ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Eléctrica y Computación

"IMPLEMENTACIÓN DE UN CLUSTER ALTAMENTE DISPONIBLE, UTILIZANDO SERVIDORES IBM PSERIES CON SISTEMA OPERATIVO AIX, COMPARTIENDO UN SISTEMA DE ARCHIVOS CONCURRENTES"

INFORME PROFESIONAL DE GRADUACIÓN

Previa a la obtención del Título de:

INGENIERO ELÉCTRICO EN COMPUTACIÓN

Presentado por

PAÚL MELITÓN VERGARA GRANDA

Guayaquil - Ecuador

2013

AGRADECIMIENTO

AL DR. SIXTO GARCÍA AGUILAR
DIRECTOR DE MI INFORME PROFESIONAL, POR SU AYUDA Y COLABORACIÓN PARA LA REALIZACIÓN DE ESTE TRABAJO.

DEDICATORIA

A MI ESPOSA Y A MIS AMADAS HIJAS POR ESTAR SIEMPRE BRINDÁNDOME ESA CONFIANZA Y ALEGRÍA TODOS LOS DÍAS.

TRIBUNAL DE SUSTENTACIÓN

Ph. D. Boris Vintimilla
SUB DECANO DE LA FIEC

MSc. Guido Caicedo Rossi
VOCAL PRINCIPAL

MSc. Ignacio Marin Garcia
VOCAL SUPLENTE

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este informe profesional de graduación me corresponde exclusivamente; y el patrimonio intelectual de la misma a la **ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**"

_____

Paúl Melitón Vergara Granda

RESUMEN

El presente trabajo describe paso a paso la ejecución del proyecto de instalación de servidores y aplicaciones en Banco DelBank durante mis años que trabajaba para IBM del Ecuador en donde me desempeñaba como Especialista de Software AIX. Escogí este proyecto porque considero que abarcó una amplia gama de factores que lo hicieron único, tanto por el cambio total de plataformas y aplicaciones, así como también por la participación de diferentes proveedores que interactuaban con soluciones completamente nuevas en el mercado. Este proyecto consta de seis capítulos y en cada uno se describe en detalle la ejecución del proyecto antes, durante y después de su terminación.

En el primer Capítulo detallo la problemática inicial del cliente y las razones que tuvieron para elegir a IBM como la empresa a proveer los servicios solicitados.

Una vez aceptada la oferta IBM, en el Capítulo 2 detallo las especificaciones de cada una de las herramientas utilizadas tanto a nivel de base de datos, hardware y software operacional. Este detalle se entregó al cliente para justificar y determinar las expectativas y alcance del proyecto.

Posteriormente en el Capítulo 3 se hace un análisis del proceso de instalación previa verificación en sitio del espacio físico y de la disponibilidad contratada por el cliente, para con esto, poder definir la configuración del software operacional y la alta disponibilidad.

Una vez definida la configuración y conociendo el tamaño de la información de aplicaciones y bases de datos, en el Capítulo 4 se hace un diseño de la solución de alta disponibilidad para proteger el hardware y el software en los servidores. Además se detalla el diseño de pruebas las mismas que previamente fueron aceptadas formalmente por el cliente.

En el Capítulo 5 se detalla la implementación del software de alta disponibilidad con cada uno de los niveles participantes: hardware, sistema operativo, bases de datos y aplicaciones. Se indica además los puntos de fallas que se están eliminando al proteger ciertos recursos del sistema.

Finalmente, en el Capítulo 6 se procede a ejecutar las pruebas, simulando caídas o fallas de ciertos recursos de software o hardware con la correspondiente verificación de tiempos de recuperación de los mismos.

ÍNDICE GENERAL

ABREVIATURAS

AIX	Advanced Interactive Executive
APAR	Authorized Program Analysis Report
CPU	Central Processing Unit
DBA	Database Administrator
DS4300	Disk Storage model 4300
FastT	An IBM System Storage product line
FC	Fibre Channel
FRU	Field Replaceable Unit
FS	File System
FSs	File Systems
Gbps	Giga Bits per Second
GPFS	General Parallel File System™
HACMP	High Availability Cluster Multiprocessing
IP	Internetwork Protocol
ISA	Industry Standard Architecture
JFS	Journaled File System
LVM	Logical Volume Manager
MTU	Maximum Transmission Unit
PCI	Peripheral Component Interconnect
POWER	Performance Optimization with Enhanced RISC
PTF	Program Temporary Fix

RAC	Real Application Cluster
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RCP	Remote Copy
RISC	Reduced Instruction-Set Computer
RPD	RSCT Peer Domain cluster
RSCT	Reliable Scalable Cluster Technology
SAN	Storage Area Network
SCO	Santa Cruz Operations
SCSI	Small Computer System Interface
SGA	System Global Area
SID	System Identifier
SMIT	Systems Management Interface Tool
SMP	Symmetric Multiprocessor
SPOF	Single point of failure
SQL	Structured Query Language
SSH	Secure Shell
TCP/IP	Transmission Control Protocol/Internet Protocol
VG	Volume Group
VGs	Volume Groups
VPD	Vital Product Data
VSD	Virtual Shared Disk

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Desde el año 1992 cuando empecé a trabajar como especialista de software para IBM del Ecuador, fui entrenado para manejar el soporte a clientes en el área de sistema operativo UNIX o IBM AIX en servidores RS/6000 que luego se denominaron pSeries. Al principio contábamos con solo tres servidores instalados a nivel país, los mismos que se encontraban en Quito en empresas del gobierno. Por esta razón tuvimos que trabajar en una campaña agresiva para captar clientes y vencer a la competencia que en ese entonces eran NCR, Sun Microsystems y el IBM AS/400 muy bien posicionados en el mercado local.

Tuvimos que realizar competencias de rendimiento con otros servidores equivalentes, hicimos un sinnúmero de campaña en diferentes ciudades y al final, en 1995 empezamos a ver los frutos de lo cosechado anteriormente, cuando la base instalada superaba los 30 servidores RS/6000. Pero el principal problema era el alto costo de cada servidor, pues éstos utilizaban un hardware micro-canal, pero en corto tiempo estos fueron reemplazados por otros más veloces y baratos que utilizaban una tecnología PCI/ISA; fue entonces cuando RS/6000 empezó a llamarse pSeries.

Con Servidores más económicos y un sistema operativo AIX mucho más maduro, tuvimos que recibir entrenamiento en diferentes y nuevas soluciones para poder abastecer al mercado local el cual crecía a pasos agigantados y donde el uso compartido de recursos para abaratar costos se veía venir. Fue así que el uso de servidores SP (Escalables) y compartidos (Clusters) fue el estándar en servidores de rango medio para finales de los años 90's e inicios del 2000.

Realicé muchas instalaciones en servidores IBM *Deep Blue* (escalables y paralelos) pero los elevados precios de los servidores hizo que esta tecnología se estanque en Ecuador dando paso a los servidores en cluster que compartían recursos externos e internos, abaratando los costos considerablemente. De todas estas instalaciones que realicé, la de banco DelBank fue para mí un verdadero reto, por el hecho de utilizar software completamente nuevo en el mercado Andino y por el impacto en la región de ser el primer país con esta nueva tecnología. El proyecto fue satisfactorio y gracias a este, IBM pudo realizar instalaciones similares en otros países en América del Sur, muchas de las cuales contaron con mi soporte directo o remotamente.

CAPÍTULO 1.

1. ANTECEDENTES Y JUSTIFICACIÓN

1.1. Antecedentes

Antes de convertirse en una entidad bancaria, BANCO DELBANK empezó sus operaciones como *Delgado Travel*, una empresa cambiaria y de envío de dinero con sucursales en diferentes países. El colapso económico ocurrido en nuestro país a finales de los años noventa y posteriormente la dolarización adoptada en 1999, ocasionó que la tasa de desempleo se eleve considerablemente ocasionando una falta del poder adquisitivo, razón por la cual muchos de nuestros compatriotas tuvieron que emigrar a otros países en busca de un mejor porvenir económico. Al poco tiempo, los migrantes empezaron a enviar dinero a sus familias en Ecuador, y eligieron a *Delgado Trave*l como la empresa indicada para realizar dichas transacciones.

En vista al notable incremento de clientes que solicitaban nuevos servicios, *Delgado Travel* se vio en la obligación de convertirse en un banco, pero esta conversión implicaba actualizar todo su personal y sus sistemas de bases de datos y servidores, acorde a la nueva realidad.

1.2. Visión General del Banco DelBank y su Problemática

Para el año 2000 el sistema financiero y contable de *Delgado Travel* se ejecutaba sobre sistema operativo SCO Unix y base de datos FoxPro en computadores Intel. En vista al incremento considerable en el número de transacciones y de clientes que enviaban dinero desde el exterior, los accionistas de *Delgado Travel* tuvieron que buscar una solución que se acople a la nueva realidad actual de aquel entonces, tanto a nivel de hardware como de software.

A nivel de Solución de software *Delgado Travel* adquirió el sistema *Abanks* a la compañía *Arango Software Internationa*l, el cual se adaptaba a sus requerimientos; pero a nivel de funcional, *Abanks* requería solucionar los siguientes problemas:
Que sus archivos de base de datos y aplicaciones sean accesados concurrentemente por más de un usuario desde varios servidores Unix.

- Que esta taza de acceso y transferencia sea elevada.
- Que la administración de la base de datos se la realice con comandos del sistema operativo y no a nivel de base de datos la cual maneja archivos crudos RAW, para facilitar la administración de la base de datos.
- El espacio en disco, es decir los **filesystems** donde deben almacenarse los archivos de datos, debería incrementarse en línea sin necesidad de parar el servicio a usuarios como se lo hacía con SCO Unix.
- Que exista un sistema de protección tipo RAID a nivel de discos duros.
- Que exista protección a nivel de caídas de servidores.
- Que maneje el protocolo de acceso y protección (Lock/unlock) de archivos acorde a la base de datos Oracle 9iRAC.
- Que exista protección a nivel de adaptadores de red, tanto de usuarios como interna manejada por la base de datos.

1.3. Justificación

Se realizaron varias pruebas con diferentes servidores, tanto a nivel de alta disponibilidad como a nivel de aplicaciones. La única solución que soportaba el manejo de un sistema de archivos en filesystems concurrentes era GPFS que funcionaba sobre sistema operativo AIX (Unix de IBM) que a su vez funciona sobre servidores IBM pSeries.

A diferencia de otras aplicaciones, GPFS permitía una tasa de transferencia mucho mayor a soluciones similares y además contaba con la certificación de Oracle 9iRAC, seleccionada como la base de datos de Abanks.

Abanks funcionaba con GPFS en el esquema de alta disponibilidad anteriormente detallado, pero en América Latina no se tenía la experiencia necesaria para hacerlo funcionar. Ese fue el principal reto que tuvimos, pues por tratarse de la primera instalación en América Latina con GPFS y Oracle 9iRAC, no se iba a contar con el soporte en sitio ni de Oracle ni de IBM.

CAPÍTULO 2.

2. ESPECIFICACIONES DEL PROYECTO

2.1. Generalidades

El detalle de mi trabajo de graduación se basa en mi experiencia laboral mientras era especialista de software AIX en IBM del Ecuador. Para este caso específico, el proyecto DelBank se realizó durante los años 2003 y 2004 e IBM se adjudicó el negocio luego de participar en una licitación contra varios proveedores, entre los cuales estaban empresas que cotizaban servidores Hewlett Packard, Sun Microsystems y Unisys; finalmente Banco DelBank decide contratar a IBM del Ecuador como su proveedor de servicios tanto a nivel de servidores con sus equipos pSeries como a nivel de networking y almacenamiento.

Nuestra oferta incluía dos equipos pSeries p630 como servidores de base de datos y dos pSeries p615 como servidores de aplicaciones; además de un sistema de almacenamiento Storage Fastt600 y como sistema para alta disponibilidad entre los servidores, la aplicación tipo cluster HACMP. Se realizó una presentación técnica de la solución al Sr. Alberto Enríquez (Jefe de Sistemas), al Sr. Edison Guamán (Gerente) y al personal del departamento técnicos del Banco DelBank. Con esta presentación se aseguró el negocio con el Cliente, eso sí, haciendo énfasis que nuestra solución era la

mejor para cubrir sus necesidades en base a rendimiento de los equipos y a referencias presentadas por otros clientes en Sur América donde se está utilizando tecnología similar a la ofertada.

2.1.1. Objetivos del proyecto

La compra realizada por Banco DelBank tenía como objetivo principal el implementar una solución de alta disponibilidad que proteja y garantice el servicio tanto de servidores como del software instalado en ellos, para luego instalar la aplicación GPFS para que la base de datos Oracle pueda disponer de un sistema de directorios y archivos en paralelo. Con este esquema de archivos se facilitó de gran manera la administración de la base datos y el manejo de respaldos en contraste con el manejo de datos crudos (*raw*). Para lograr todo esto, era necesario cumplir con los siguientes objetivos específicos:

- Definir y configurar los ambientes de base de datos y de aplicaciones en servidores separados y que se protejan mutuamente en caso de alguna contingencia; para eso, se debía configurar la aplicación HACMP para poder proteger servidores ante eventuales fallas software o hardware.
- Verificar las conexiones de hardware entre los servidores para poder definir las redes internas (para el GPFS y el Oracle 9iRAC) y Públicas (para clientes).
- Proveer protección del hardware que conforma el cluster: Servidores, discos duros, fuentes de poder y adaptadores de red.
- Detectar los posibles puntos únicos de falla y tomar las respectivas políticas para su eliminación o prevención.

- Planificar la asignación del disco compartido y de los recursos del LVM y del sistema de archivos.

2.1.2. Alcance del proyecto

Antes de empezar con la ejecución, el personal de técnico y de ventas de IBM hizo un alcance del proyecto para estar seguros que éste cumplía con las expectativas de banco DelBank y sus clientes tanto externos como internos. Para lograr esto, se elaboraron planes de trabajo específicos en dos diferentes partes:

Una primera parte del proyecto fue el diseño y configuración del Sistema, que contempló la creación de un conjunto de pasos y procedimientos para la instalación, configuración y diseño con la respectiva asignación de roles y responsabilidades por parte del cliente como por parte de IBM. En esta parte se analizó la factibilidad de todos los puntos a ejecutarse, entre ellos se tomaron en cuenta lo siguiente:

- Validación del quórum de los servidores. Dos de Base de datos que se protegerían mutuamente y dos de Aplicaciones que también estarían protegidos.
- Analizar el ancho de banda de la red interna para garantizar un tráfico adecuado a la carga de trabajo.
- Verificación de los tiempos de recuperación en caso de contingencias en los servidores de base de datos y aplicaciones.

Y una segunda parte que implicó el desarrollo en si del proyecto en base a los pasos y procedimientos previamente planteados. Para esto se designó el personal a cargo

del proyecto, herramientas de consultas disponibles y un área con los respectivos equipos y software adquiridos para ser utilizada mientras durara el proyecto.

2.2. Requerimientos del proyecto

A nivel de requerimientos técnicos, banco DelBank debía contar con servidores pSeries con su respectivo software, tanto para la base de datos como para aplicaciones, así como también de un conjunto de equipos de comunicación que permitían la conexión de las redes internas de Oracle y de GPFS, así como la red conectada a usuarios.

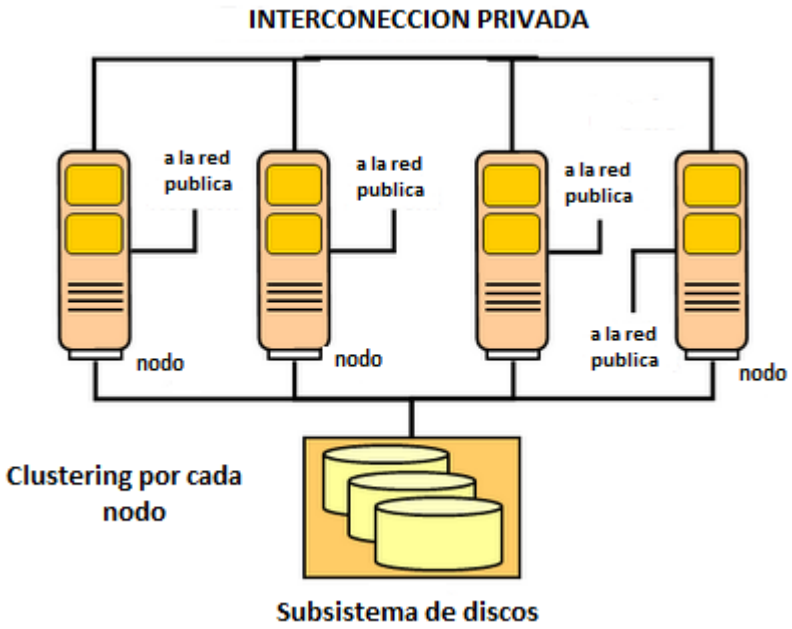
A nivel de requerimientos funcionales, el proyecto implicaba que los tres proveedores de servicios: IBM con los servidores, Oracle con la base de datos y Arango Software con la aplicación, entreguen una única solución trabajando correctamente. Esto implicaba que se debía comprobar tanto de rendimiento como de estabilidad del sistema, definiendo un conjunto de pruebas a ejecutarse antes de aceptar el proyecto como válido.

2.3. Herramientas y tecnologías utilizadas

El cliente Banco DelBank optó por adquirir una tecnología tipo cluster en los tres niveles de arquitectura: Hardware con servidores IBM pSeries; Base de Datos con Oracle y su versión de cluster 9iRAC y la Aplicación aBanks propuesta por Arango Software. Para un mejor entendimiento de la tecnología utilizada. Un cluster está formado por dos o más servidores independientes pero interconectados. Algunos clusters están

configurados de modo tal que puedan proveer alta disponibilidad permitiendo que la carga de trabajo sea transferida a un nodo secundario si el nodo principal deja de funcionar. Otros clusters están diseñados para proveer escalabilidad permitiendo que los usuarios o carga se distribuya entre los nodos.

Una característica importante de la solución tipo cluster presentada a banco DelBank, es que se presentaban a las aplicaciones como si fueran un solo servidor, permitiendo que la administración de diversos nodos del cluster con servidores AIX se la realice desde un solo punto. El software de administración del cluster provisto por el sistema operativo AIX permitió proveer este nivel de transparencia, logrando así que todos los nodos puedan actuar como si fueran un solo servidor, los archivos fueron almacenados de modo tal que podían ser accesados por todos los nodos del cluster y por todos los usuarios del sistema.



F*igura 2.1 - Servidores compartiendo recursos en un cluster (tomado de [3])*

2.3.1. Descripción de la arquitectura GPFS

General Parallel File System (GPFS) es un sistema de archivos distribuido de alto rendimiento desarrollado por IBM y que proporciona un acceso concurrente de alta velocidad a aplicaciones que se encuentran ejecutando en múltiples servidores (nodos) de un cluster. Existen versiones de GPFS para sistemas operativos AIX desde 1998 y para Linux desde el 2001 y se incluye como parte de *IBM System Cluster 1350* cuando este es comercializado con un subsistema de arreglo de discos (*tomado de [7]*). Desde el inicio, GPFS ha sido aplicado con éxito en multitud de aplicaciones comerciales incluyendo: servicios digitales y de archivos escalables.

El sistema de ficheros GPFS está compuesto de un conjunto de servidores IBM pSeries llamados nodos, los mismos que forman un cluster GPFS. Algunos de los miembros del cluster proporcionan los discos físicos accesibles por todos los nodos del sistema. La inclusión y exclusión de miembros del cluster puede realizarse en funcionamiento. Cuando un nodo realiza una operación sobre el sistema de ficheros los datos se distribuyen secuencialmente ("striping") en diferentes discos duros que físicamente podrían residir en otro(s) servidor(es) de almacenamiento. De esta forma se obtiene un mayor rendimiento al acceder a los distintos bloques en paralelo, alta disponibilidad (la información puede almacenarse en discos de dos o más servidores), recuperación en caso de fallo, seguridad, gestión jerárquica del almacenamiento y gestión del ciclo de vida de la información. La constante inclusión de nuevos servidores de discos implica una mejora del rendimiento al redistribuir la información permitiendo un mayor nivel de paralelismo en el acceso.

2.3.2. Descripción de la arquitectura Oracle 9iRAC

Oracle “Real Application Cluster” u Oracle RAC es la versión paralela para la base de datos de Oracle 9i, la cual permite correr múltiples instancias en nodos separados del cluster accesando a la misma base de datos al mismo tiempo. Para lograr esto, Oracle RAC utiliza algunas características propias del cluster y del sistema operativo. En vista que diversas instancias tienen acceso a los mismos archivos, se necesitarán espacios de almacenamiento compartido, los cuales serán accesados por los nodos del cluster de una manera concurrente. Esta concurrencia puede ser provista por hardware, utilizando almacenamiento conectados vía fibra (FC) o por software, el cual provee acceso al espacio de almacenamiento por medio de redes entre los nodos formando discos virtuales compartidos (VSD) *(tomado de [2])*.

El espacio compartido y accesado concurrentemente por los nodos, puede ser utilizado y accesado como RAW (crudo) es decir a nivel de volúmenes lógicos o como filesystems como es el caso de banco DelBank, pero en ambos casos Oracle dispone de algoritmos de bloqueo y desbloqueo de registros para poder proveer la concurrencia requerida. Las instancias de base de datos que funcionan en diversos nodos del cluster, requieren de una red confiable y de alta velocidad para mantener la interconexión y transferencia de información de la base de datos que en la mayoría de los casos son de gran tamaño. En los comienzos del servidor paralelo de Oracle, es decir el precursor de Oracle RAC, las instancias de bases de datos en los nodos del cluster no podían intercambiar bloques de datos directamente entre sus memorias “caches”, sino que esos bloques de datos tenían que ser escritos a una zona de discos compartidas por la primera instancia y entonces ser leída por la segunda.

