente, pues no solamente tienes que tener el grado o saber temas técnicas, si no también ser capaz de interrelacionarse con las personas que trabajas diariamente de una manera aceptable y crear un ambiente de trabajo agradable en donde tu te sientas bien para poder desempeñarte eficientemente.

GLOSARIO.

CPU	Central Unit Procesa
Alta Disponibilidad	Sistemas con componentes redundantes que proporciona consistencia y servicio ininterrumpido, aunque algún componente de hardware o software falla, lo cual involucra algún grado de redundancia.
Backup	Un backup es un respaldo de datos para que en caso de que el sistema falle se aplique y no haya pérdidas de datos.
Balanceo de cargas	Es la característica que balancea el número de conexiones activas entre varias instancias y dispatcher de un servidor compartido para el mismo servicio.
Buffer	Es un tipo de memoria temporal para el almacenamiento de información, y sirve para eliminar el acceso de disco y disminuir la contención en disco.
Canal de Fibra	El término genérico de alta velocidad en la transferencia serial de datos recientemente estandarizada por American National Standards Institute (ANSI), se implementa mediante un canal de fibra óptica el cual es el hardware de comunicación entre los nodos del cluster.
Cache	Es la memoria temporal en donde se alojan objetos para evitar el acceso a disco y mejorar el performance de la aplicación.
CFS	Es un sistema de archivo propio de Oracle, el cual permite el acceso de todos los nodos que conforman el cluster para la lectura y escritura de los archivos que están almacenados sobre él.
Concurrencia	Acceso simultáneo a los mismos datos por diferentes usuarios. Un sistema multiusuario de base de datos deben proporcionar un control adecuado de concurrencia, ya que los datos no pueden ser actualizados o cambiados sin que se cumpla con la integridad de los datos.
Consistencia	En un ambiente multiusuario, cuando muchos usuarios pueden acceder a los datos al mismo tiempo, la consistencia de los datos significa que cada usuario observa una vista consistente de los datos, incluyendo cambios visibles hechos por las transacciones del usuario y de otros usuarios.
Data block	Es la unidad mínima de almacenamiento de la base de datos
Diccionario de datos	El conjunto central de tablas y vistas que son usadas de solo lectura las cuales hacen referencia a una base de datos particular. Un diccionario de datos almacena información como: la estructura lógica y física de la base de datos, usuarios, integridad, espacio, esquemas. Un diccionario de datos es creado cuando una base de datos es creada y se actualiza automáticamente cuando la estructura de la de la base es modificada.

Demonio	Es un programa que no es invocada explícitamente, pero esta en standby esperando a que una condición ocurra.
Dispatchers	Un proceso que habilita a los clientes para conectarse al mismo servidor sin la necesidad de un servidor dedicado para cada cliente, un dispatcher maneja y direcciona múltiples requerimientos de conexiones para un proceso de servidor dedicado.
DSS	Ambiente de base de datos y aplicaciones que ayudan a tomar desiciones o sistemas de data warehouse.
Enqueues	Estructura de memoria compartida que serializa el acceso a los recursos de la base de dato y que están asociados con una sesión o transacción.
Failover	Significa que ocurrió una falla y la recuperación ante esta falla por Real Application Clusters.
Heartbeat	Mensaje periódico que muestra que una instancia esta activa
Integridad	Reglas del negocio que dictan los estándares para datos aceptados. Estas reglas son aplicadas a las base a la base de datos utilizando constraint de integridad y triggers para prevenir la entrada de información invalidas dentro de tablas.
Interfaz	Programa o herramienta que interactúa con el usuario final, para un correcto uso del sistema.
listener.ora	Archivo de configuración que identifica el protocolo de dirección en donde el listener acepta requerimientos de conexión y los servicios que el listener escucha.
Parche	Es un software que se utiliza para arreglar problemas o actualizar un determinado software.
Partición	Es una pieza pequeña y más manejable de una tabla o un índice.
Performance	Es el rendimiento de cualquier aplicación, es decir el tiempo de respuesta de la aplicación.
Procesos	Cada proceso en una instancia de oracle realiza un trabajo específico. Dividiendo el trabajo de oracle y aplicaciones en muchos procesos, múltiples usuarios y aplicaciones pueden conectarse a una instancia simultáneamente, los procesos de oracle corren un código del servidor de oracle. Incluye proceso de servidor y procesos de background.
Row cache	La memoria que almacena accesos recientes a datos para un registro.
Raw devices	Es un dispositivo de disco que aun no ha sido formateado por un file system. Raw devices son usados para Real Application Clusters desde el storage compartido.
Redundancia	Tener dispositivo y software repetido, para que en caso de falla se tengan componentes disponibles.
Sesiones	Conexión específica de un usuario a la instancia de oracle a través del proceso de usuario.

SQL	Structured Query Language, es un lenguaje no procedural que accesa a datos, los usuarios programan en SQL que quieren hacer en la base de datos, y el compilador de SQL a analiza el procedimiento para navegar en la base de datos y regresar la tarea especificada.
Tablas	Unidad básica de almacenamiento lógico de la base de datos. Los datos son almacenados en renglones y columnas.
TNS	(Transport Network Services).
Tnsnames.ora	Es un archive que contiene net service names este archivo se necesita en los clientes, nodos.
Transacción	Es una operación en la base de datos que consume recursos del sistema y que es lanzada por el usuario final con el fin de obtener datos específicos.