Oracle RAC introdujo el concepto de ***Cache Fusion*** el cual permite a cualquier instancia accesar bloques de datos localizadas en otras caches de otros servidores transfiriéndolos vía interconexión rápida. Ahora, múltiples instancias de la misma base de datos proveen redundancia de las aplicaciones; esto permite tener disponible la aplicación en caso de caída de un nodo y además permite mantener equilibrada la carga entre las instancias. Los archivos de base de datos quedan almacenados en discos física o lógicamente conectados a cada nodo, de modo tal que todas las instancias activas pueden leerlos o escribirlos *(tomado de [2])*.

El software de RAC maneja el acceso a los datos, de modo tal que los cambios en los datos son coordinados entre las instancias y cada instancia ve imágenes consistentes de la base.

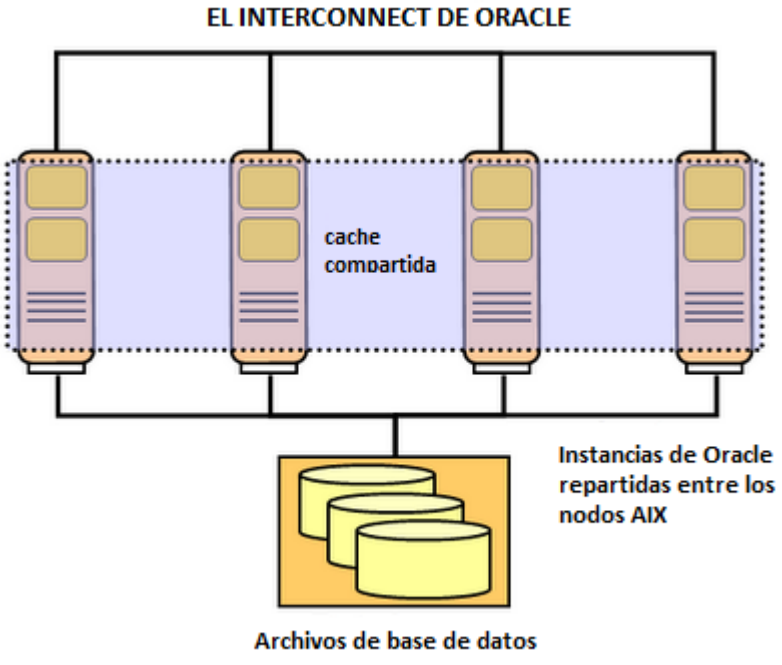


Figura 2.2. Esquema básico de interconexión de Oracle 9iRAC (tomado de [3])

La interconexión o *InterConnect* del cluster permite que las instancias se pasen entre ellas información de coordinación e imágenes de los datos. Esta arquitectura permite que los usuarios y aplicaciones se beneficien de la potencia de procesamiento de múltiples máquinas. La arquitectura RAC también ofrece redundancia; por ejemplo, en el caso de que un nodo quede inutilizado, la aplicación continuará accediendo a los datos vía el resto de las instancias disponibles.

2.3.3. Descripción de almacenamiento de discos FastT

La solución de almacenamiento de disco FastT600 fue en ese entonces la adición más reciente a la familia de productos FAStT. El FastT600 es un servidor de almacenamiento que utiliza como medio de comunicación un canal de fibra (*FC Storage Server*) de 2Gbps de nivel de entrada y salida, altamente escalable que ofrece el mejor desempeño de su clase para obtener una solución para grupos de trabajo UNIX, AIX e Intel. El *IBM Total-Storage FastT600 Storage Server* permite un incremento escalable de discos y de conexiones a servidores, con una arquitectura modular que permite soportar modelos de negocios de alta demanda de recursos de almacenamiento; Además permite una configuración de entrada que puede crecer fácilmente a la medida que crecen las necesidades de almacenamiento *(tomado de [9])*. Soporta más de 6TB de capacidad máxima, muy apropiado para consolidar almacenamiento directamente conectado en un ambiente administrado centralizadamente o compartido utilizando almacenamiento conectado vía red SAN, tecnología seleccionada por nuestro cliente.

Storage Area Network (SAN)

Es una red dedicada que provee acceso compartido de datos a varios servidores. SAN es principalmente utilizado para almacenar dispositivos como arreglos de discos, librerías de cinta y discos ópticos, los mismos que pueden ser accesados desde varios servidores como si estuviesen localmente conectados y controlados por el sistema operativo de éstos. Una SAN contiene y maneja su propia red de área local para almacenamiento de dispositivos, los mismos que generalmente NO son accesados a través de una red de área local de usuarios. Una SAN no provee un manejo de archivos sino operaciones a nivel de bloques, sin embargo permite la creación de sistemas de archivos (*filesystems*) para proveer acceso a nivel de archivo sin importar el sistema operativo que lo maneje (Windows, Linux, AIX, Unix).

2.3.4. Justificación de la tecnología utilizada

Banco DelBank adquirió una solución que se adaptaba correctamente a lo que en ese entonces tenían, tanto a nivel de clientes como a nivel transaccional. Esta solución desde el punto de vista de hardware y software ofrecía un gran nivel de **Alta disponibilidad** que permitía tener en el peor de los casos, al menos un nodo activo ante una eventual falla de alguno(s) de ellos o de una instancia de la base de datos.

Cuando utilizo el término Alta disponibilidad, hago referencia a una serie de medidas tendientes a garantizar la continuidad del servicio durante las veinticuatro horas, permitiendo a los usuarios acceder al sistema, someter nuevos trabajos o recoger los resultados de trabajos previos. La disponibilidad es expresada como un

porcentaje del tiempo de funcionamiento en un año. En un año, el número de minutos de tiempo de inactividad no planeado es dividido por el número total de minutos en un año (aproximadamente 525.600) produciendo un porcentaje de tiempo de inactividad; el complemento es el porcentaje de tiempo de funcionamiento el cual es lo que denominamos como disponibilidad del sistema *(tomado de [5]).* Valores de disponibilidad, para sistemas altamente disponibles son:

- **99,9%** = 43.8 minutos/mes u 8,76 horas/año ("tres nueves").
- **99,99%** = 4.38 minutos/mes o 52.6 minutos/año ("cuatro nueves").
- **99,999%** = 0.44 minutos/mes o 5.26 minutos/año ("cinco nueves").

Para lograr estos rangos de alta disponibilidad, tuvimos que eliminar potenciales puntos de falla como las instancias de la base de datos, nodos, adaptadores, discos y fuentes de poder; asegurando así la integridad de la base de datos en caso de una falla en algún recurso de hardware o de software. Utilizaremos aquí el término **recurso** con carácter general para referirnos a cualquier dispositivo o servicio, hardware o software, susceptible de ser compartido.

Un punto único de falla (***SPOF)*** por su sigla en inglés, es una parte de un sistema computacional que, si llegase a fallar ocasionará una caída total del sistema, dejando sin servicio a los clientes y ocasionando posibles pérdidas de información. La evaluación de un potencial punto de falla involucra la identificación de componentes críticos de un sistema que podría provocar una caída total del sistema en caso de un mal funcionamiento. Los sistemas pueden ser más robustos y seguros añadiendo redundancia en todos potenciales puntos de falla. Esta redundancia puede ser lograda a nivel de componentes internos, a nivel de sistemas

(múltiples servidores) o a nivel de sitio (replicación). En sistemas computacionales esto se logra a través de clusters de alta disponibilidad *(tomado de [8])*.

Tanto a nivel de servidores como a nivel de almacenamiento de discos externos para datos se obtuvo una gran escalabilidad, permitiendo que Banco DelBank pueda agregar en línea nuevos nodos o discos sin tener que parar el servicio a los usuarios; además, el banco DelBank pudo proteger su inversión pues la decisión de compra un nuevo nodo o nodos dependía solo del crecimiento que se necesite, pudiendo a futuro dividir la carga entre los nuevos y viejos servidores.

Con esta configuración, Banco DelBank pudo crecer y reducir el servicio computacional bajo demanda, esto es a medida que los requerimientos a nivel de recursos aumenten o disminuyan, el Administrador del sistema podrá asignar o quitar nodos o recursos de ellos para incrementar o disminuir su capacidad computacional.

Los servidores pSeries en configuración tipo cluster GPFS o HACMP, están diseñados especialmente para dar un servicio de alta disponibilidad 24x7. Además, permiten programar paradas para mantenimiento preventivo por etapas, removiendo del cluster a un nodo para soporte mientras el resto continua operando.

Finalmente a nivel de discos duros, los datos se almacenaron en configuraciones tipo RAID 1 o espejamiento (*mirroring*). Con esto se garantiza que a la pérdida o daño de un disco, el sistema operativo automáticamente recupera la información almacenada en su espejo.

CAPÍTULO 3.

3. ANÁLISIS DEL PROCESO DE INSTALACIÓN

Una de las razones del porque este proyecto fue especial desde el comienzo, fue porque se trataba de la primera instalación GPFS en América del Sur y adicionalmente por que el Cliente ya había realizado la compra de dos servidores pSeries para base de datos y dos para aplicaciones, cuando los vendedores IBM reconocieron que se había pasado por alto el tema concerniente al quórum para mantener el cluster GPFS activo, el mismo que debía mantenerse con mínimo tres nodo y no con dos como fue inicialmente configurado y vendido. Este problema fue detectado al momento de realizar la recepción de los equipos por parte del cliente, es decir antes de proceder con la instalación.

Al momento de la adquisición de la solución con IBM del Ecuador, la configuración inicial consistía de dos clusters diferentes. El primer cluster era del tipo GPFS que abarcaba dos nodos pSeries 630 y correspondía a los servidores de base de datos; el segundo cluster del tipo HACMP con dos nodos pSeries 615 y correspondía a los servidores de aplicaciones.

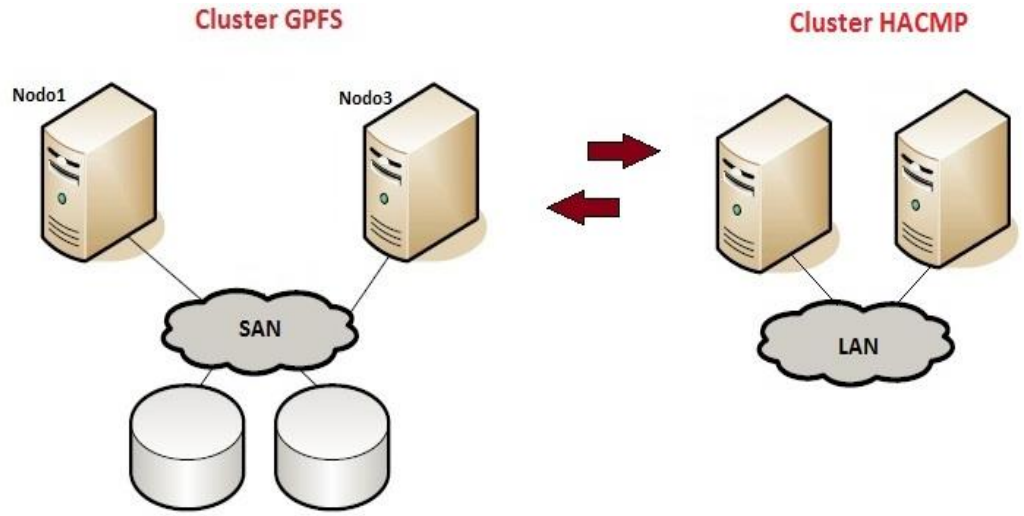


Figura 3.1 Bosquejo inicial de la solución planteada a banco DelBank (tomado de [8])

En vista de que este tipo de instalación era completamente nueva en Sur América no hubo ningún soporte local a quien recurrir, por lo que comencé a investigar el problema en la base de datos de soporte IBM y encontré que existía un caso similar reportado en Australia pocas semanas antes. En este caso abierto recién se determina que el cluster GPFS no puede mantenerse con dos nodos, pues al momento que uno de ellos falle el otro no podía quedar disponible. En este mismo caso se indicaba que había una opción que podía garantizar el funcionamiento del cluster GPFS con dos nodos y estaba soportada en la versión 8.4 del software de administración de los discos FastT600 *IBM Storage Manager*. Esta versión permitía cambiar un parámetro llamado **Persistent Reservation** con el que se lograba administrar un cluster de dos nodos *(tomado de [9]).*

La versión del Storage Manager que arribó para Banco DelBank era la 8.3, la misma que no contaba con la opción descrita anteriormente. A esa fecha el caso problema en mención seguía abierto y no existía certeza aun si el problema estaba solucionado con el Storage Manager v8.4 y además su actualización implicaba un gasto superior a los USD 25.000, por lo que optar por cambiar el parámetro de “Persistent Reservation” quedaba descartado.

Finalmente, IBM tuvo que asumir el costo de adicionar un nodo pSeries p615 para poder cumplir con la solución exigida por el cliente y para lograr el requerimiento mínimo de quórum para GPFS. Este cambio implicaba adicionar tarjetas Gigabit Ethernet y switch adicional con al menos 8 puertos Gigabit Ethernet TX para la interconexión de los nodos del cluster con HACMP y GPFS.

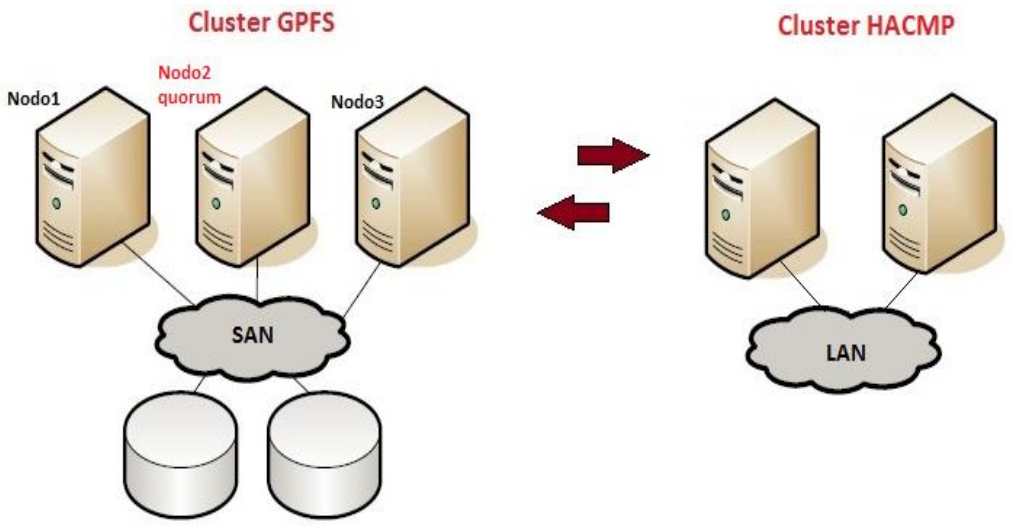


Figura 3.2 Bosquejo final de la solución para banco DelBank (tomado de [8])

En esta figura se puede observar los dos clusters; el primero con tres nodos que define el cluster GPFS con servidores de base de datos, mientras el segundo con dos nodos, definiendo el cluster HACMP de aplicaciones. Claramente se observa una red tipo SAN (Storage Area Network) conectando a los nodos del primer cluster a los discos de almacenamiento externo tipo FastT600, mientras los servidores de aplicaciones, conectados a una red de área local (LAN), también se encuentran físicamente conectados a los servidores de base de datos.

3.1. Preparación de la instalación GPFS

Los pasos previos a la instalación del cluster GPFS, incluyeron el cumplimiento de un grupo de pre-requisitos que fueron analizadas y ejecutadas por los Ingenieros de Sistemas de IBM y DelBank. Se tomaron en cuenta un conjunto de consideraciones relacionadas con los recursos de hardware y el ambiente operacional para poder llegar a una instalación exitosa.

El núcleo del cluster GPFS es un filesystem concurrente y antes de crearlo se debió especificar su tamaño en base a los archivos a ser contenidos dentro de éste y la manera como los datos debían ser accesados en el menor tiempo posible; para lograr esto se necesitó cumplir un conjunto de pre-requisitos los mismos que se inter-relacionaban con el sistema operativo AIX y con el hardware y que se detallan a continuación en la siguiente tabla.

Pre-requisito		Detalle	Cumple
1	**Hardware Necesario**	Al menos tres servidores IBM pSeries	Si
		Espacio en disco suficiente para la creación de filesystems compartidos	Si
		Al menos una red de área local con el ancho de banda de al menos 100Mbs	Si
2	**Software Necesario**	GPFS 2.1 y parches requeridos	Si
		Sistema Operativo AIX 5.2	Si
3	**Recuperación de recursos**	Datos	Si
		Nodos	Si
		Redes y adaptadores	Si

		Discos	Si
		Sistema operativo	Si
4	**Almacenamiento**	FastT600 modelo 1722-60U o IBM DS4300 con conexión vía canal de la fibra FC	Si
5	**Creación de cluster**	Utilizando comandos propios de GPFS	Si
		GPFS con RPD RSCT	Si
		HACMP	No
6	**Acceso remoto a los nodos**	Vía comando rcp	Si
		Ssh	No
		Acceso vía archivo /.rhosts	Si
7	**Creación de FS**	Vía comandos GPFS	Si
		Montaje automático	Si
		Acceso concurrente por los tres nodos	Si
		Se utilizará replicación de datos para su protección	Si
		Quota de filesystems en los *VGs*	Si

Tabla 1. Lista de prerrequisitos a cumplir para instalar GPFS (tomado de [6])

3.1.1. Definición de versiones de sistema operativo y software operacional

El sistema operativo AIX 5L versión 5.2 que fue adquirido por el Cliente incluía muchas ventajas tanto a nivel de administración como a nivel de utilitarios las mismas que en otras variedades de Unix tenían un costo adicional. El software adquirido por el Banco DelBank y que tuvimos que instalar en el hardware ya antes mencionado incluía herramientas poderosas que iban a ser de mucha utilidad para la

administración diaria de los servidores. Algunas de estas herramientas se nombran a continuación:

- Manejo de Volúmenes lógicos (LVM) utilizado para la creación en línea de espejamiento (RAID 1) de discos, tanto a nivel de sistema operativo como de datos.
- Soporte de *Journal* Filesystem (JFS2) y administración de FileSystems.
- Herramientas de monitoreo de recursos y carga (*Workload* manager)
- Ejecución de comandos vía menús vía SMIT
- Ejecución de comandos en un sistema distribuido.
- Manejo de direcciones IP virtuales
- Autenticación vía *Kerberos* V5
- Manejo inteligente de subsistemas de sistema operativo.

Todos los servidores, tanto los de bases de datos como de aplicaciones, fueron configurados con la versión de sistema operativo AIX 5.2 los mismos que debían mantener la misma configuración (tanto a nivel de versión, *release* y modificación), con los mismos componentes y aplicativos para cada uno de los casos. Para la correcta definición de espacios de paginamiento y espejamiento del sistema operativo, se realizó el análisis de cada uno de los servidores. Una vez realizada la correspondiente verificación de hardware y GPFS, se procedió a la instalación del sistema operativo AIX V5.2, GPFS 2.1, Oracle 9i RAC 9.2.0.2 manteniendo la siguiente estructura jerárquica:

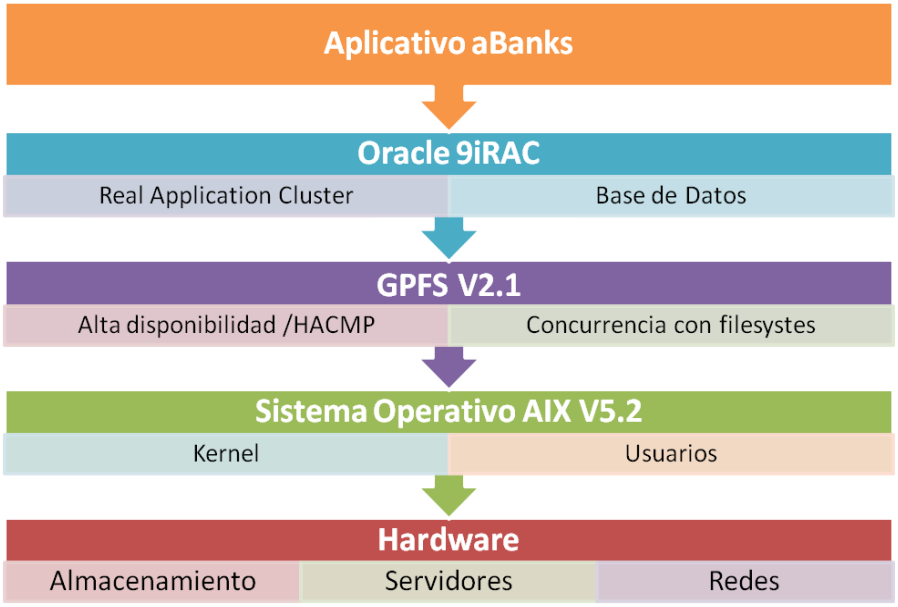


Figura 3.3 Estructura jerárquica del software en el cluster

Esto quiere decir que la instalación empezó con la configuración de hardware la misma que incluía configurar el almacenamiento FastT6000 con la interface SAN, los servidores pSeries y las redes públicas y privadas. Luego se procedió a la instalación del sistema operativo AIX 5.2, después el GPFS versión 2.1 y finalmente Oracle 9iRAC. La instalación del aplicativo aBanks, corrió a cargo de la empresa. La verificación del sistema operativo se la realiza ejecutando los comandos de sistema operativo **lppchk –v**. Un ejemplo de la ejecución de este comando se muestra a continuación:

```
bos.net.tcp.server
5.2.0.10    COMMIT       COMPLETE     01/09/04     08:35:15
5.2.0.11    COMMIT       COMPLETE     01/09/04     20:09:45
5.2.0.14    COMMIT       COMPLETE     01/09/04     20:45:29
bos.net.tcp.smit
5.2.0.10    COMMIT       COMPLETE     01/09/04     08:35:15
5.2.0.11    COMMIT       COMPLETE     01/09/04     20:45:27
bos.net.uucp
5.2.0.10    COMMIT       COMPLETE     01/09/04     08:35:15
bos.perf.libperfstat
5.2.0.10    COMMIT       COMPLETE     01/09/04     08:35:06
5.2.0.12    COMMIT       COMPLETE     01/09/04     20:45:27
bos.perf.perfstat
```

```
5.2.0.10   COMMIT       COMPLETE     01/09/04     08:35:06
5.2.0.11   COMMIT       COMPLETE     01/09/04     20:09:45
5.2.0.12   COMMIT       COMPLETE     01/09/04     20:45:28
bos.perf.tools
5.2.0.10   COMMIT       COMPLETE     03/16/04     23:31:27
5.2.0.14   COMMIT       COMPLETE     03/16/04     23:35:12
bos.perf.tune
5.2.0.10   COMMIT       COMPLETE     01/09/04     08:35:06
5.2.0.11   COMMIT       COMPLETE     01/09/04     20:09:45
5.2.0.13   COMMIT       COMPLETE     01/09/04     20:45:28
bos.rte
5.2.0.10   COMMIT       COMPLETE     01/09/04     08:28:30
5.2.0.11   COMMIT       COMPLETE     01/09/04     20:45:27
```

Figura 3.4 Salida del comando lppchk -v

Donde 5.2.0.11 en el fileset **bos.rte** indica lo siguiente: 5: Versión de sistema operativo. 2: Nivel de liberación del software. 0: modificación y 11: nivel de PTF. **COMMIT** significa que ese fileset ha sido fijado como parte del kernel.

3.1.2. Definición de redes internas y externas

Las redes externas son las redes públicas que se conectan a los usuarios del sistema, en cambio las redes internas son las de uso privado utilizadas por GPFS y Oracle para la respectiva configuración y normal operación de sus clusters. Una red interna también conocida como privada no permite ser accedida por usuarios externos sino solo por aplicaciones dedicadas y por los administradores del sistema. Cuando se definió las redes internas y externas, se tuvo que diferenciar los servidores de base de datos con los de aplicaciones.

Los servidores de aplicaciones que incluían los dos pSeries 615, tenían configurado HACMP para proteger posibles fallas de adaptadores Ethernet o fallas de procesador, fuentes de poder o falla de los mismos servidores; no estaban

conectados a las redes GPFS y Oracle RAC, pues únicamente ejecutan las aplicaciones de Abanks, solicitando información y requerimientos a los servidores de base de datos vía red externa, es decir la misma red que comparten todos los usuarios del sistema. Las redes privadas en los servidores de aplicaciones eran las definidas por HACMP e incluían las redes seriales rs232 y eran las encargadas de identificar como última instancia si un servidor estaba caído o no y su estatus actual. A nivel de complejidad, la red de servidores de aplicaciones era mucho más sencilla y simple y utilizaba los siguientes recursos:

- Dos puertos seriales por servidor pSeries, para definir la única red privada.
- Dos adaptadores ethernet, el uno para servicio a usuarios y el segundo para adquirir una dirección temporal al momento de reiniciar el servidor (*boot*).
- Un hub o un switch para la respectiva conexión de las redes externas.

El proceso de reintegración del segundo nodo previo a una caída implicaba realizar ciertas validaciones para que no duplique la dirección IP. Al terminar de reiniciar se tuvo que asignarle una dirección de "*boot*" temporal para verificar si su dirección IP de servicio a usuarios estaba activa; si este era el caso, éste nodo enviaría una señal vía "*heartbeats*" para solicitar que se libere la IP solicitada.

En los servidores de base de datos, el diseño de los clusters incluía los dos servidores pSeries 630 y un pSeries 615. Todos los servidores iban conectados por fibra al switch y al almacenamiento de disco fastT600. Los servidores de base de datos pSeries 630 y el pSeries 615 conectados a la red GPFS y Oracle RAC, tenían 4 tarjetas de red Ethernet: 2 de 10/100/1000 y 2 de 10/100; las dos de 10/100/1000 se utilizaron para la red primaria y secundaria de Oracle (punto a punto); una de

10/100 fue utilizada para GPFS, que era un requisito del producto, y la restante es la que se conectaba a la red del banco para el acceso de usuarios.

En el gráfico que se muestra a continuación se muestran los servidores de base de datos y de aplicaciones conectados a sus redes privadas y externas. La externa se muestra en amarillo y conecta los dos ambientes, mientras la red GPFS se muestra con las líneas en rojo, la red Oracle redundante (red de respaldo) con las dos líneas en azul y finalmente la red interna que conecta al almacenamiento FastT600, con color verde. Otra red que se configuró es la red privada serial rs232 que se muestra en azul intenso y que a diferencia de la Ethernet, es mucho más lenta, en rango de los 9600Kbps.

Esta red serial era punto a punto, es decir establecida entre dos puertos seriales de dos nodos diferentes. La finalidad era indicarnos el estado de un nodo mediante el envío y recepción de pequeños paquetes de datos (**handshaking)**. Es decir, que al ser una red directa o punto a punto que no utilizaba ninguna fuente de poder como en el caso de un HUB Ethernet y en consecuencia resultaba ser mucho más confiable pues no existía el punto de falla en caso de avería o fallas eléctricas *(tomado de [1]).*

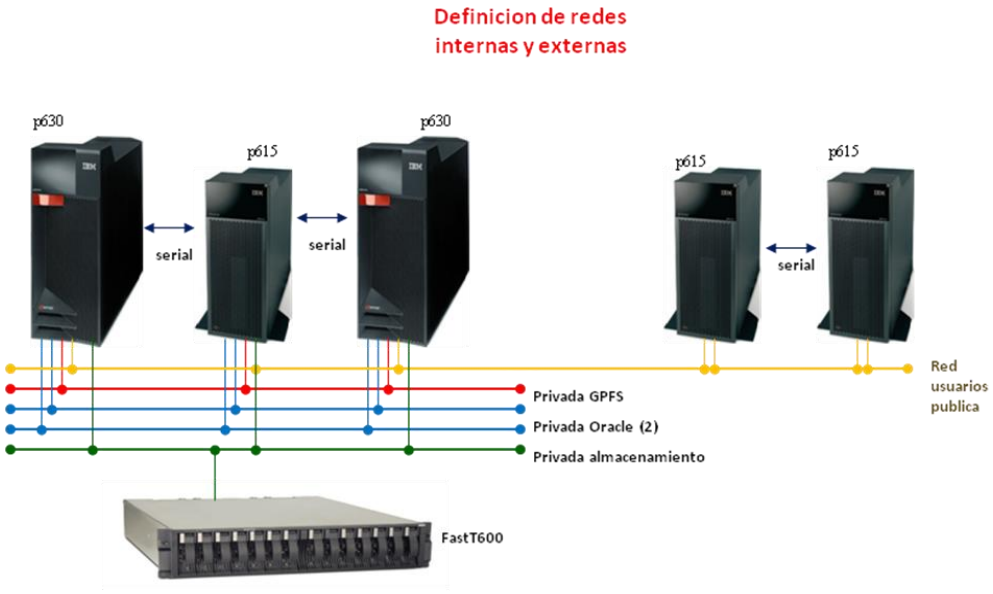


Figura 3.5 Definición de redes internas y externas

Como cada servidor estaba diseñado con dos puertos seriales rs232, no hizo falta instalar una regleta serial externa de 8 puertos y se pudieron conectar directamente vía cable serial. Los 3 servidores de base de datos contaban con 6 adaptadores (4 ethernet y 2 seriales) formando las siguientes redes:

- 1 red pública para servicio a usuarios con tarjeta Ethernet 10/100 Mbps.
- 1 red privada GPFS con tarjeta Ethernet 10/100 Mbps.
- 2 redes privadas de Oracle RAC con tarjetas Ethernet 10/100/1000 Mbps.
- 2 redes privadas seriales rs232

3.1.3. Definición de nodos primarios y secundarios

En el ambiente de base de datos los tres nodos que se conectaban tanto al GPFS como al Oracle RAC trabajaban en modo concurrente, es decir que tenían igual

jerarquía y estaban ejecutando accesos a la base de datos ininterrumpidamente, repartiendo la carga de trabajo de acuerdo a las configuraciones y requerimientos de la aplicación aBanks y de la base de datos, para mantener un repartimiento de carga balanceada. Estaba claro que los servidores pSeries 630 al ser los que mayor capacidad de procesamiento tenían la capacidad de recibir mayor cantidad de trabajo tipo *batch* o transaccional que el pSeries 615. Así mismo, en el supuesto caso que uno de los servidores deje de funcionar, los otros dos se dividirían automáticamente la carga de trabajo, dejando al GPFS que maneje la disponibilidad de archivos y de Filesystem entre los dos restantes. Un ambiente concurrente muestra a los nodos compartiendo la misma cantidad de discos, la misma información como se muestra en el gráfico siguiente:

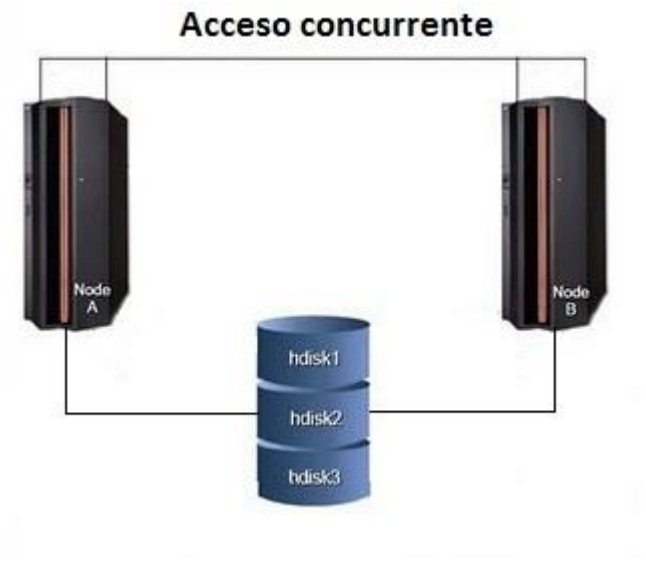


Figura 3.6 Servidores pSeries compartiendo un rack de discos

En el ambiente de aplicaciones, ambos servidores pSeries 615 contaban con la misma capacidad de procesamiento y la alta disponibilidad estaba manejada por HACMP. En vista a que en este ambiente los servidores no manejaban acceso a

disco de manera concurrente como en el anterior, se analizó los posibles accesos de disco que el software HACMP maneja:

- Hot-standby
- Third-party takeover
- Mutual takeover

De estas tres configuraciones, la más indicada para DelBank fue la de **Mutual Takeover**, es decir, la configuración donde ambos servidores se protegían mutuamente mientras ambos estaban en producción ejecutando las aplicaciones aBanks. Es decir que en el ambiente a aplicaciones tampoco existía ninguna jerarquía entre servidores y en el caso fortuito o no, de que uno deje de operar, el otro adquiriría todos recursos del nodo caído y operaría con la doble carga en pocos segundos.

El gráfico a continuación puede detallarnos de una mejor manera las prioridades similares de cada uno de los nodos con sus recursos compartidos. Aquí se puede observar a los nodos delsrv01 y delsrv02 compartiendo discos externos y protegiéndose mutuamente ante una eventual falla de algún recurso.

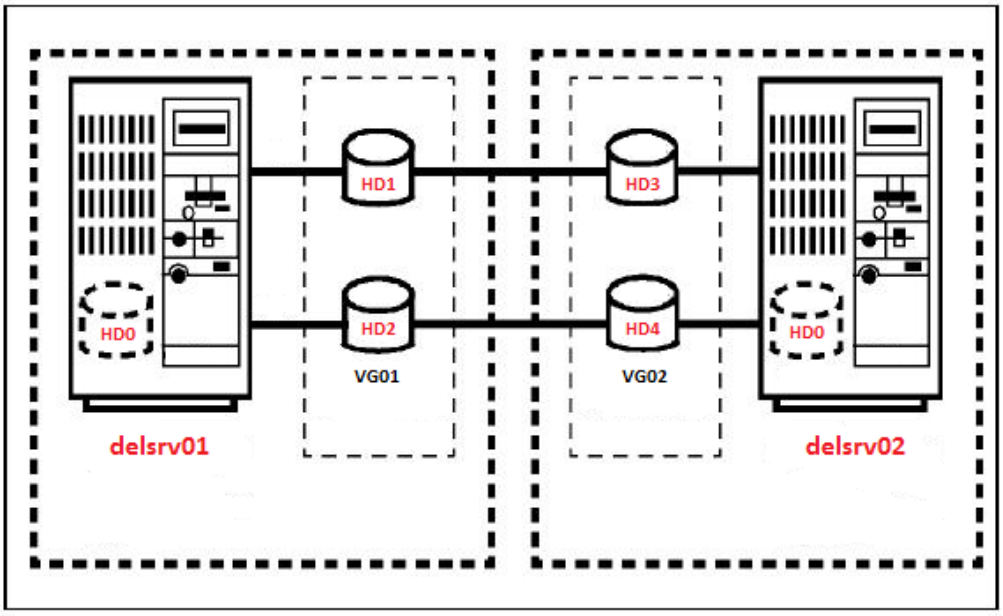


Figura 3.7 Ambiente cluster en mutual takeover (tomado de [8])

Si por ejemplo, el servidor delsrv01 fallaba, en servidor delsrv02 tomaba control de los grupos de volúmenes (**volumen groups**) VG01 y VG02, direcciones IP públicas y control de las aplicaciones de ambos servidores.

3.1.4. Definición de quórum de servidores en el cluster GPFS

Una vez que se instalaron los tres servidores de base de datos, el quórum de los mismos quedó definido en un 100%, obviando de esta manera el problema de falta de quórum causado por la caída de uno de ellos cuando se contaba con dos nodos, en cuyo caso, cuando uno de ellos fallaba, el quórum se reducía al 50% por debajo del 51% mínimo permitido por GPFS.

Con tres nodos, al momento de la caída o falla de uno de ellos, el quórum quedaba reducido al 66.66% pero el cluster estaba activo y este fue uno de los requerimientos de nuestro cliente. Al momento de una caída de un segundo nodo, el quórum se reduciría al 33.33% lo que implicaba que el cluster GPFS dejaba de funcionar. Este problema de la falta de quórum fue eliminado con la liberación de la nueva versión 8.4 del Storage Manager con la cual se pudo activar el parámetro **Persistent Reservation** y poder obtener un quórum con solamente dos servidores, pero ocurrió más de 12 meses después de la instalación inicial. El gráfico mostrado a continuación, permite observar como el quórum cambia dependiendo del número de nodos caídos.

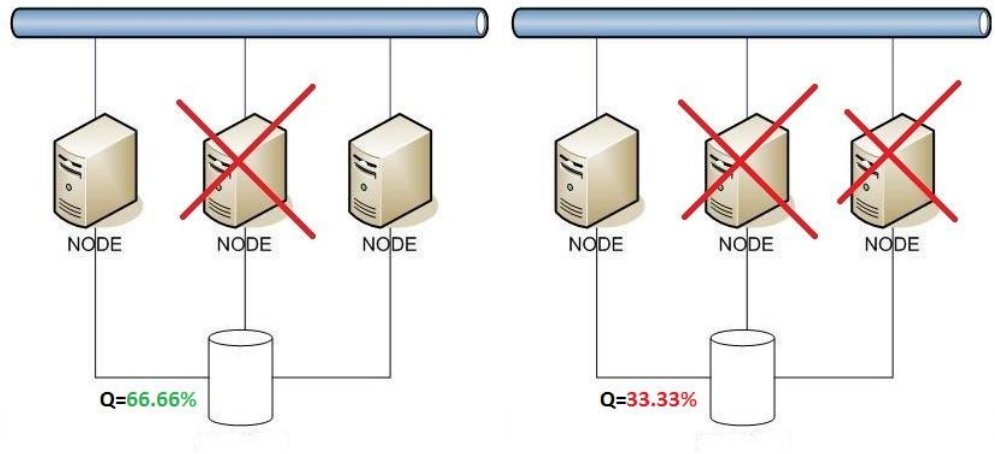


Figura 3.8 Cambios del quorum por caídas de nodos (tomado de [9])

En la actualidad con la versión del Storage Manager V10.0, el cluster GPFS, puede ser manejado inclusive con un solo nodo, igualando la disponibilidad de servidores obtenida con GPFS.

3.1.5. Configuración del arreglo de discos en un ambiente cluster

Históricamente las primeras centrales de datos o *data centers* crearon arreglos aislados de discos SCSI, los mismos que estaban dedicados a una aplicación, conectados directamente a un servidor y numerados como discos virtuales o unidades lógicas (LUNs). Los sistemas operativos mantenían sus propios filesystems dedicados y definidos sobre discos con LUNs no compartidos; si múltiples servidores intentaban compartir un LUN, esto podría interferir con otro, produciendo una corrupción de datos. Esta limitación fue subsanada con la tecnología SAN, la misma que permitió la consolidación de varios espacios de almacenamiento restringidos a un servidor, en arreglos de discos accesados por varios servidores, logrando así un incremento de la capacidad de espacio compartido.

El modelo de sistema de discos 1722 60U o IBM DS4300 utilizado por nuestro cliente DelBank, fue construida sobre la tecnología del canal de la fibra de 2 Gbps (*FC*) la misma que permitía configurar sistemas de discos altamente disponibles con capacidad escalable y conectividad creciente, permitiendo de esta manera una amplia gama de los usos de la red de almacenamiento SAN. La base DS4300 permitía hasta 14 discos internos y hasta 4TB de almacenamiento. Conectados en cascada con sistemas de expansión se lograba hasta 18TB utilizando canal de fibra. Inclusive permitía manejar discos intercambiables en caliente (**hot-swappable**), ayudando a evitar puntos de falla sin tener que detener el sistema para el reemplazo y sincronización de los mismos. La unidad de discos IBM DS4300 soporta configuraciones de RAID 0, 1, 3, 5 y 10 y estaba administrada por el software

Storage Manager V8.3 la cual permitía configurar los discos de una manera gráfica y sencilla utilizando interface Java y conexión Web.

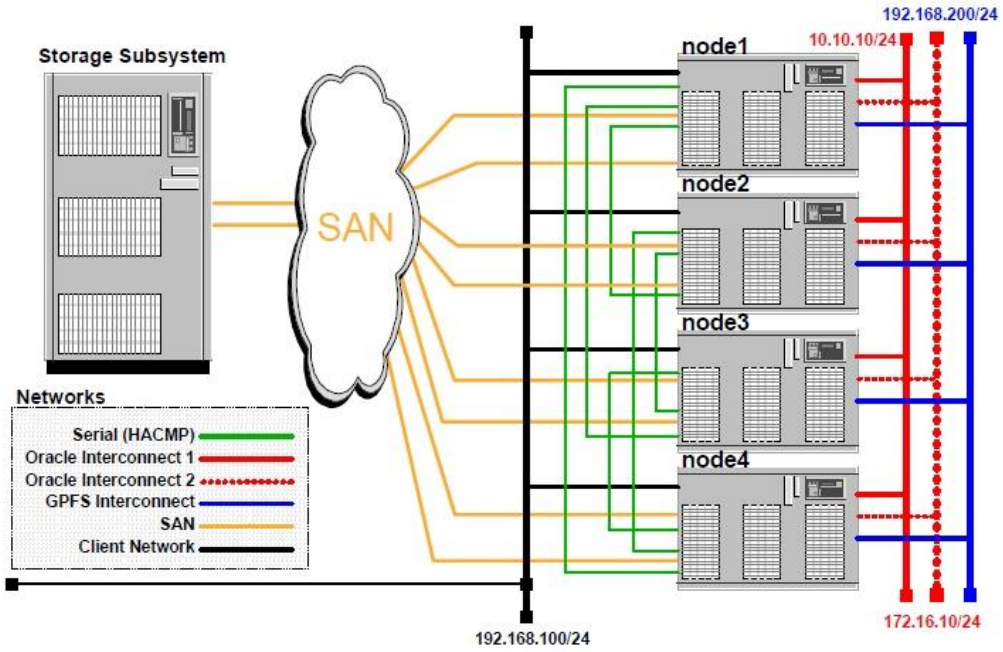


Figura 3.9 Subsistema de discos en un cluster (tomado de [2])

En esta se puede observar un conjunto de cuatro nodos pSeries conectados a una red SAN, por medio de switch FC. El subsistema de almacenamiento soporta accesos concurrentes desde cualquiera de los cuatro servidores. En este diagrama se puede observar las conexiones GPFS y Oracle RAC y una red de clientes. En una transacción típica, el cliente puede realizar consultas o actualizaciones de base de datos a través de la red de usuarios en color negro; el servidor que recibe la transacción realiza la localización física del registro de la base de datos sin importar donde se encuentre el almacenamiento. En vista a que una SAN no provee un manejo de archivos sino operaciones a nivel de bloques, GPFS hace la labor de creación de sistemas concurrente de archivos (*filesystems*) para proveer acceso a

nivel de archivo que necesita Oracle RAC. Ambas aplicaciones se denotan en azul y rojo respectivamente.

3.2. Preparación de la instalación HACMP

El objetivo principal por el cual se utilizaba HACMP en los clientes con configuraciones tipo cluster, era la eliminación puntos de fallas únicos. HACMP proporcionaba instalaciones numerosas que se podían utilizar para construir clusters "altamente disponibles". El diseño del cluster que proporcionaba la mejor solución para DelBank, requirió el planeamiento cuidadoso para una mejor disponibilidad. HACMP fue definido únicamente en los dos servidores de aplicaciones*, pues a nivel de los de base de datos, GPFS y Oracle 9iRAC proveían el servicio de alta disponibilidad que protegían los discos, adaptadores de Oracle RAC y servidores*; *en cambio a nivel de adaptadores GPFS y de usuarios no habían redundancias de los mismos por lo que HACMP quedó descartado (tomado de [8]).*

Para proveer una correcta protección de los recursos del cluster, se verificó que ninguno de estos debía tener un punto de la falla y parte de la preparación de instalación HACMP implicó identificar y tratar de eliminarlos. Los puntos verificados fueron los siguientes:

- Los servicios que se requieren estar altamente disponibles
- La prioridad de estos servicios
- Costos de una falla comparada con el hardware necesario para eliminarla.
- Disponibilidad requerida de estos servicios (24x7)
- Lo que sucedería si se interrumpe la disponibilidad de estos servicios.