FUENTES DE INFORMACIÓN.

Libros.

1. DATE , C.J. Introducción a los sistemas de bases de datos. Prentice Hall.

Documentos Completos Electrónicos.

2. CYRAN, Michele. Oracle9i Database Concepts, Release 2 (9.2) [en línea] Marzo 2004 Oracle Corporation, Part No. A96524-01 [citado abril 19, 2004]
Disponible en el sitio World Wide Web:
http://download-west.oracle.com/docs/cd/B10501_01/win.920/a95499/toc.htm
Disponible también en versiones de PDF:
http://download-west.oracle.com/docs/cd/B10501_01/server.920/a96524.pdf
3. Baylis, Ruth. Oracle9i Database Administrator's Guide, Release 2 (9.2) [en línea] Marzo 2004 Oracle Corporation, Part No. A96521-01 [citado abril 19, 2004]
Disponible en el sitio World Wide Web:
http://download-west.oracle.com/docs/cd/B10501_01/server.920/a96521/toc.htm
Disponible también en versiones de PDF:
http://download-west.oracle.com/docs/cd/B10501_01/server.920/a96521.pdf
4. BAUER, Mark. Oracle9i Real Application Clusters Concepts Release 2 (9.2) [en línea] Marzo 2004 Oracle Corporation, Part No. A96597-0101 [citado abril 19, 2004]
Disponible en el sitio World Wide Web:
http://download-west.oracle.com/docs/cd/B10501_01/rac.920/a96597/toc.htm
Disponible también en versiones de PDF:
http://download-west.oracle.com/docs/cd/B10501_01/rac.920/a96597.pdf

Artículos Electrónicos.

5. DEMEL, Sohan. Oracle9i Real Application Clusters Cache Fusion Delivers Scalability. An Oracle White Paper, Oracle Corporation Review [en línea] Febrero 2004 [citado Julio 2004]. Disponible en www.oracle.com.
6. CAI, Jack, Building Highly Available Database Servers Using Oracle Real Application Clusters, An Oracle White Paper, Oracle Corporation Review [en línea] Mayo 2004 [citado Julio 2004]. Disponible en www.oracle.com.
7. BUCH, Vineet. Database Architecture: Federated vs. Clustered. An Oracle White Paper Oracle Corporation Review [en línea] Febrero 2004 [citado Julio 2004]. Disponible en www.oracle.com.

8. CHANDRASEKARAN, Sashikanth. Technical Comparison of Oracle9i Real Application Clusters vs. IBM DB2 UDB EEE v8.1. An Oracle White Paper, Oracle Corporation Review [en línea] Octubre 2002 [citado Julio 2004]. Disponible en www.oracle.com.

Sitios Electrónicos.

9. Monografías, la Página de Monografías es para todo público interesado [en línea], [monografias.com](http://www.monografias.com), España, 2003, [fecha de consulta: 02 de Mayo del 2005], disponible en: <http://www.monografias.com/trabajos14/datos/datos.shtml>.
10. Wikipedia, la [enciclopedia libre](http://es.wikipedia.org/wiki/Software), La Página de Monografías es para todo público interesado [en línea], [citado mayo 2004]. Disponible en la Internet: <http://es.wikipedia.org/wiki/Software>.
11. Zator, La pagina Zator Monografias es para todo público interesado [en línea], [citado junio 2004]. Disponible en la Internet http://www.zator.com/Internet/A4_4.htm.
12. Microsoft, la pagina TechNet es para todo público interesado [en línea], [citado junio 2004]. Disponible en la Internet: <http://www.microsoft.com/latam/technet/articulos/200302/art01/>.
13. Tecnologías de Cluster en Windows, la pagina es para todo público [en línea]. [citado enero 2005] Disponible en la Internet: <http://www.microsoft.com/latam/technet/articulos/windows2k/clusters/>.
14. Recuperar un centro de datos mediante copias de seguridad de la base de datos, la pagina es para todo público [en línea]. [citado enero 2005] Disponible en la Internet: <http://www.microsoft.com/latam/technet/articulos/200302/art03/>.
15. Minimizar el tiempo de inactividad mediante servidores redundantes, la pagina es para todo público [en línea]. [citado enero 2005] Disponible en la Internet: <http://www.microsoft.com/latam/technet/articulos/200302/art05/>.
16. Introducción al Servicio de Cluster Server de Microsoft (MSCS) en Windows .NET Server 2003, la pagina es para todo público [en línea]. [citado enero 2005] Disponible en la Internet: <http://www.microsoft.com/spanish/msdn/articulos/archivo/100103/voices/wms-introclustermcs.asp>.
17. Federated SQL Server 2000 Servers, la pagina es para todo público [en línea]. [citado enero 2005] Disponible en la Internet: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/architec/8_ar_cs_4fw3.asp.