- Tiempos necesarios para substituir un recurso fallado.
- El grado aceptable de degradación de funcionamiento después de una falla.
- Las fallas que se detectan como eventos del cluster.
- El nivel de formación del grupo que ejecutará el cluster en el cliente.

En la siguiente tabla se hace un sumario de los posibles puntos únicos de falla y el procedimiento para su eliminación y si estos aplicaban o no para DelBank.

Recursos del Cluster	Como eliminar punto de falla	Aplica
Nodos	Utilizando múltiples nodos	Si
Fuentes de Poder	Con múltiples circuitos o fuentes de poder	Si
Adaptador de red	Con adaptadores redundantes	Si
Red	Con múltiples redes para conectar los nodos	No
Subsistema TCP/IP	Usando redes seriales conectadas a nodos	Si
Adaptadores de disco	Con adaptadores redundantes	No
Controladoras	Con controladoras redundantes	No
Discos	Con hardware redundante y espejamiento	No
Aplicaciones	Asignando un nodo para reemplazar al caído	Si

Tabla 2. Listado de los posibles puntos únicos de falla a eliminar

En base a los puntos de falla detallados anteriormente, se procedió a definir la información relevante para poder instalar y configurar el cluster denominado **delapp** el cual contenía dos nodos asociados a un grupo de recursos **cascading**; con esta configuración se permitía que un recurso sea tomado por uno o más nodos de acuerdo a la prioridad de los mismos. En nuestro caso, no existían recursos externos compartidos, pues las aplicaciones, el motor de base de datos y adaptadores residían dentro de cada

nodo por lo que ambos nodos mantenían la misma prioridad. En caso de falla de un nodo, el segundo tomaba control de las direcciones IP de usuarios y seguía operando con la carga de usuarios de los dos nodos, pues el motor de la base de datos y las aplicaciones estaban duplicados entre ellos en los discos internos y no hizo falta manejarlos como recursos compartidos. Al momento de reintegrar al segundo nodo, éste debía reclamar su dirección IP sin duplicarla en la red. Para lograr esto se le asignó una dirección de **boot** temporal para verificar si su IP estaba activa; de ser así, éste nodo enviaba una señal vía **heartbeats** para solicitar que se libere la IP solicitada. El siguiente gráfico muestra la configuración del cluster HACMP a ser instalada en banco DelBank:

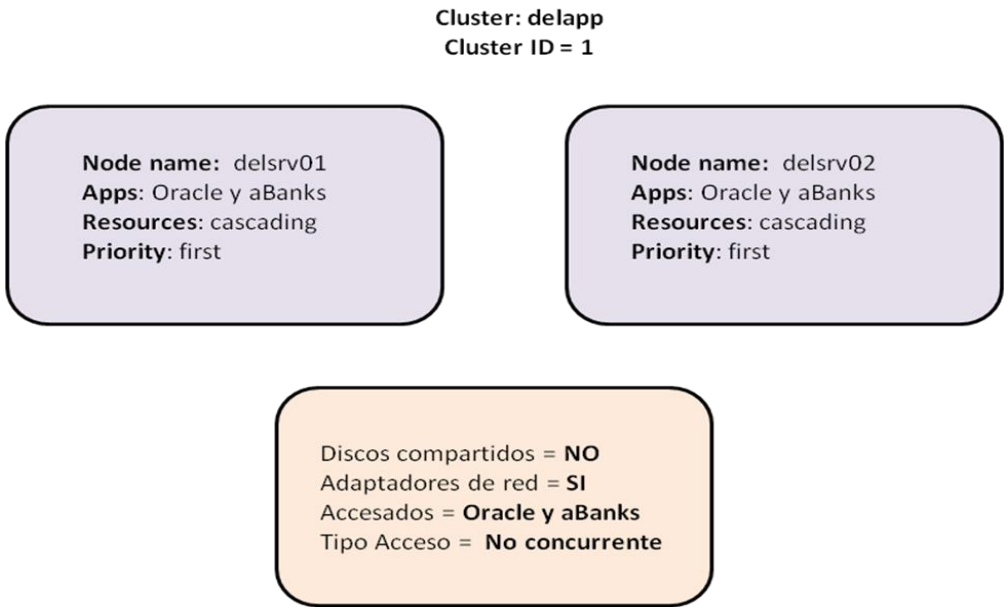


Figura 3.10 Configuración HACMP para Banco DelBank (tomado de [8])

El cluster delapp definido con dos nodos: delsrv01 y delsrv02 no compartía discos duros externos por lo tanto no hubo ninguna definición de accesos concurrentes. Lo que HACMP protegió fueron los servidores y los adaptadores de red en una configuración

“mutual takeover”, es decir cualquiera de los dos estará en capacidad de tomar control del otro.

3.2.1. Definición de los adaptadores de red

En el punto 3.1.2 se hizo la definición de las redes públicas y privada para HACMP y la respectiva diferenciación entre ellas. Es claro que la única definición de red interna con HACMP era la red serial rs232 que servía para la verificación en última instancia del estado de las interfaces de red, dispositivos de comunicación y de los nodos. En cambio, la red pública era la de usuarios. En Banco DelBank no se definió una segunda red para proteger el servicio a usuarios, es decir que en este caso existió un punto de falla. El gráfico que bosqueja la red y sus adaptadores es la siguiente:

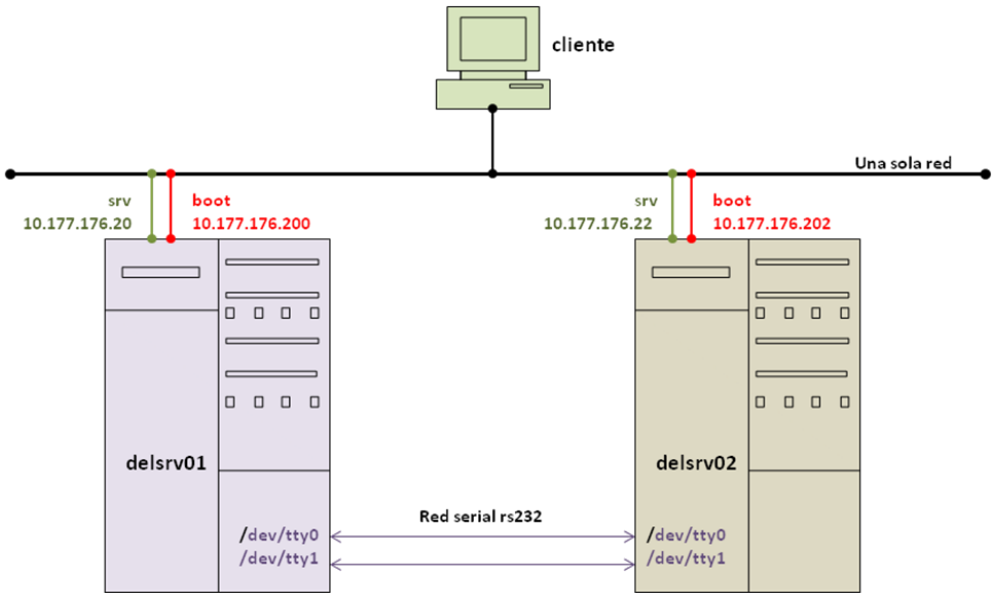


Figura 3.11 Servidores de aplicaciones con sus respectivos adaptadores

Cada servidor fue configurado con dos adaptadores Ethernet, uno de servicio a usuarios con las direcciones 10.177.176.20 y 10.177.176.22 y el otro de boot para la correspondiente verificación e inclusión al cluster luego de una caída del nodo. El problema de esta configuración es que no se definió redundancia de red y en este caso una caída del switch que conectaba los servidores con los clientes, implicaba una falla en el servicio y en consecuencia es un punto único de falla. El cliente aceptó éste caso, pues le resultaba más económico tener un switch de respaldo que realizar un doble cableado estructurado a nivel de racks.

3.2.2. Definición del tipo de configuración del cluster a utilizar

Cuando se definen recursos en HACMP, se deben agruparlos en diferentes modalidades dependiendo de la disponibilidad y costo de los mismos. Ejemplo de recursos son: discos duros compartibles, tarjetas de red, servidores, redes y grupo de volúmenes con sus respectivos datos. HACMP ha implementado diferentes mecanismos para agrupar recursos y manejarlos en caso de falla:

- *Recursos en cascada*: El nodo con la más alta prioridad controla el grupo.
- *Recursos concurrentes:* Todos los nodos accesan al grupo de recursos.
- *Recursos rotacionales*: El nodo con la dirección IP de servicio controla el grupo de recursos. *(tomado de [5]).*

En vista que los servidores de aplicaciones no manejaban discos duros concurrentes ni tampoco una sola dirección IP de servicios sino dos, el *Grupo de recursos elegido para DelBank fue el tipo cascada*, pero ambos nodos con la misma prioridad.

Tomando en cuenta la no concurrencia en el acceso a discos, se debía determinar una de las siguientes configuraciones HACMP referente a los recursos de discos no concurrentes:

Hot Standby: Es un cluster conformado por dos nodos y solo uno de ellos puede manejar los discos duros compartidos mientras el otro permanece en alerta (standby) en espera de la ocurrencia de alguna falla del nodo principal para tomar su lugar.

Mutual takeover: Conformado por al menos dos nodos, cada uno manejando su propia información y datos en discos que son conectados físicamente entre dos o más nodos. Al momento de falla de unos de éstos, un segundo nodo con la siguiente prioridad (definida en el recurso cascada) importará los discos del nodo caído y sus direcciones IP de servicios, empezando a trabajar con la carga de los dos nodos.

Third-party takeover: consta de por lo menos tres nodos, pero uno de ellos permanece inactivo pero conectado físicamente a los discos compartidos. Al momento que uno de los otros nodos falle, el nodo inactivo tomará el control del nodo caído.

De éstas tres configuraciones, la que se seleccionó para banco DelBank fue la de *mutual takeover* la cual manejaba un arreglo de recursos en cascada con un conjunto de nodos teniendo la misma prioridad. En nuestro caso, el recurso comprendía los dos nodos de aplicaciones que manejaban diferentes direcciones IP para usuarios pero que no compartían discos o almacenamientos entre sí *(tomado de [8]).*

En el gráfico que se muestra a continuación se puede observar que en cada nodo se definen dos grupos de recursos: delsrv0N_rg1 para los nodos y delsrv0N_rg2 para los adaptadores de red, donde N=1,2 dependiendo del nodo de aplicaciones.

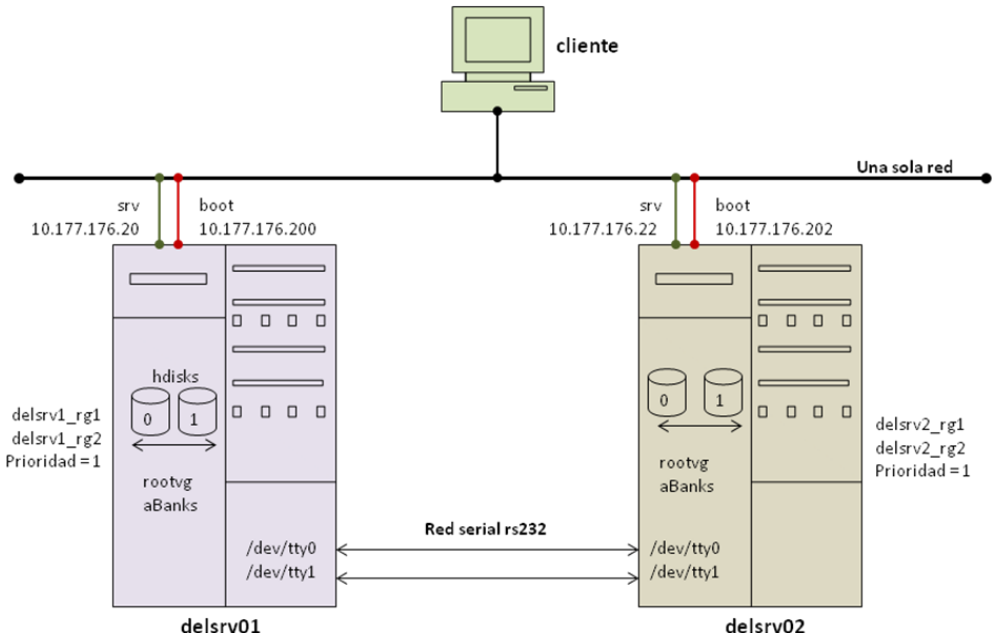


Figura 3.12 Definición de recursos para adaptadores y nodos de aplicaciones

En dos discos internos residía el sistema operativo AIX y la aplicación aBanks. Estos discos fueron configurados con espejamiento el mismo que era manejado por AIX. Al momento de falla de uno de los nodos, por ejemplo si llegase a fallar el servidor delsrv01, el delsrv02 debería ser capaz de absorber la cargar del delsrv01 + delsrv02, manejar las dos direcciones IP de servicios y los usuarios definidos en cada uno de estos, que deberían contar con el mismo UID para un correcto levantamiento de usuarios en el nuevo servidor. Este cambio de control una vez que se ha perdido el nodo delsrv01 se muestra a continuación. En éste se puede observar como la dirección de boot ha sido reemplazada por la dirección de servicio del nodo delsrv01. Un usuario conectado al servidor delsrv01 perdía la sesión de su aplicativo por unos pocos segundos mientras duraba el proceso de "*takeover*"

realizado por deldb02 reclamando las direcciones IP del nodo caído. Como el motor de Oracle y el aplicativo aBanks estaban copiados exactamente igual en el nuevo servidor, no hacía falta crear aplicativos o scripts para que los levanten. Al momento que delsrv01 reingrese al cluster, el levantará el sistema operativo y las redes con direcciones de *booteo* para no duplicar las direcciones IP. Este proceso de eliminación de un nodo del cluster también puede ejecutarse de una manera planificada por parte del administrador del Sistema, en casos de mantenimiento o fallas controladas.

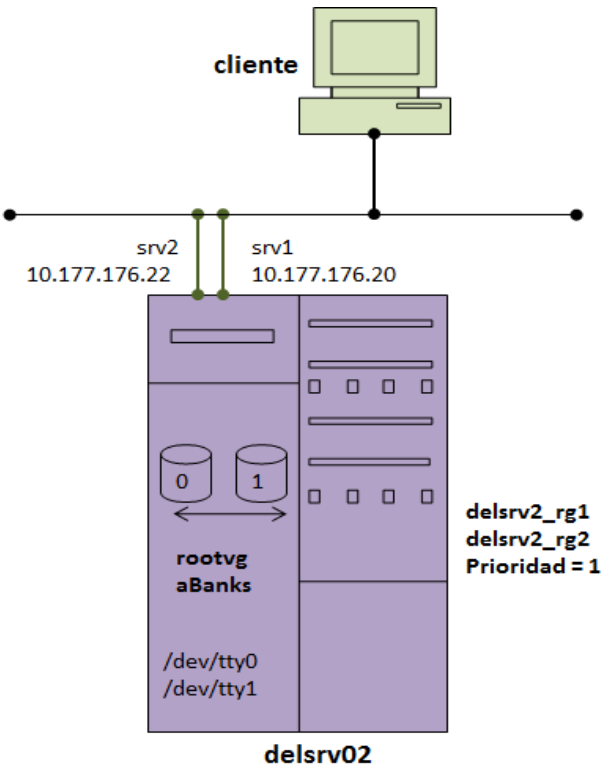


Figura 3.13 Nodo delsrv02 una vez que ha remplazado al caído delsrv01

Como se puede observar, los discos duros de sistema operativo están contenidos dentro del grupo de volúmenes “rootvg” el cual contiene dos discos con espejamiento (RAID 1) donde residen el sistema operativo y el aplicativo aBanks.

3.3. Preparación de la instalación de la base de datos Oracle 9iRAC

Los pasos previos a la instalación del Oracle RAC que se realizaron, incluyeron un análisis secuencial de cada una de las tareas ejecutadas por los Ingenieros de IBM, Oracle y DelBank. Estos pasos detallan la preparación previa la instalación del Oracle.

Operaciones		Realizado en nodo?		
		deldb01	deldb02	delap02
1	Chequeo de requerimientos de Hardware	Si	Si	Si
2	Chequeo de requerimientos de Redes	Si	Si	Si
3	Requerimientos de software	Si	Si	Si
4	Crear usuarios y grupos para Oracle en AIX	Si	Si	Si
5	Configurar Parámetros del Kernel y límites de AIX	Si	Si	Si
6	Identificar directorios para el software	Si	Si	Si
7	Crear un directorio base para el motor de Oracle	Si	Si	Si
8	Crear filesystems y directorios para la base de datos	Si	Si	Si
9	Crear filesystems para archivos	Si	Si	Si

	de recuperación			
10	**Configurar los discos para el almacenamiento de datos**	Si	Si	Si
11	**Sincronizar la hora del sistema en los nodos**	Si	Si	Si
12	**Detener todos los procesos Oracle**	Si	Si	Si
13	**Configurar e instalar Oracle**	Si	Si	Si

Tabla 3. Pasos previos a la instalación del Oracle RAC (tomado de [2])

Es claro entonces que Oracle necesitaba de un software especializado que le permita formar el cluster con varios nodos y en nuestro caso ese software era el GPFS, el mismo que permitía compartir filesystems utilizado servidores IBM RS/6000 o pSeries. A nivel de hardware, Oracle necesitaba los siguientes recursos:

- *Servidores*: Utilizamos tres servidores IBM pSeries: dos de ellos p630 y uno p615 los cuales permitían ser configurados en cluster y manejaban SMP, es decir varios procesadores internamente.
- *Almacenamiento*: Utilizamos un sistemas de almacenamiento que permitía ser conectado a dos o más servidores del cluster. En nuestro caso, utilizamos FastT600 conectados vía canal de fibra (FC).
- *Redes*: Obligatoriamente de alta velocidad; Para nuestro caso, utilizamos una red interna formada por adaptadores Gigabit Ethernet cuya transferencia está en el orden de los 1000Mbps o más. También tuvimos que configurar una red redundante de respaldo en caso de caída de una de ellas.

La versión de Oracle 9i RAC que corría sobre AIX 5.2 es la 9.2.0.2 la misma que estaba debidamente soportada en servidores pSeries 64bits y GPFS 2.1. Fue necesario la instalación de los siguientes parches de software: U488744, U488745, U488746, U488747. Para proceder a la instalación del motor de la base de datos, se debió crear un usuario "oracle", que pertenecía al grupo "dba". Se asignó el correspondiente UID=250 y GID=250 en el Filesystem o "home directory**" /oracle**. Los directorios y archivos para recuperación fueron creados dentro del mismo **/oracle**. Un compendio de los valores definidos previa la instalación, se muestra a continuación:

DESCRIPCIÓN	NOMBRE	ID/TAMAÑO
Dueño de la base datos	Oracle	250 GB
Grupo primario	Dba	250 GB
Grupo secundario oracle	Oinstall	251 GB
Home Directory	/oracle	9 GB

Tabla 4. Valores definidos para el ambiente de instalación de Oracle RAC

Nuestro diseño de la arquitectura Oracle RAC que debía quedar instalada y funcionando en banco DelBank, se muestra en la gráfica a continuación. En esta se observan los tres niveles de software bien diferenciados por colores: Oracle, GPFS y sistema operativo AIX; cabe indicar que RSCT, TCP/IP y el manejador de volúmenes lógicos (LVM) son parte del sistema operativo, razón por la cual se los muestra utilizando el mismo color verde. La configuración GPFS proveía los filesystems compartidos para Oracle y la información de dispositivos en caso de tener redundancia y necesitase protegerlos de alguna falla. La línea solida de color rojo representa la "red primaria" de interconexión que usaba Oracle para el intercambio de los bloques de datos, mientras que la línea punteada azul muestra la red secundaria; ambas proveen la alta disponibilidad requerida por Oracle para su interconexión interna. La línea fina color verde es la red GPFS. *En*

nuestro caso, GPFS fue implementada en RSCT es decir una tecnología de clusters escalable de la cual es parte de AIX y proveía la infraestructura de alta disponibilidad como la membrecía del cluster y la generación de eventos. Es más o menos una "goma" para parear nodos de un cluster (RPD).

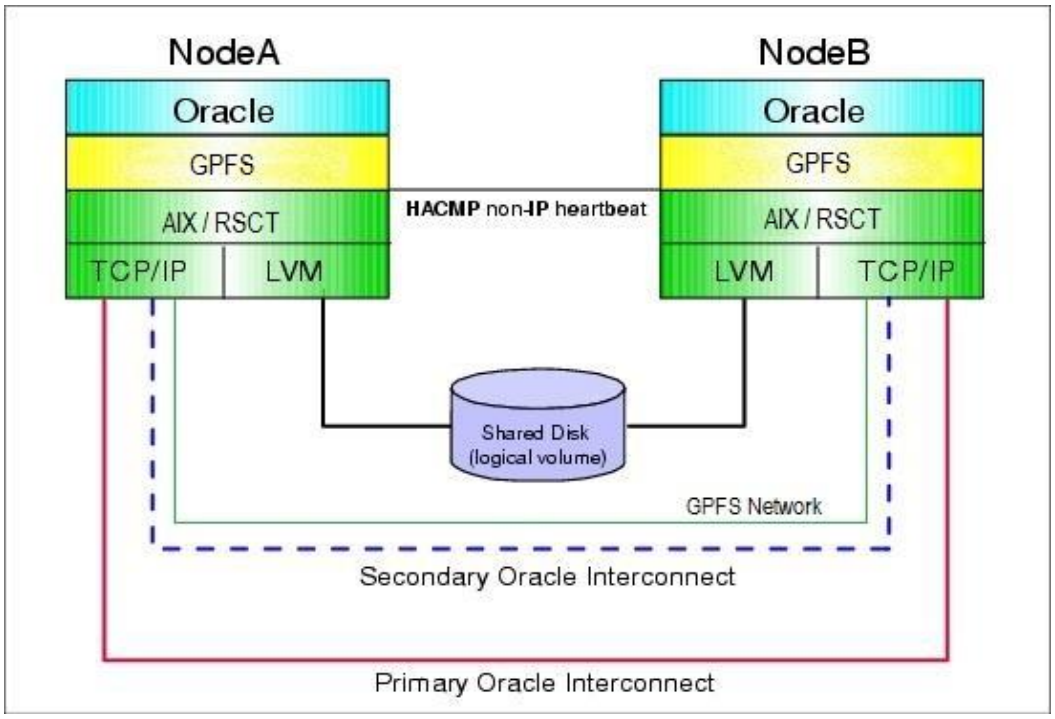


Figura 3.14 Los diferentes niveles de software en los nodos GPFS (tomado de [1])

El tamaño de bloque de transferencia de datos entre Oracle RAC y GPFS debía ser el mismo para mejorar la velocidad de transferencia entre las transacciones de entrada-salida entre Oracle y GPFS y se lo definió en 64K y poder mapear archivos grandes de hasta 10TB. **db_block_size = 64KB.**

CAPÍTULO 4.

4. DISEÑO DE LA SOLUCIÓN GPFS

Para poder realizar el diseño disponible y escalable de la solución GPFS, todas las personas implicadas en el proyecto debieron estar familiarizadas con las tecnologías de almacenamiento, funcionamiento de los nodos, funcionamiento del sistema redundante de discos independientes (RAID), Redes de área de almacenamiento (SAN) y almacenamiento conectado en red. Cada uno de los implicados comprendía claramente cómo utilizar estas tecnologías para lograr la máxima disponibilidad de los datos del cliente DelBank. En este capítulo se detallan el diseño de todas las áreas implicadas en el proyecto y su interrelación trabajando como un todo: Base de datos Oracle, sistema operativo, GPFS, servidores y hardware de almacenamiento.

4.1. Esquema general de sistema GPFS

Antes de proceder a instalar GPFS V2.1, se procedió a la respectiva verificación de hardware y software, tanto a nivel de aplicaciones, bases de datos y software de alta disponibilidad. Una vez que se procedió a la entrega y revisión formal de los servidores, el departamento técnico de hardware procedió a la instalación física y eléctrica de los

servidores dentro de un rack drawer tipo y modelo 7014-T42. Los dos servidores de base de datos y los dos de aplicaciones fueron configurados de una manera similar para permitir una mejor y fácil administración tanto a nivel de software como de hardware. En cambio, el quinto servidor pSeries 615, fue utilizado para lograr el quórum.

Servidores de Base de Datos pSeries 630	
Recursos	**Configuración**
Microprocesador	64 Bits POWER4+
Cantidad de procesadores	2 de 1.45Ghz. (Max 4 procesadores)
Memoria RAM	4GB (Max 32GB)
Memoria Cache (por procesador)	**L1**: 32KB **L2**: 1.5MB **L3**: 8MB
Slots PCI	20 de 64 bits
Fuentes de Poder	2 (Redundantes)
Ventilador	2 (Redundantes)
Discos internos	2x73.4GB (Total 146.8GB) de 15K RPM.
Adaptador integrado 10/100 Ethernet	2
Adaptador Gráfico	GXT135P
Controladoras ULTRA3 SCSI	2
Adaptador Gigabit FC PCI-X	2
Adaptador 10/100/1000 Ethernet	2
Unidad interna de respaldo	Tapes de 20 y 40GB
DVD ROM DRW	16X/48X
Puertos Seriales	3

Tabla 5. Configuraciones de los servidores pSeries 630 de base de datos

Las configuraciones de Hardware fueron revisadas detalladamente antes de proceder a la instalación del GPFS y se detallan en la tabla anterior.

Ambos servidores de base de datos eran similares, con la misma configuración de Hardware y de software para que la carga de trabajo sea repartida proporcionalmente. El servidor pSeries 615 tenía la capacidad de absorber la carga de trabajo en caso de la caída de uno de los servidores pSeries 630 de base de datos, pero su capacidad de procesamiento se estimaba sea 40% menor, pero se preveía que en los próximos 6 meses la versión 8.3 del Storage Manager pueda ser actualizada sin costo a la versión 8.4, de éste modo poder activar el parámetro llamado **Persistent Reservation** y poder obtener un quórum con solamente dos servidores en vez de tres, limitante que en ese entonces existía.

El servidor de base de datos pSeries 615 fue configurado con un adaptador *Gigabit FC* PCI-X como los otros 630's, para así poder conectarlo a la red SAN y por consiguiente a los discos dentro del arreglo FastT6000. El detalle de la configuración de este servidor se detalla a continuación:

Servidor de Base de Datos pSeries 615	
Características	**Configuración**
Microprocesador	64 Bits POWER4+
Cantidad de procesadores	2 de 1.2Ghz. (Max 2 procesadores)
Memoria RAM	4GB (Max 16GB)
Memoria Cache (por procesador)	**L1**: 32KB **L2**: 1.5MB **L3**: 8MB
Slots PCI	20 de 64 bits
Fuentes de Poder	2 (Redundantes)

Ventilador	2 (Redundantes)
Discos internos	2x73.4GB (Total 146.8GB).
Almacenamiento interno máximo	1.1TB
Adaptador Gráfico	GXT135P
Controladoras ULTRA3 SCSI	1 (Dual)
Adaptador integrado 10/100/1000 Ethernet	2
Adaptador integrado 10/100 Ethernet	2
DVD ROM DRW	IDE Slimline
Puertos Seriales	3
Adaptador Gigabit FC PCI-X	2

Tabla 6. Configuraciones de los servidores pSeries 615

En vista a que GPFS V2.1 controlaba el acceso a los datos en forma de filesystems y no en forma “raw”, éste encargaría de manejar el lock de registros internos para otorgar un acceso concurrente a nivel de datos; por esta razón se procedió a la revisión correspondiente a nivel de almacenamiento externo, donde va a residir la base de datos. El servidor de almacenamiento adquirido por DelBank era del tipo y modelo: 1722-60U FastT600 Storage Server, el mismo que contenía 14 discos duros de 73.4GB, cada uno de 10000 rpm y 2GB FC (Fiber Channel). Para la interconexión entre estos discos y los servidores de base de datos se procedió a conectar las dos tarjetas Gigabit FC PCI-X en cada uno de los servidores a los dos TotalStorage SAN Switch de 8 puertos. La idea de configurar dos SAN Switch era tener alta disponibilidad entre ellos.

A nivel de software, se analizó detalladamente el ambiente sobre el cual la aplicación Abanks va a correr y su interoperabilidad con la base de datos, sistema operativo y software de alta disponibilidad. Se hizo una revisión detallada de cada uno de los pre-requisitos de software necesarios en cada uno de los siguientes ambientes:

- A nivel de interoperabilidad entre el software, GPFS V2.1 con AIX 5.2 se debió instalar los siguientes parches (APARs) de sistema operativo IY36782, IY37744, IY37746, IY36626.
- A nivel del software de base de datos Oracle para el ambiente en DelBank, se instaló Oracle9*i,* Release 2 (9.2.0.2) la misma que era soportada para sistemas basados en AIX 5.2 y equipos de 64bits y GPFS 2.1. Fue necesario la instalación de los siguientes parches de software: U488744, U488745, U488746, U488747.
- A nivel de HACMP 4.5 se debía instalar el parche U487607
- A nivel del software Storage Manager que administra el subsistema de discos y que corre bajo un cliente Windows, se instaló la versión 8.3 y se procedió a configurar RAID 1 o *mirroring*, obteniendo una capacidad real de 513.8 GB para almacenamiento de datos, aunque GPFS V2.1 utiliza un sistema de acceso a archivos tipo *stripe* (o RAID 0).
- El tamaño de bloque de GPFS se configuró en 64K, de igual tamaño que el de Oracle (utilizando la variable **db_block_size**) mejorando de esta manera las transacciones de entrada-salida entre Oracle y GPFS. Con el tamaño de bloque de 64K, se aseguraba un tamaño máximo de archivo de hasta 10TB.

En vista a que GPFS requería de al menos una red dedicada sobre la cual las conexiones tipo socket iban a establecerse, se procedió a la correspondiente configuración de hardware, verificando los requerimientos mínimos a cumplirse con respecto a tarjetas de red y switch de conexión. En esta red se establecían los sockets de conexión para los procesos o demonios GPFS. ***Es importante aclarar que GPFS 2.1 no necesitaba HACMP, en vista a que esta versión ya contaba con la capacidad de reintegrar recursos redundantes cuando un principal falla, tal como lo hace HACMP.***

La red GPFS fue utilizada por los procesos permanentes para intercomunicar los servidores y conocer el estado de los archivos de un filesystem compartido, *teniendo en cuenta que los datos reales del sistema de archivos viajaban a través de los adaptadores Gigabit FC directamente entre los nodos y los discos*.

Una segunda red del IP pudo ser configurada para mejorar la disponibilidad del servicio de red IP dedicada de GPFS, pero para banco DelBank se definió solamente una red y no dos. Se requirió que en cada servidor se utilice los adaptadores Ethernet de 10/100 Mbps; además definir en el archivo ***/*.rhosts** los nombres de host de cada servidor asociado a la red GPFS para de esta manera tener acceso al usuario administrador **root** remotamente. La nomenclatura a utilizar era la siguiente: Ngpfs0X, donde N=nodo X=1,2,3 dependiendo si es el nodo 1 (pSeries 630) 2 (pSeries 630) o 3 (pSeries 616) de bases de datos.

Figura 4.1 Detalle de la red interna o dedicada para GPFS

4.2. Definición de arreglo de discos con su correspondiente protección

El software “Storage Manager” versión 8.3, permitía la administración y configuración centralizada del sistema de almacenamiento FastT600 modelo DS4300 tanto a nivel local como vía red LAN, utilizando una interface basada en Java con interface Web. Esta aplicación también ha sido diseñada para mejorar y realizar cambios de las configuraciones así como añadir nuevos volúmenes, definir mapeos de discos, manejar interfaces de rutinas y añadir dinámicamente nuevos arreglos de discos sin interrumpir el acceso a los datos de usuario. Finalmente, con el Storage Manager era posible administrar la recuperación automática por falla de discos y analizar el rendimiento de los mismos. Un ejemplo de la interface gráfica del Storage Manager V8.3 corriendo bajo un servidor Windows se muestra a continuación:

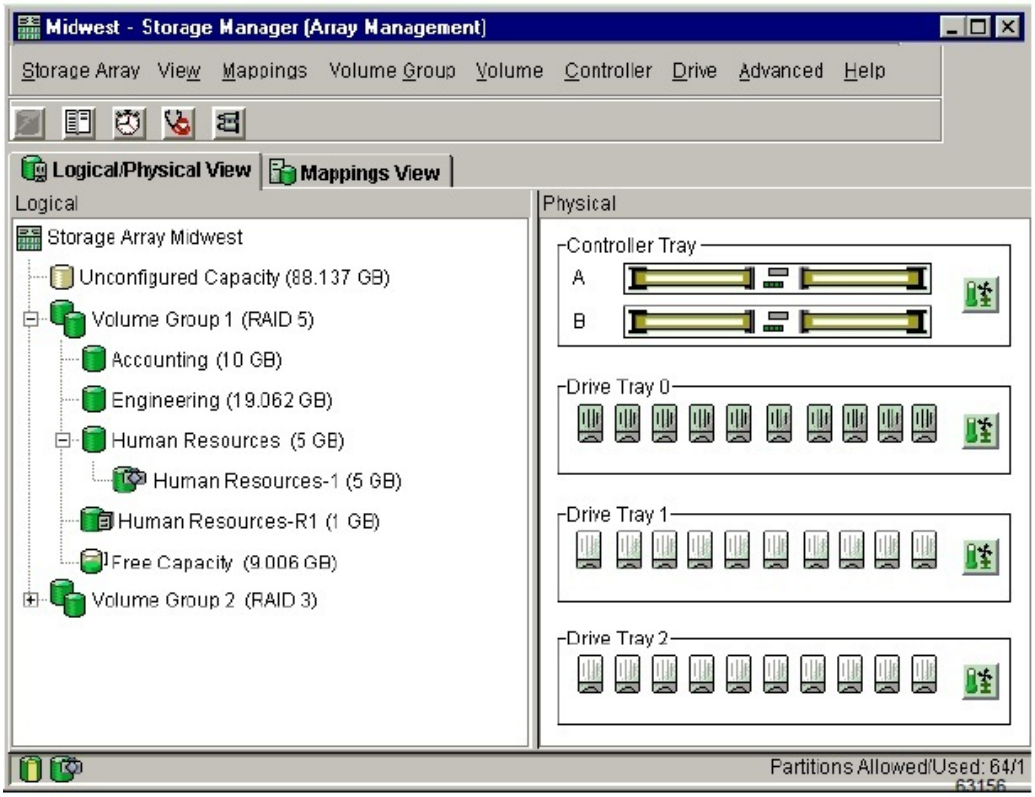


Figura 4.2 Ejemplo de interface gráfica del Storage Manager 8.3 (tomado de [9])

En base a las bondades anteriormente descritas, la definición del arreglo de discos debió ser configurada para sacar el mejor provecho al servidor DS4300 y adaptarse a los requerimientos del cliente. Bando DelBank. Decidió utilizar la opción de RAID 1 o espejamiento; aunque era la más costosa de todas por que utiliza solo el 50% de la real capacidad de todos los discos, fue la elegida por el cliente por ofrecer mejor protección contra fallas de disco, mejor performance y mejor protección de datos. Una ventaja adicional de tener el espejamiento realizado por el subsistema 1722-60U, era que se liberaba de esta carga al sistema operativo AIX. En el gráfico que se muestra a continuación se observa en detalle cómo es la configuración de los discos en RAID 1.

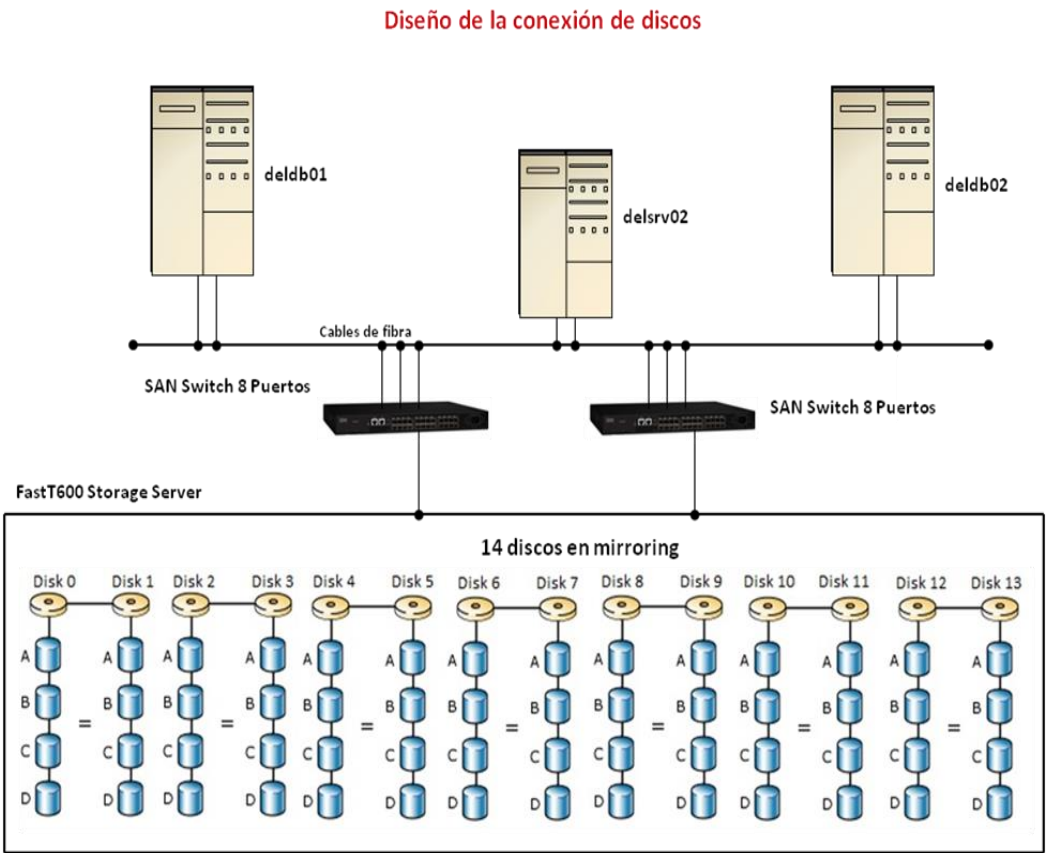


Figura 4.3 Configuración de discos del FastT600 en RAID 1

Como se puede apreciar, se ha definido cuatro diferentes archivos A, B, C y D que están siendo escritos en varios discos; cada escritura en un disco, implica que habrá una copia en el disco imagen o viceversa, pues cualquier disco puede ser escrito primero. Al momento de una caída o falla de un disco, el servidor DS4300 trata de corregir el error o de lo contrario envía una señal de alerta para que se proceda a la verificación y al reemplazo en línea correspondiente.

Se puede apreciar a la unidad de almacenamiento 1722-60U o DS4300 con los 14 discos se conecta a los dos switch de fibra de 8 puertos por medio de cables desde sus controladores. Desde los puertos de cada switch, otros cables de fibra conectan estos dispositivos a los adaptadores *Gigabit* FC en cada uno de los servidores pSeries 630 y 615 formando una doble conexión a los servidores, la misma que protege de una posible caída de uno de los switch FC formando un cluster a nivel de discos y servidores.

En vista a que la unidad de almacenamiento 1722-60U soporta IP sobre los adaptadores de fibra, se asignaron direcciones IP's para su administración desde un computador personal conectado a la misma red.

Finalmente, los tres servidores de base de datos deldb01, deldb02 y delap02 tenían dos adaptadores FC, un cable de fibra conectaba cada uno de los adaptadores al switch FC ocupando seis de los ocho puertos de éste, tal cual se muestra en el gráfico.

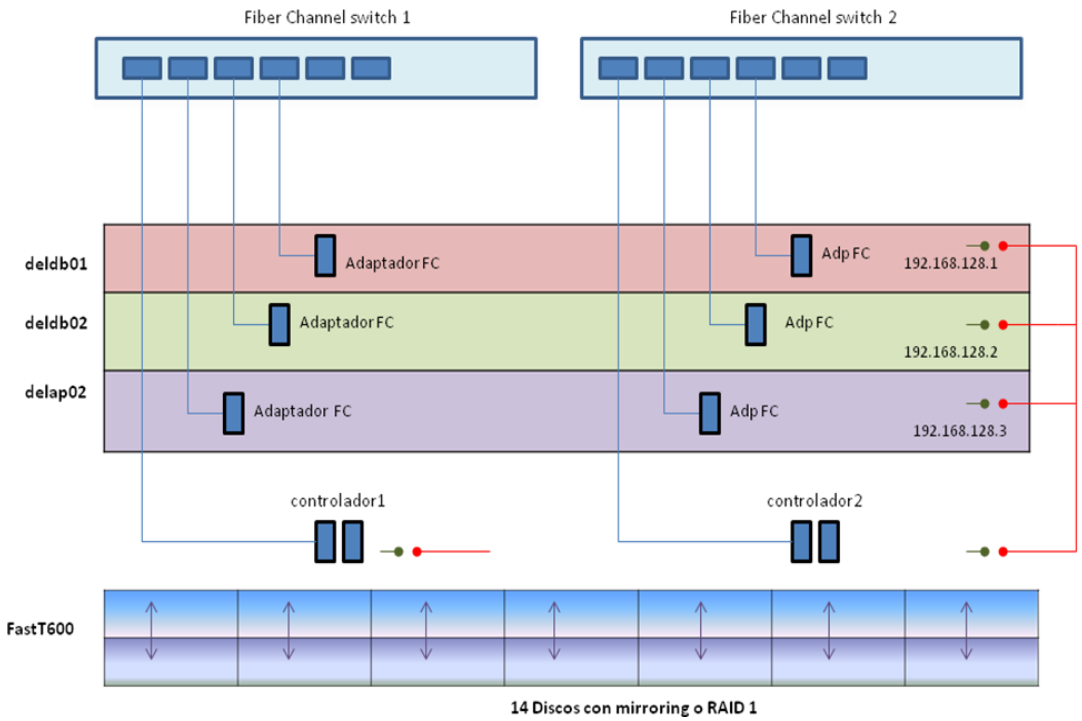


Figura 4.4 Conexiones con adaptadores FC en un nodo (tomado de [9])

Otro cable de fibra enlazaba cada uno de los switch FC a un controlador del arreglo de discos FastT600 (donde internamente residen los 14 discos en RAID 1). Con este cableado se garantizaba:

- Duplicidad y por ende protección mutua de adaptadores FC en los servidores
- Duplicidad y protección mutua de switch FCs.
- Mejor performance en la creación de dos caminos para acceso de datos.
- Utilización de dos vías (por cada adaptador) para acceso a la información.

Con este acceso vía lazo (*loop*): Tarjeta1 FC del ServidorN **->** Switch FC **->** Arreglo FastT600 **->** Tarjeta2 FC del ServidorN se garantizó un acceso dual o de dos vías y por ende un rendimiento mejorado al doble.

4.3. Definición de redes internas y de usuarios

En el punto 3.1.2, se hizo una definición detallada de las redes internas y externas. Luego de esto se procedió a la asignación de direcciones a cada uno de los adaptadores de red, tanto internas como externas y diferenciando los servidores de base de datos con los de aplicaciones. Los dos servidores pSeries 615 de aplicaciones utilizaban HACMP para proteger posibles fallas de adaptadores Ethernet, fallas de procesador, fuentes de poder o falla de los mismos servidores. No estaban conectados a las redes GPFS y Oracle RAC, pues únicamente ejecutaban las aplicaciones de aBanks, solicitando información y requerimientos a los servidores de base de datos vía red externa, es decir la misma red utilizada por los usuarios del sistema. Las redes privadas en los servidores de aplicaciones eran las definidas por HACMP e incluían las redes seriales rs232 y eran las encargadas de identificar como última instancia si un servidor estaba caído o no; en general, estatus actual. La red de aplicaciones incluía los dos puertos seriales por servidor pSeries y dos adaptadores ethernet, el uno para servicio a usuarios y el segundo para adquirir una dirección temporal al momento de reiniciar.

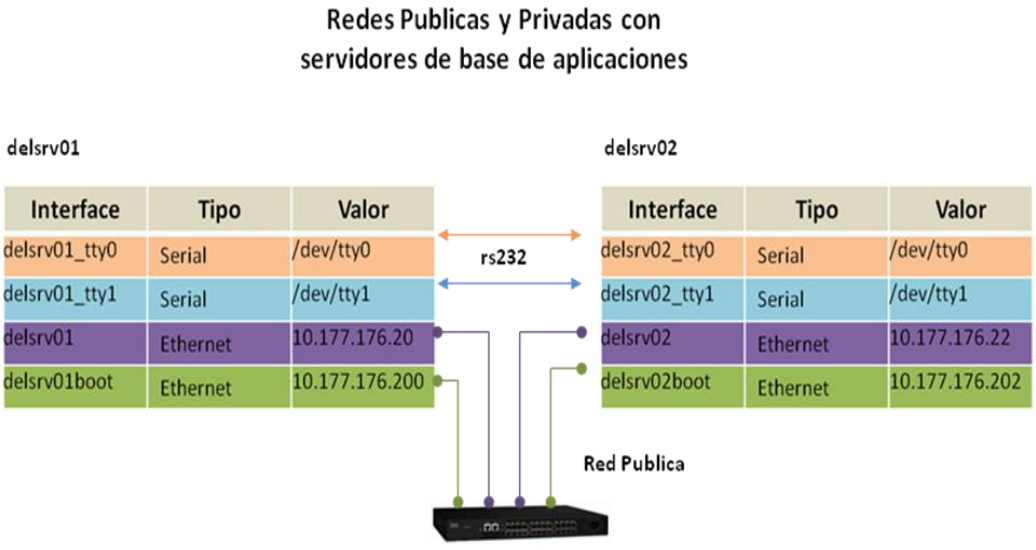


Figura 4.5 Redes públicas y privadas en servidores de base de datos

En los servidores de base de datos, el diseño de los clusters incluía los dos servidores pSeries 630 y un pSeries 615. Todos los servidores estaban conectados por fibra al switch y al almacenamiento de disco fastT600. Los servidores de base de datos pSeries 630 y el pSeries 615 conectados a la red GPFS y Oracle RAC, tenían 4 tarjetas de red Ethernet: 2 de 10/100/1000 y 2 de 10/100; las dos de 10/100/1000 se utilizarían para la red primaria y secundaria de Oracle (punto a punto); una de 10/100 para GPFS, que era un requisito del producto, y la restante para conectarse a la red del banco para el acceso de usuarios.

La red privada serial rs232 se conectó directamente a los adaptadores vía cable serial. Se definieron los nombres de los 3 servidores de base de datos: **deldb01**, **deldb02** y **delap02**, cada uno provisto con 6 adaptadores (4 ethernet y 2 seriales) formando las siguientes redes:

- 1 red pública para servicio a usuarios con tarjeta Ethernet 10/100 Mbps.
- 1 red privada GPFS con tarjeta Ethernet 10/100 Mbps.
- 2 redes privadas de Oracle RAC con tarjetas Ethernet 10/100/1000 Mbps.
- 2 redes seriales rs232.

Adaptador	Tipo	Red	Tipo	Uso	Nodo	Dirección
delap02_tty1_01	Servicio	net_rs232_02	rs232	serial	delap02	/dev/tty1
delap02_tty2_01	Servicio	net_rs232_03	rs232	serial	delap02	/dev/tty2
delap02	Boot	Public_network	ether	Public	delap02	10.177.176.22
orac_30	Servicio	rac92_network1	ether	Privada	delap02	182.16.16.22
orac_31	Servicio	rac92_network2	ether	Privada	delap02	172.16.16.22

ngpfs03	Servicio	red_gpfs01	ether	Privada	delap02	192.16.16.3
deldb01_tty0_01	Servicio	net_rs232_01	rs232	Serial	deldb01	/dev/tty0
deldb01_tty1_01	Servicio	net_rs232_02	rs232	Serial	deldb01	/dev/tty1
deldb01	Boot	Public_network	ether	Public	deldb01	10.177.176.24
orac_10	Servicio	rac92_network1	ether	Privada	deldb01	182.16.16.24
orac_11	Servicio	rac92_network2	ether	Privada	deldb01	172.16.16.24
ngpfs01	Servicio	red_gpfs01	ether	Privada	deldb01	192.16.16.1
deldb02_tty0_01	Servicio	net_rs232_01	rs232	Serial	deldb02	/dev/tty0
deldb02_tty2_01	Servicio	net_rs232_03	rs232	Serial	deldb02	/dev/tty2
deldb02	Boot	Public_network	ether	Public	deldb02	10.177.176.26
orac_20	Servicio	rac92_network1	ether	Privada	deldb02	182.16.16.26
orac_21	Servicio	rac92_network2	ether	Privada	deldb02	172.16.16.26
ngpfs02	Servicio	red_gpfs01	ether	Privada	deldb02	192.16.16.2

Tabla 7. Detalle de las redes en los servidores GPFS

En la tabla anterior se puede observar a cada uno de los nodos con su respectivo nombre (deldb01, deldb02, delap02) y diferenciados por un color especifico. En la columna “adaptador” se puede observar el nombre que el sistema operativo asigna a un dispositivo serial o el nombre del hostname y que hace referencia a un adaptador Ethernet definido en el archivo ***/etc/hosts***. La columna “tipo” indica si el adaptador está diseñado para trabajar dando el servicio al cual ha sido encomendado o si trabaja como un dispositivo de “boot” donde adquiere una dirección de “booteo” temporal antes de ingresar, verificando si su dirección de servicio no está activa en la red y si es así, adquiere la mencionada IP, caso contrario se mantiene con la dirección de boot. Esto se

hizo para evitar duplicidad de direcciones en la red. La columna “red” contiene los nombres de las diferentes redes creadas para ser utilizadas en la configuración de GPFS, Oracle RAC y base de datos.

Como se puede observar, en ningún caso se utilizó doble protección a nivel de hubs o de switches, únicamente con la red privada de Oracle 9iRAC se utilizaron dobles tarjetas de red de 10/100/1000 en caso de contingencia por falla de una de estas.

Figura 4. 6 Detalle de las conexiones de red en los servidores de bases de datos

4.4. Interconexión entre GPFS y Oracle 9iRAC

Con la utilización de GPFS se permitió a la base de datos Oracle 9iRAC distribuirse en filesystems concurrentes, dejando en el pasado la obligatoriedad de crear la base de datos en almacenamiento crudo (RAW), cuya administración era demasiado complicada y desperdiciaba espacio en disco. GPFS hacía un mapeo de todo el almacenamiento como un espacio dividido en pequeños bloques (*striping*) mejorando así el acceso a información de una manera considerable. La interconexión de Oracle y GPFS implicaba conocer algunos componentes de Oracle RAC, su interconexión con las redes internas y la definición de un conjunto de parámetros para que la interacción entre estos dos niveles de software sea el óptimo.

Las peticiones a la base de datos eran generadas por la aplicación; normalmente desde un servidor de aplicaciones y el Oracle RAC en su conjunto era el encargado de direccionar las peticiones al servidor que estaba en funcionamiento. El diagrama a continuación contempla cómo funcionaba realmente el Oracle RAC:

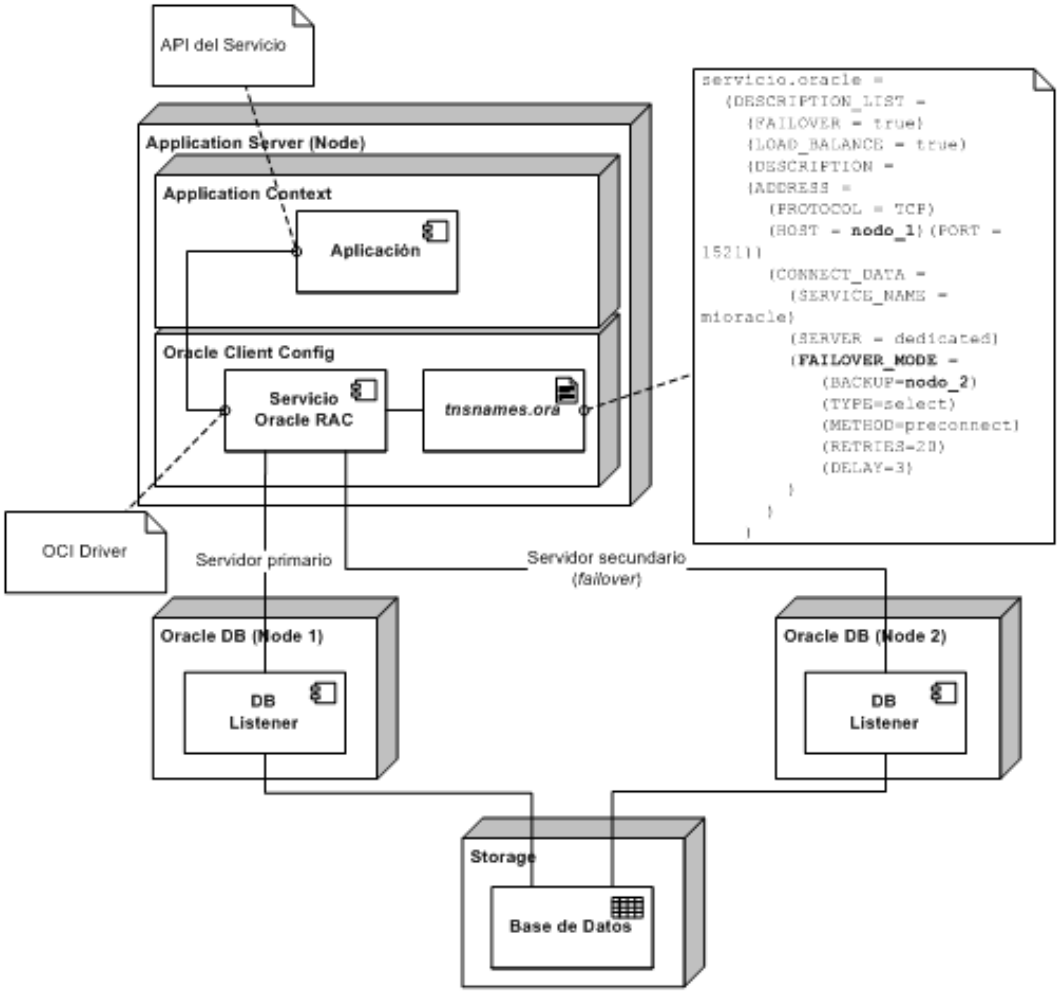


Figura 4.7 Componentes de software del Oracle RAC (tomado de [2])

En realidad, el cliente o servicio de Oracle estaba configurado para tener como conexión primaria el Nodo 1; si no era posible ejecutar la petición enviada a dicho servidor, el servicio se encargaba de realizar un re-direccionamiento de la misma hacía el servidor de respaldo (en este caso el Nodo 2). Adicionalmente, es necesario mencionar que lo que estaba corriendo en los Nodos 1 y 2 es el **listener** del motor de base de datos. El listener "escucha" las peticiones de conexión y enruta las conexiones a un proceso servidor. Cuando se realiza una solicitud a un listener de la base de datos, el listener le envía un mensaje de regreso al proceso cliente con una dirección de un proceso de

servidor. El proceso de cliente procede entonces a comunicarse con la base de datos utilizando el proceso de servidor, mientras que el listener continúa a la escucha de nuevas peticiones de conexión de otros procesos de cliente. Esta comunicación entre clientes y servidores por medio del listener, se la logra por medio de una capa de software de red denominada **OracleNet**.

Para conectar una base de datos a través de la red, un cliente utiliza un *descriptor de conexión*; un descriptor de conexión contiene detalles de la red relacionados con el servicio de base de datos y pasa esta información al listener. Los detalles de la red en el descriptor de conexión contienen la ubicación del listener (dirección de protocolo) y el nombre de base de datos global (nombre del servicio) de la base de datos. El nombre de base de datos global es generalmente una combinación del nombre de base de datos y la base de datos de dominio. Por ejemplo, con un nombre de base de datos de PROD y un dominio de xyz.com, el nombre de base de datos global sería PROD.xyz.com. Un descriptor de conexión es una cadena de caracteres (**string**) que por lo general incluye un nombre de usuario y una contraseña, un nombre de base de datos, con o sin detalles de la especificación del protocolo. Un descriptor de conexión de Oracle se muestra a continuación:

```
CONNECT <username>/<password>@(description=(protocol=tcp)
(host=<hostname>) (port=1521))
(connect_data=(service_name=PROD.xyz.com)))
```

Figura 4.8 Descriptor de conexión de Oracle

Oracle RAC trabaja en ambientes donde existen múltiples requerimientos de clientes y múltiples procesos despachadores por lo tanto, las conexiones de base de datos pueden ser compartidas entre diferentes usuarios compartiendo múltiples conexiones de

servidor, con esto permitía la creación de múltiples instancias del motor de base de datos de forma independiente, compartiendo un mismo almacenamiento.

Una vez que un nodo del Oracle RAC se desconectaba (ya sea por falla de hardware, red o sobredemanda de recursos), todas las transacciones debían ser re-direccionadas automáticamente al nodo de respaldo. El punto de dicho esquema consistía en no perder las transacciones que se encontraban abiertas en el momento de la desconexión. Cabe indicar que la versión 9i de Oracle sólo realizaba una recuperación (**failover)** transparente para consultas (**queries**) de tipo SELECT, mientras que para todos los demás tipos de operaciones marcaban un error automáticamente:

- Queries transaccionales o de manipulación de datos: INSERT, UPDATE y DELETE.
- Operaciones de manejo de sesión: ALTER SESSION y SQL*Plus.
- Objetos temporales (aquellos que utilizan el espacio de trabajo TEMP).
- Estados obtenidos durante la ejecución.

Era claro entonces que Oracle 9iRAC no garantizaba por sí mismo un failover transparente, pues sólo garantizaba la disponibilidad de la base de datos. Para lograrlo, necesitaba de un software adicional que en nuestro caso era el GPFS, con lo cual Oracle 9iRAC se convertía en un cluster de discos compartidos, con lo cual se permitía múltiples instancias que podían ser ejecutadas contra la misma base de datos. Una instancia en Oracle consistía de un Area Global de Sistema (SGA) y de procesos Oracle ejecutados en *background* como demonios (es decir que siempre estaban ejecutándose). Un SGA es una memoria compartida que contiene datos y controla la información para una instancia de Oracle. Oracle asigna el SGA cuando una instancia inicia y la desasigna cuando la instancia termina.

Cada instancia de Oracle RAC tiene su propia SGA y su tamaño es determinado por las siguientes variables del sistema: DB_CACHE_SIZE, LOG_BUFFER, SHARED_POOL_SIZE y LARGE_POOL_SIZE. En vista a que Oracle RAC soporta el uso de SGA dinámico, la cantidad de memoria virtual que utiliza Oracle puede ser configurada en line sin tener que bajar las instancias. Esto se logra iniciando una instancia con una cantidad de memoria pequeña, la cual puede crecer a medida que se incrementa el tamaño de los componentes del SGA hasta un máximo de SGA_MAX_SIZE.

Un típico nodo del cluster estaba definido como una colección de procesos, memoria RAM y discos donde se almacenan datos y códigos de una instancia; en cambio los discos compartidos almacenaban el sistema de tablas (**tablespace**), los *redo logs* en línea, respaldos temporales de tablespaces, el control de archivos de Oracle y de la base de datos. Con todo esto, el usuario podía accesar clientes y aplicaciones en cualquier instancia de la base de datos haciendo “creer” que la aplicación corre en una sola imagen del sistema. El ejecutable de Oracle podía residir en los discos internos con el sistema operativo AIX o también en los discos compartidos, pero muy aparte de donde se los instale, los procesos Oracle eran los mismos, los cuales se categorizan en dos grupos importantes:

- Procesos de Usuario que corren el código de una alguna aplicación.
- Procesos de Oracle que corren el código del servidor Oracle. Estos incluyen procesos del servidor y procesos background.

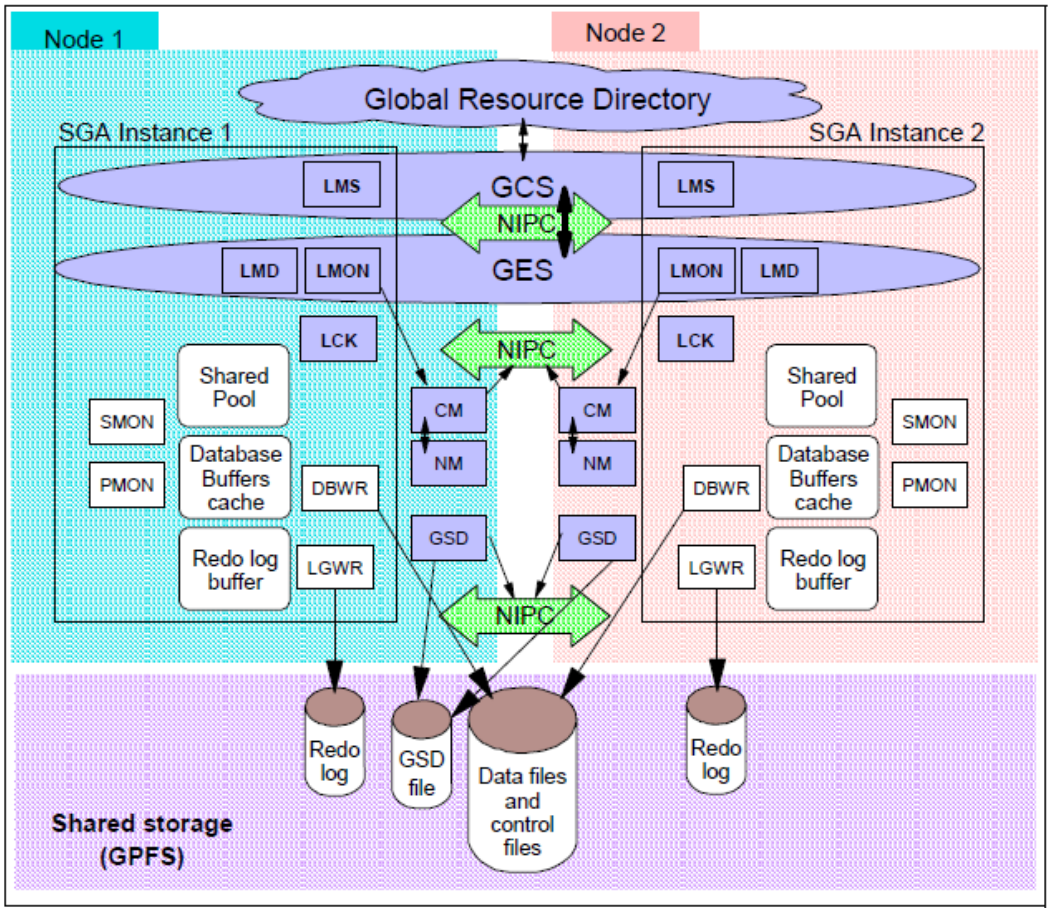


Figura 4.9 Interacción entre los procesos Oracle 9iRAC (tomado de [2])

Este gráfico muestra los siguientes procesos:

PMON - Process Monitor

SMON - System Monitor

DBRW - Database Writer

LOGW - Log Writer

LMON - Lock Monitor Process (Global Enqueue Service Monitor).

GES - Global Enqueue Service Daemon.

LMS - Lock Manager Server (Global Cache Service Processes).

GCS - Global Cache Service.

LMD - Lock Manager Daemon (Global Enqueue Service Daemon)

LCKn - Lock Process

NPIC - Network Inter-Process Communication

CM - Cluster Manager

NM - Node Monitor

GSD - Global Service Daemon

Además de las variables nombradas anteriormente, para una perfecta interconexión de Oracle con GPFS, se definieron otras variables, muchas de los cuales asignadas por el personal de base de datos que colaboró con la instalación.

DB_BLOCK_SIZE

Especifica en tamaño en bytes del bloque de la base de datos Oracle. El valor de esta variable tomaba efecto al momento de crear la base de datos la misma que variaba de: 2 KB, 4 KB, 8 KB, 16 KB, and 32 KB. Para el RAC implementado bajo GPFS el soporte Oracle recomendó 16KB, el mismo que el tamaño de bloque de GPFS.

ORACLE_SID

Identifica una instancia dentro de un archive cuyo nombre es **init<SID>.ora**, el cual identifica los parámetros de inicialización para esa instancia. SID es un número identificador asignado por Oracle.

PARÁMETROS DE INICIALIZACIÓN

Para la inicialización de la base de datos se utilizó un archivo ASCII que contenía todos los parámetros necesarios para levantar la base de datos. Aunque el nombre de este archivo podía variar, nosotros utilizamos el nombre estándar de asignado por Oracle: **init<SID>.ora**, el mismo que identificaba el parámetro de inicialización con una instancia especifica. SID es el identificador de la instancia y puede ser asignado en la variable **ORACLE_SID.**

4.5. Diseño de la disponibilidad eléctrica y comunicaciones

Cada uno de los servidores de base de datos IBM pSeries 630 y de aplicaciones IBM pSeries 615 fueron configurados con fuentes de poder y ventiladores redundantes, tal cual se indicaba anteriormente en las tablas 5 y 6. Cada fuente de poder estaba conectada a una red de poder diferente con su respectivo CPU; y cada red de poder estaba conectada a su vez a un circuito AC diferente tal cual se muestra en la figura a continuación:

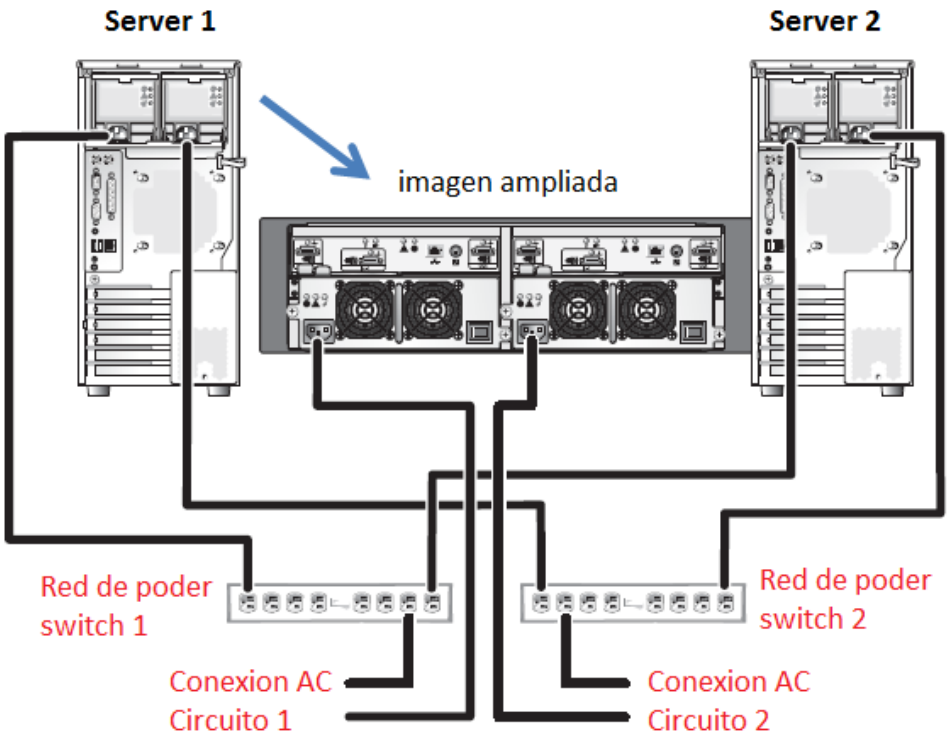


Figura 4.10 Detalle de las conexiones eléctricas redundantes (tomado de [1])

Con esta configuración cruzada e independiente, se protegía a los servidores de los siguientes posibles puntos de falla: Daños de una fuente de poder, daños en la

ventilación de cada fuente, falla en uno de los CPU's y falla eléctrica en uno de los circuitos de entrada AC

La configuración eléctrica estuvo a cargo del personal técnico de hardware de IBM del Ecuador y la revisión e instalación de la misma fue un paso previo a la instalación física de los servidores. A nivel de comunicaciones de red y como se indicó en el capítulo 3, banco DelBank no definió ninguna red de usuarios redundante, quedando bajo su responsabilidad la instalación de un nuevo equipo de comunicaciones externo (hub o routers) en caso de que existiera alguna falla. Este punto de falla abierto se puede observar en la figura 3.11.

4.6. Diseño de pruebas

En este capítulo se presenta un diseño de las pruebas que se realizaron en Banco Delbank con el fin de verificar el correcto funcionamiento de la alta disponibilidad que ofrecía la solución propuesta, tomando en cuenta la conmutación y la recuperación luego de la falla de alguno de los componentes que formaban el cluster GPFS.

4.6.1. Consideraciones generales del ambiente de pruebas a verificar

Consideramos el hecho de que los componentes del Oracle9i RAC como el listener, instancias de base de datos e interfaz de cliente dependían directamente de la disponibilidad de la base de datos; también era claro que estos componentes de

Oracle estaban directamente relacionados con los componentes de hardware, redes locales, sistema operativo AIX y el GPFS.

Definición de Instancias de base de datos:

Para las diferentes pruebas de simulación de fallas, creamos tres diferentes instancias de base de datos, una por cada nodo (ngpfs01, ngpfs02, ngpfs03) para verificar como una sentencia SQL cambiaba de instancia para terminar la operación indicada. La variable **ORACLE_SID** almacenaba el nombre de una instancia en un servidor y para nuestros casos de prueba, utilizamos el nombre **RAC** como prefijo para el nombre de una instancia, tal cual se muestra en la siguiente tabla:

Servidor	Nodo	Instancia	SID
Nodo 1	ngpfs01	1	rac1
Nodo 2	ngpfs02	2	rac2
Nodo 3	ngpfs03	3	rac3

Tabla 8. Nombre de las instancias definidas en los nodos de base de datos

Para comprobar el nombre de la instancia de base de datos en un servidor, nos conectamos como usuario oracle y ejecutamos el siguiente comando:

```
{ngpfs01:oracle}/oracle/home-> echo $ORACLE_SID
rac1
```

Figura 4.11 Comprobación de la instancia en un servidor

Tipos de pruebas:

Para las pruebas de simulación con Oracle, se utilizó dos tipos diferentes de SQLPLUS lanzados desde sesiones clientes:

- Sesión Inactiva – Se tomó nota de fecha, hora y el nombre de la instancia antes y después de la falla. No había consultas activas pero si clientes conectados a una instancia en el momento de la falla. Queríamos verificar si la sesión podía volver a conectarse a otra instancia y si era posible realizar consultas desde la misma sesión después de la falla.
- Sesión Activa – Se tomó nota de la fecha y hora y el nombre de la instancia antes y después de la falla. Después de la primera marca de tiempo emitimos varias instrucciones **SELECT** y durante este intervalo iniciamos la condición de falla. Queríamos verificar que la consulta que se estaba ejecutando no se afectaba por la falla y comprobar si que ésta consulta continuaba procesándose en otra instancia como resultado de una recuperación (failover).

Configuración del cliente:

Para todas las pruebas de fallas, se implementó una configuración especial del archivo **tnsnames.ora** en los clientes, para permitir la recuperación, mientras los clientes se conectaban a una base de datos denominada **rac**, tal como se muestra a continuación:

```
#--- 1st entry: load balancing, connect time failover, TAF
method=basic
RAC =
  (DESCRIPTION =
    (enable=broken)
    (LOAD_BALANCE=ON)
    (FAILOVER=ON)
      (ADDRESS = (PROTOCOL = TCP)(HOST = node1)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node2)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node3)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node4)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = RAC)

(failover_mode=(type=select)(method=basic)(retries=20)(delay=5))
    )
  )
# --- 2nd entry: no load balancing, but connect time failover and TAF
method=basic
```

```
RAC1F =
  (DESCRIPTION =
    (enable=broken)
    (LOAD_BALANCE=NO)
    (FAILOVER=ON)
      (ADDRESS = (PROTOCOL = TCP)(HOST = node1)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node2)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node3)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node4)(PORT = 1521))
    (CONNECT_DATA =
       (SERVICE_NAME = RAC)

(failover_mode=(type=select)(method=basic)(retries=20)(delay=5))
    )
  )
RAC2F =
  (DESCRIPTION =
    (enable=broken)
    (LOAD_BALANCE=NO)
    (FAILOVER=ON)
      (ADDRESS = (PROTOCOL = TCP)(HOST = node2)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node3)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node4)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node1)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = RAC)

(failover_mode=(type=select)(method=basic)(retries=20)(delay=5))
    )
  )
# --- 3rd entry: no load balancing, but connect time failover, TAF
method=preconnect
RAC1P =
  (DESCRIPTION =
    (enable=broken)
    (LOAD_BALANCE=NO)
    (FAILOVER=ON)
      (ADDRESS = (PROTOCOL = TCP)(HOST = node1)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node2)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node3)(PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP)(HOST = node4)(PORT = 1521))
    (CONNECT_DATA =
      (SERVICE_NAME = RAC)
      (failover_mode=(type=select)(method=preconnect)(backup=rac2f))
    )
  )
```

Figura 4.12 Entradas del archivo tnsnames.ora en el cliente

La primera entrada permitía la conmutación por falla de conexión entre las tres instancias si la conexión se establecía mediante el descriptor de conexión @**rac**. La línea (**enable = broken**) permitía al cliente hacer uso del mecanismo **keep alive** de TCP y detectar si una sesión debía ser mantenida viva o se la consideraba caída.

Esto se utilizaba si el servidor fallaba o no podía llegar a través de la red del cliente. El método de recuperación a fallas automática era utilizado, conectando la sesión a otra instancia cuando la conexión a la instancia original se perdía. Esto se volvía a intentar 20 veces con un retraso de 5 segundos entre reintentos.

La segunda entrada se utilizaba con el descriptor de conexión @**rac1f**. Esta entrada era la misma que la primera, excepto que el equilibrio de carga estaba desconectado. Las sesiones utilizando esta cadena de conexión por lo tanto se conectaban a la instancia **rac1** siempre y cuando esté disponible, caso contrario todas las conexiones se dirigían hacia la instancia **rac2**.

La última entrada utilizaba el método de recuperación por pre-conexión siempre y cuando el string de conexión **@rac1p** era especificado. Esto establecía dos conexiones a la vez cuando se iniciaba la sesión, pero solamente la conexión primaria la utilizaba siempre que esté disponible. Si esta conexión no estaba disponible, la sesión cambiaba a la conexión secundaria. Esto debía proporcionar una recuperación más rápida debido a que la conexión secundaria ya estaba establecida cuando la producía la falla simulada.

Bases de datos ejemplo:

Para una mejor planificación y gestión del ambiente de pruebas, no se almacenó los datos de Oracle (bases de datos, archivos de control y archivos de registro de recuperación) en el directorio $ ORACLE_HOME. Para esto creamos un filesystem **/data** del tipo GPFS que era compartido por los tres nodos y almacenaba los archivos de datos que fueron provistos por IBM para la ejecución de pruebas con Oracle.

Nos aseguramos los archivos de datos tenían como dueño a oracle y grupo dba, con los respectivos permisos de lectura y escritura. Los tamaños de los archivos de base de datos de prueba se muestran en la siguiente tabla:

Archivo de base de datos	tamaño	Nombre de archivo AIX
DATA_USER1	**5GB**	/data/rac/data01.dbf
DATA_USER2	**5GB**	/data/rac/data01.dbf
DATA_USER3	**5GB**	/data/rac/data01.dbf

Tabla 9. Nombre y tamaño de las bases de datos de prueba

Cada una de estos archivos de base de datos eran en si una tabla denominada DATA que contenía 5.2 millones de filas y ejecutaba búsquedas no paralelas que tardaban aproximadamente 100 segundos. Durante este tiempo, los datos fueron leídos desde los "*datafiles*" localizados en los filesystems del GPFS y las instancias intercambiaban información utilizando la red primaria del InterConnect. La estructura de cada registro era simple, con una sola columna de 50 caracteres.

4.6.2. Recuperación de redes internas y de usuarios luego de fallas de adaptadores

Para la prueba de recuperación de las redes internas y de usuarios, se simularon fallas de adaptadores ethernet de dos maneras: desconectando el cable RJ45 en uno de sus puertos y la segunda, vía shell script, ambas eran válidas pues se obtenía el mismo resultado. Las pruebas más importantes fueron las relacionadas con la interconexión del clúster de Oracle9i RAC para la comunicación entre las

instancias. La recuperación de esta red implicaba que el tráfico cambiaba a la interconexión secundaria, manteniendo así las sesiones activas.

Previa a esta simulación de falla de adaptadores, se prepararon y ejecutaron los respectivos scripts SQL descritos en la sección 4.5.1 para verificar el funcionamiento o no de la base de datos Oracle. Un diagrama de los tres servidores de base de datos con sus respectivos adaptadores, direcciones IPs y las redes que forman, puede ser observado a continuación:

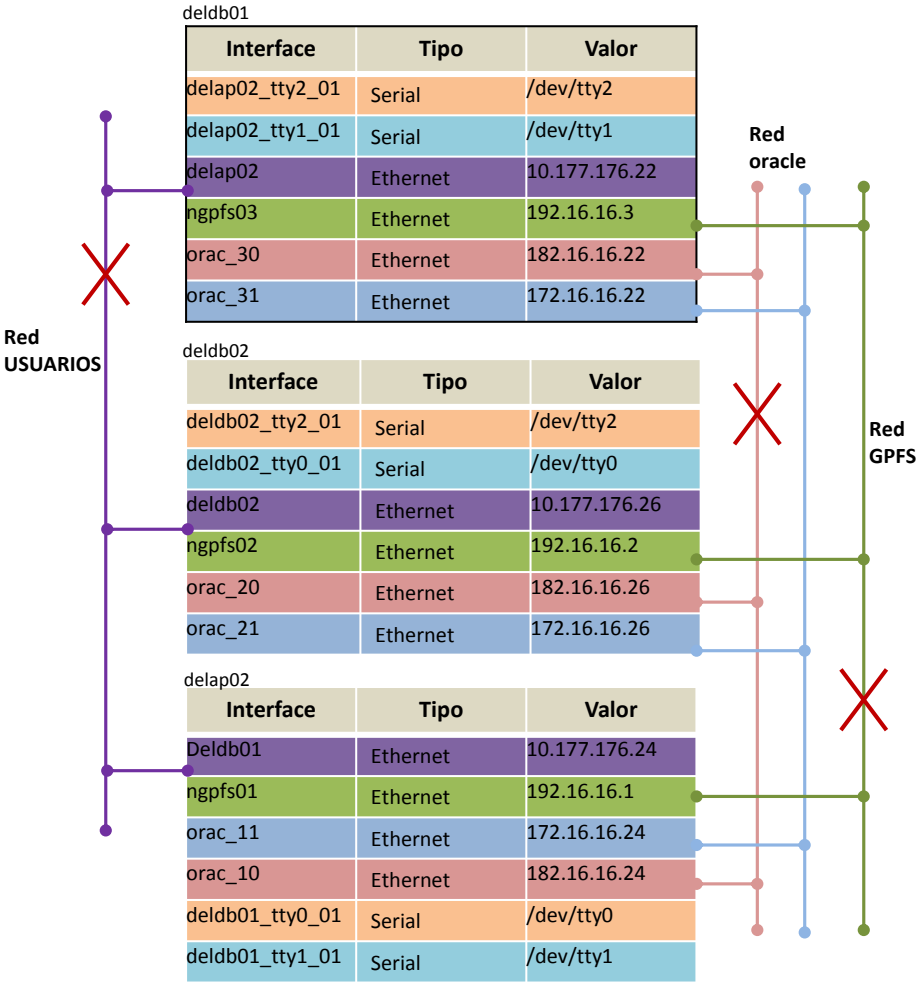


Figura 4.13 Redes internas y sus pruebas por fallas en adaptadores

En la figura anterior se puede apreciar las tres redes: Oracle con seis adaptadores formando dos redes; GPFS con tres adaptadores formando una red; Usuarios con tres adaptadores y una red. El proceso de pruebas implicaba tres casos de pruebas simulando la caída de adaptadores en diferentes instantes; la primera prueba consistía en simular la falla de uno de los adaptadores del InterConnect de Oracle.

Componente Fallando	Test QUERY	Se recuperó Sesión?
1. Ethernet Oracle orac_N0	Sleep 120, select from data;	Si o No
2. Ethernet Oracle orac_N1	Sleep 120, select from data;	Si o No
3. Ethernet GPFS ngpfs0N	Sleep 120, select from data;	Si o No
4. Ethernet usuarios	N/A	Si o No

Tabla 10. Secuencia de fallas de redes internas y de usuarios

La segunda prueba consistía en desconectar la red GPFS y verificar que una de las dos redes de Oracle RAC remplace y permita que GPFS siga funcionando.

Finalmente, para la última prueba que era la desconexión de la red de usuarios, el cliente debía proveer un Switch ethernet adicional para la formación de una segunda red de usuarios. Estaba claro que esta prueba no podía ser automática e implicaba un riesgo aceptado por el personal de Banco DelBank.

4.6.3. Recuperación de la base de datos luego de fallas en discos duros

Como se indicó en secciones anteriores, los discos duros internos y los del arreglo FastT600 estaban configurados con RAID 1 o espejamiento; por lo tanto la simulación de una falla de disco significaba deshabilitarlo o removerlo, verificando que la copia de éste se mantenga. Los discos internos eran SCSI y por estar dentro del servidor no podían ser removidos; por lo tanto, para simular una falla el disco debía ser deshabilitados vía comandos de sistema operativo y ser eliminado del espejamiento tal como se muestra a continuación:

```
#lsvg -l rootvg
hdisk0
hdisk1                          ← Disco a ser deshabilitado
unmirrorvg rootvg hdisk1
reducevg rootvg hdisk11
bosboot -a -d /dev/hdisk0       ← hdisk0 Nuevo disco bootable
```

Figura 4.14 Procedimiento para eliminar un disco interno

Para el caso de los discos del arreglo FastT600, era factible remover físicamente uno de ellos del slot, simplemente presionando el seguro y jalando de la agarradera para luego hacer la remoción lógica vía storage manager.

Para Oracle, la remoción de un disco interno o del FastT600 era algo transparente que no afectaba la operación de la base de datos, pues el espejamiento era manejado por AIX en el caso de discos internos y por el FastT600 en el caso de los discos del arreglo.

Figura 4.15 Disco duro del arreglo FastT600 removido del rack (tomado de [4])

La excepción podía darse cuando fallen las dos copias del espejamiento, caso que implicaría un punto de falla crítico y que acarrearía remplazar ambos discos y restaurar la base de datos de un respaldo o replicación. En todos estos años que Banco DelBank lleva en producción con los equipos IBM, nunca se ha dado un caso similar. A nivel de discos internos donde se aloja el sistema operativo AIX, una falla de los dos discos implicaría que el servidor ha fallado y automáticamente HACMP realizaría un remplazo (TAKEOVER) del mismo, concentrando toda la carga en el servidor disponible. Este caso es analizado en la siguiente sección.

4.6.4. Recuperación del cluster luego de la caída de un nodo

Para simular la caída de un nodo, tuvimos que separar los ambientes, tanto de aplicaciones como de base de datos. En el ambiente de aplicaciones, se tuvo que apagar uno de los dos nodos del cluster HACMP, mientras que en el de base de datos uno de los tres nodos del GPFS.

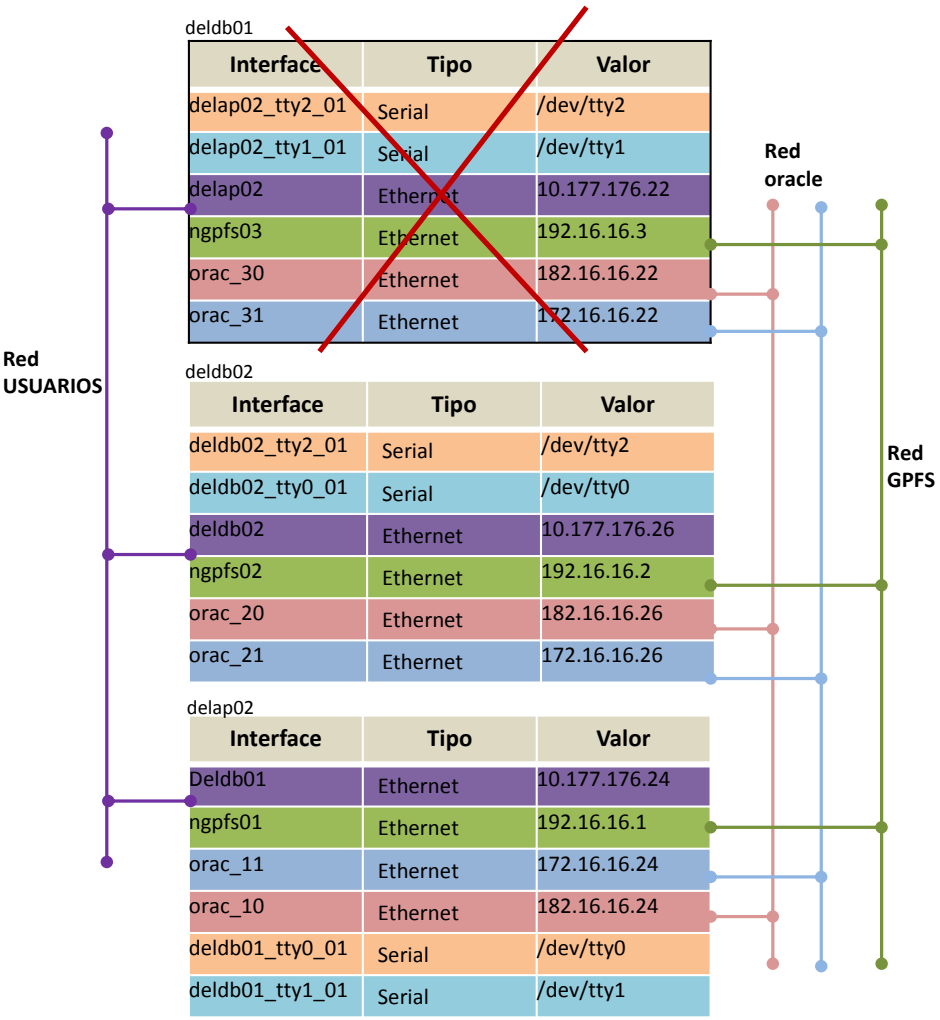


Figura 4.16 Simulación de caída del nodo deldb01

El proceso de simulación de caída de un nodo implicaba primero, hacerlo de una manera controlada por medio de comandos de sistema operativo y luego de una manera abrupta, apagando uno de los tres nodos. La secuencia en la cual simulamos las fallas de nodos se muestra en la figura.

Componente Falla	Tipo de Falla	Tipo de query	Se recupero sesión?
1. Nodo deldb01	Controlada	Select(…. Unión..); Select … from data;	Si o No
2. Nodo deldb01	Abrupta	Select(…. Unión..); Select … from data;	Si o No
3. Nodo delap01	Abrupta	Select(…. Unión..); Select … from data;	Si o No

Tabla 11. Secuencia de fallas de servidores pSeries

La manera controlada era necesaria para cuando se necesitaba realizar mantenimientos preventivos o correctivos en alguno de los servidores sin tener que parar el funcionamiento de todo el conjunto de servidores fuera de horarios de oficina.

4.7. Revisión de disponibilidad de aplicaciones y usuarios

Antes de realizar la simulación de fallas, preparamos tres computadores personales que se conectaban vía **telnet** o **ssh** a un servidor de aplicaciones; en cada uno de ellos, se ejecutaba una aplicación tipo shell script, que lanzaba consultas SQL a una base de datos. Esta aplicación era supervisada por nosotros y observábamos su comportamiento mientras se simulaban las diferentes fallas de recursos; es decir que se verificaba que finalicen correctamente las consultas realizadas sin importar la instancia a la cual se tenía que conectar para hacerlo.

Creamos un usuario AIX llamado **npora** sin atributos de usuario administrativo y se ejecutó el shell script antes mencionado cuya ubicación era el directorio **$TESTDIR** y cuyo nombre era **test_query.sh**; con esto se pudo chequear la alta disponibilidad y su contenido se muestra a continuación:

```
#!/bin/ksh
if [[ "$(whoami)" != "npora" ]]
then
echo I am $(whoami).
echo please run me as npora...
exit
fi

cd $TESTDIR

sqlplus pet/pet@$1 << ! > /dev/null
spool ha_test
@time_instance

select count(*) from
(  select * from dba_source
union
select * from dba_source
union
select * from dba_source
union
select * from dba_source
union
select * from dba_source);
@time_instance
spool off
exit
```

Figura 4.17 Script test_query.sh que verifica la disponibilidad de la base de datos

Este script se utilizaba para la sesión activa y la consulta SQL contenida en éste se ejecutaba durante un tiempo prudencial para verificar si la consulta deja de correr o no durante una condición de falla en el servidor. Para la sesión inactiva, solamente tuvimos que remplazar todo el shell script por el comando **sleep 120.**

El comando **spool**, creaba un archivo llamado ha_test que mostraba la secuencia de los comando ejecutados. Una salida típica guardada en el archivo antes mencionado se muestra en el ejemplo siguiente:

```
SQL> @time_instance
CURRENT_TIMESTAMP                         INSTANCE_NAME
--------------------------------- ----------------
05-JUN-03 09.56.30.181362 AM -04:00 rac1
SQL> select count(*) from
2 ( select * from dba_source
3 union
4 select * from dba_source
5 union
6 select * from dba_source
7 union
8 select * from dba_source
9 union
10 select * from dba_source);
COUNT(*)
----------
116652
SQL> @time_instance
CURRENT_TIMESTAMP                         INSTANCE_NAME
--------------------------------- ----------------
05-JUN-03 09.58.09.507318 AM -04:00 rac1
SQL> spool off
```

Figura 4.18 Salida del archivo spool ha_test ejecutado por test_query.sh

Aquí se puede observar que el query fue lanzado el 05 de Junio del 2004 a las 09:56:30 en la instancia rac1 y que terminó de ejecutarse a las 09:58:09 del mismo día, haciendo una búsqueda de 116652 registros de la base de datos dba_source, terminando de ejecutarse en la instancia de base de datos rac1.

4.7.1. Determinar accesos a la base de datos

El objetivo de esta sección era verificar la disponibilidad de la base de datos luego de una falla de servidor o de adaptador, verificando las instancias a las cuales se conectaban los clientes antes y después de la falla. En la sección 4.5.2 creamos el shell script test_query.sh para enviar un query de la base de datos utilizando una instancia denominada “racN” en un nodoN. En esta sección vamos a utilizar el mismo script pero dentro de un lazo infinito, el cual iba mostrando las diferentes salidas de ejecución a medida que se iban desconectando o conectando los cables ethernet o apagaban o prendiendo los nodos. El script database_access.sh se muestra a continuación:

```
#!/bin/ksh
sleeptime=45
connect=$1
if [[ “$(whoami)” != “root” ]]
then
echo I am $(whoami).
echo please run me as root ...
exit
fi
if [[ “$connect” = “” ]]
then connect=rac1f
fi
echo “$(date) Test will connect to $connect ...”
echo “$(date) Starting test ...”
echo “$(date) Starting query ...”

While true
do
su - npora “-c $TESTDIR/test_query.sh $connect” &
echo “$(date) Waiting for $sleeptime seconds ...”
sleep $sleeptime

# INICIO DEL CODIGO DE FALLA (ADAPTADOR O NODO)
echo “$(date) Waiting for query to finish ...”
wait
echo “$(date) Query finished ...”
cat ha_test.lst

# INICIO DEL CODIGO QUE REPARA LA FALLA
echo “$(date) Test finished ...”
done
```

Figura 4.19 database_access.sh. Muestra conexiones de la base de datos

En las líneas resaltadas en rojo, es donde se debería incluir el código para simular el tipo de falla, sea esta por caída de adaptador o de nodo. Por supuesto, estos scripts variaban dependiendo del tipo de simulación y se adjuntan en el capítulo 6.

Con este script era fácil de determinar la instancia y el nodo inicial al cual se conectaba un cliente antes de una falla, y luego nos mostraba la instancia y nodo final al cual se conectó luego de la misma.

4.7.2. Determinar número de conexiones de usuarios en nodos

Como parte del trabajo de monitoreo de las diferentes simulaciones de falla, se utilizaron shell scripts propios de Oracle los mismos que monitoreaban y certificaban que las conexiones de usuarios Oracle seguían activas.

Los scripts de Oracle a los cuales estoy haciendo referencia, utilizaban vistas; estas vistas nos permitían mostrar información sobre instancias de base de datos como el rendimiento, vistas internas del disco, conexiones, usuarios, estructuras de memoria para una instancia cualquiera y más. Las vistas se identificaban mediante el uso de un prefijo " V_$".

Uno de esto scripts disponibles y que fue utilizado por nosotros, era **sessions.sql**, el mismo que utilizaba la vista **v$session** y nos proporcionaba información sobre todas las sesiones conectadas a una instancia de base de datos. Además, junto con la vista anterior, utilizamos la **v$process** para obtener el número de proceso de

sistema operativo (SPID) para cada sesión. Una salida típica de este proceso que fue utilizado para monitorear las conexiones de usuarios se muestra a continuación:

```
SQL> @sessions
USERNAME OSUSER SID SERIAL# SPID STATUS MACHINE PROGRAM
LOGON_TIME
(oracle) SYSTEM 170    1    1016 ACTIVE BART    ORACLE.EXE (PMON) 22-
AUG-2005 11:50:51
(oracle) SYSTEM 169    1    2492 ACTIVE BART    ORACLE.EXE (MMAN) 22-
AUG-2005 11:50:51
(oracle) SYSTEM 168    1    2212 ACTIVE BART    ORACLE.EXE (DBW0) 22-
AUG-2005 11:50:51
(oracle) SYSTEM 167    1    3612 ACTIVE BART    ORACLE.EXE (LGWR) 22-
AUG-2005 11:50:51
(oracle) SYSTEM 166    1    3828 ACTIVE BART    ORACLE.EXE (CKPT) 22-
AUG-2005 11:50:51
(oracle) SYSTEM 157    1    3048 ACTIVE BART    ORACLE.EXE (MMNL) 22-
AUG-2005 11:51:06
(oracle) SYSTEM 160    7    308  ACTIVE BART    ORACLE.EXE (MMON) 22-
AUG-2005 11:51:06
(oracle) SYSTEM 165    1    2208 ACTIVE BART    ORACLE.EXE (SMON) 22-
AUG-2005 11:50:51
(oracle) SYSTEM 164    1    2572 ACTIVE BART    ORACLE.EXE (RECO) 22-
AUG-2005 11:50:51
(oracle) SYSTEM 163    1    3280 ACTIVE BART    ORACLE.EXE (CJQ0) 22-
AUG-2005 11:50:51
(oracle) SYSTEM 159    1    3020 ACTIVE BART    ORACLE.EXE (QMNC) 22-
AUG-2005 11:51:02
(oracle) 155          27    2152 ACTIVE                           22-
AUG-2005 12:01:35
(oracle) 153           3    2296 ACTIVE                           22-
AUG-2005 11:51:32
(oracle) 145           3    3392 ACTIVE                           22-
AUG-2005 11:51:30
14 rows selected.
```

Figura 4.20 Salida del shell script sessions.sql

Las sesiones activas se podían identificar en la columna de status con su respectivo número de proceso (SPID) y de usuario. Algunas veces insertábamos **sessions.sql** dentro de un lazo infinito para observar si las conexiones de usuario se perdían

.

CAPÍTULO 5

5. IMPLEMENTACIÓN DE LA SOLUCIÓN GPFS

En los capítulos anteriores se han revisado los pre-requisitos tanto de software como de hardware para realizar la instalación del cluster GPFS. En este capítulo se detalla el ambiente sobre el cual se realizó la instalación del cluster, la definición de los discos compartidos y finalmente la creación de los filesystem GPFS.

5.1. Instalación de servidores y dispositivos de red

Los tres servidores pSeries del cluster GPFS, fueron configurados con 4GB de memoria RAM y todos con sistema operativo AIX 5.2; se realizó un chequeo del cumplimiento de pre-requisitos de sistema operativo (***filesets***) y GPFS. A nivel de configuración del sistema operativo, se definió un espacio para paginamiento equivalente al doble del tamaño de la memoria RAM y finalmente se procedió a realizar el espejamiento de los dos discos de sistema operativo.

Las tareas ejecutadas para instalar los servidores y los dispositivos de red se realizaron secuencialmente y se detallan a continuación:

Se verificó que el software de sistema operativo AIX y del aplicativo GPFS estén correctamente instalados y a último nivel de parches. Para lograr esto, se ejecutaron los comandos **lppchk** y **oslevel**. Para la interoperabilidad entre el software, GPFS versión 2.1 con AIX 5.2 se instalaron los siguientes parches de sistema operativo: IY36782, IY37744, IY37746, IY36626. Mientras que a nivel de soporte de base de datos Oracle: U488744, U488745, U488746, U488747.

Se asignaron nombres únicos (***hostname***) a cada uno de los tres nodos participantes del cluster a nivel de red pública: deldb02 (IP=10.177.176.24), deldb01 (IP=10.177.176.24) y delap02 (IP=10.177.176.22) y se verificó que exista la conexión respectiva entre ellos utilizando el comando ping.

Se asignaron tres nuevos nombres de host para cada una de las direcciones IPs de la red privada GPFS: ngpfs01 (IP=192.16.16.2), ngpfs02 (IP=192.16.16.2) y ngpfs03 (IP=192.16.16.3) luego se verificó que exista la conexión respectiva entre ellos utilizando el comando ping. Se definió la conexión entre nodos para la copia de archivos remotos. Para esto se creó el archivo **.rhosts** en cada directorio raíz de los nodos. Dentro de este archivo se añadió en cada línea el nombre de cada uno de los hosts tanto de la red pública como de la GPFS, para de esta manera tener acceso al usuario administrador **root** de manera remota. Al ejecutar el comando **cat** para cada uno de estos archivos se mostraba lo siguiente:

```
#cat .rhosts
deldb01
deldb02
delap02
ngpfs01
ngpfs02
ngpfs03
```

Figura 5.1 Contenido del archivo /.rhosts

Para evitar el acceso remoto por parte de otro usuario diferente al usuario **root**, se asignó un permiso de archivo **600(rw- --- ---),** es decir solo permisos de lectura y escritura para el dueño del archivo, el usuario **root**.

Se realizaron pruebas de acceso remotos desde cualquier nodo. Para esto se ejecutaron comandos para copiar archivos; uno de ellos fue el **rcp** por ejemplo:

ngpfs03# **rcp** ngpfs02:/etc/hosts ngpfs01:/

Es decir, que desde el nodo ngpfs03 se copió el archivo **/etc/hosts** que reside en el nodo ngpfs02, al directorio raíz del nodo ngpfs01.

Se verificó que a nivel de sistema operativo los 14 discos aparezcan definidos en el arreglo FastT600. Para lograr esto, se ejecutó el comando **lspv**.

Se revisó la memoria física de los servidores con el comando: **lsattr –El sys0 –a realmem** el mismo que nos devolvió el valor 4096MB es decir 4GB, lo cual es lo adquirido por el cliente. Con esta información, se procedió a incrementar el área de paginamiento en disco al doble de la memoria RAM que era lo recomendado por GPFS. Con esto se podría realizar el *swap* o transferencia de datos desde la RAM al disco y viceversa en caso de falto de recursos de memoria real. Con el comando **chgps** se incrementó el tamaño del espacio para paginamiento de 4GB en el disco hdisk0 del volumen lógico **/dev/hd6**, y se creó un segundo de 4GB en el disco hdisk1 cuyo nombre fue: **/dev/paging00**. El comando **lsps –a** nos mostró el tamaño creado 8192MB es decir 8GB.

Por cada servidor pSeries, se definió un disco para sistema operativo AIX y ejecutables de Oracle con su respectivo disco para espejamiento. Con el comando **lsvg –l rootvg,** se verificó que el sistema operativo AIX reside en el disco **hdisk0**, mientras que el

hdisk1 no se encontraba asignado a ningún grupo de volúmenes, entonces se añadió el disco hdisk1 al grupo de volúmenes **rootvg** con el comando **extendvg rootvg hdisk1.**

Se definió el espacio temporal de Oracle, cuya instalación requirió de al menos 400MB de espacio libre en el filesystem **/tmp**. Las variables de Oracle TEMP y TEMPDIR se direccionaban al mismo filesystem temporal.

Una vez creados los archivos se pudo diferenciarlos. En rojo, los filesystems localizados dentro de los discos internos de los nodos de base de datos y en verde los filesystems en los discos compartidos vía canal de fibra (FC).

```
Filesystem    1024-blocks      Used      Free %Used Mounted on
/dev/hd4           131072     98316     32756   76% /
/dev/hd2          1441792    924756    517036   65% /usr
/dev/hd9var        131072     72112     58960   56% /var
/dev/hd3           917504    234140    683364   26% /tmp
/dev/hd1           131072       548    130524    1% /home
/proc                   -         -         -    -  /proc
/dev/hd10opt       131072     10012    121060    8% /opt
/dev/oralv        9043968   4811708   4232260   54% /oracle
/dev/dbdata01   213384192  92741632 120642560   44% /dbdata01
/dev/dbdata02   142081024   3088384 138992640    3% /dbdata02
/dev/dbdata03    71040000  19283968  51756032   28% /dbdata03
```

Figura 5.2 Filesystems del GPFS compartidos por los servidores base datos

Esto quiere decir que los filesystems /dbdata01 /dbdata02 /dbdata03 son los que residían en el almacenamiento externo y accesados concurrentemente y mantenían un espacio de 203.5GB, 135.5GB y 67.70GB respectivamente.

5.1.1. Instalación del cluster GPFS en servidores pSeries

Una vez que se ha realizado la respectiva revisión del hardware y del software y sus prerrequisitos, se procedió a la instalación del software GPFS en cada uno de los servidores pSeries del cliente DelBank. Antes de empezar, se tuvo que revisar el cumplimento de ciertos prerrequisitos y verificar que la variable de ambiente **PATH** abarcaba la definición **/usr/lpp/mmfs/bin** directorio donde residía el software GPFS en cada uno de los nodos del cluster.

Adicionalmente hubo que crear un archivo ASCII que contenía en cada línea un nombre de nodo perteneciente al cluster. Previa a la instalación, tuvimos que crear el mencionado archivo el mismo que lo nombramos: **/tmp/gpfs.allnodes** y contenía los nodos ngpfs01 ngpfs02 ngpfs03.

Tal cual se indicó en el punto 3.3 relacionado a la preparación de la base de datos, el cluster a implementarse era del tipo RPD y no HACMP por lo tanto, los filesets que debían ser instalados en cada nodo eran los siguientes.

```
rsct.basic.rte 2.2.1.20 COMMITTEDRSCT Basic Function
rsct.compat.basic.rte 2.2.1.20 COMMITTEDRSCT Event Management
rsct.compat.clients.rte 2.2.1.20 COMMITTEDRSCT Event Management
rsct.core.auditrm 2.2.1.20 COMMITTEDRSCT Audit Log Resource Manager
rsct.core.errm 2.2.1.20 COMMITTEDRSCT Event Response Resource Manager
rsct.core.fsrm 2.2.1.20 COMMITTEDRSCT File System Resource Manager
rsct.core.hostrm 2.2.1.20 COMMITTEDRSCT Host Resource Manager
rsct.core.rmc 2.2.1.20 COMMITTEDRSCT Resource Monitoring and Control
rsct.core.sec 2.2.1.20 COMMITTEDRSCT Security
rsct.core.sr 2.2.1.20 COMMITTEDRSCT Registry
rsct.core.utils 2.2.1.20 COMMITTEDRSCT Utilities
```

Figura 5.3 Software GPFS instalado en los nodos

Se hizo la respectiva verificación del software del cliente y se comprobó que todos los filesets del RPD estaban disponibles. El proceso de instalación del software

GPFS implicó le ejecución de un conjunto de pasos secuenciales usando el comando de instalación **installp**. Este procedimiento instalaba GPFS en un nodo a la vez y se detallan a continuación.

Creación del directorio GPFS

Para instalar el software GPFS, hubo que bajar las imágenes desde el DVD de instalación a un directorio ubicado en cualquiera de los nodos. Nosotros elegimos al **ngpfs01** y en éste creamos el subdirectorio **/tmp/gpfslpp** y a continuación copiamos las imágenes de instalación en éste utilizando el comando **bffcreate**:

Bffcreate –QVX -t /tmp/gpfslpp –d /dev/cd0

Este procedimiento copió los siguientes filesets en el directorio de imágenes:

mmfs.base.usr.3.5.m.f

mmfs.gpfs.usr.2.1.m.f

mmfs.msg.en_US.usr.3.5.m.f

mmfs.gpfsdocs.data.3.5.m.f

Crear las imágenes de instalación GPFS.

Hicimos del nuevo directorio donde estaban las imágenes, sea directorio actual. Utilizando el comando **inutoc** creamos un archivo **.toc** el mismo que es utilizado por el comando **installp** para instalar filesets, tanto de sistema operativo como de GPFS.

_____

```
cd /tmp/gpfslpp
```

```
inutoc .
installp -l -d . mmfs | more
```

Figura 5.4 Creación de imágenes de instalación GPFS

Instalar GPFS desde la red interna Gigabit

En vista a que los nodos ya se encontraban en red, interconectados por medio de sus tarjetas ethernet Gigabit, se procedió a crear un sistema de archivos compartidos. Utilizando NFS (Network File System) procedimos a la correspondiente exportación del directorio **/tmp/gpfslpp** para que estuviese disponible en todos los nodos a instalarse. Para esto tuvimos que exportar el directorio local definiéndolo dentro del archivo de exportación **/etc/exports**.

exportfs -i /tmp/gpfslpp

En el servidor de donde exportamos el filesystem verificamos si ha sido exportado correctamente.

showmount -e ngpfs01

El sistema mostraba información similar al siguiente:

export list from ngpfs01:

/tmp/gpfslpp ← **En todos los nodos**

En cada nodo montamos el filesystem que mantenía la copia de la imagen del software GPFS y que residía en ngpfs01:

mount ngpfs01 /tmp/gpfslpp /mnt

En el primer nodo ngpfs01 ejecutamos el comando **installp** para instalar GPFS:

installp –agXYd /tmp/gpfslpp all

Finalmente instalamos GPFS en el resto de los nodos de forma individual, ejecutando el mismo comando **installp** en cada uno de ellos:

Installp -agXYd /mnt all

Verificación de la instalación GPFS

Para la respectiva verificación de los filesets de GPFS, utilizamos el comando **lslpp** en cada uno de los nodos:

```
ngpfs01:/ # lslpp -l mmfs\*
Path: /usr/lib/objrepos
mmfs.base.cmds 3.5.0.0 COMMITTED GPFS File Manager Commands
mmfs.base.rte 3.5.0.0  COMMITTED GPFS File Manager
mmfs.gpfs.rte 2.1.0.0  COMMITTED GPFS File Manager
mmfs.msg.en_US 3.5.0.0 COMMITTED GPFS Server Messages - U.S. EN.
Path: /etc/objrepos
mmfs.base.rte 3.5.0.0 COMMITTED GPFS File Manager
mmfs.gpfs.rte 2.1.0.0 COMMITTED GPFS File Manager
Path: /usr/share/lib/objrepos
mmfs.gpfsdocs.data 3.5.0.0 COMMITTED GPFS Server Manpages
```

Figura 5.5 Software GPFS instalado

5.1.2. Solución de problemas por definiciones de quórum

Tal cual se indicó en el capítulo 3, el cluster GPFS fue inicialmente vendido para que trabaje solo con dos nodos IBM p630; pero si un nodo llegaba a fallar el cluster dejaba de funcionar. Estaba claro que un solo nodo activo no podía mantener el quorum de al menos 51%, pues se necesitaba de al menos dos nodos mas para completar el quorum requerido. La alternativa seleccionada por IBM para solucionar el problema fue adicionar un tercer nodo modelo pSeries 615 al cluster, asumiendo el costo de éste.

Una vez añadido el nuevo nodo, los filesystems GPFS definidos en el almacenamiento FastT600 **/dbdata01, /dbdata02 y /dbdata03** se mostraban de la siguiente manera:

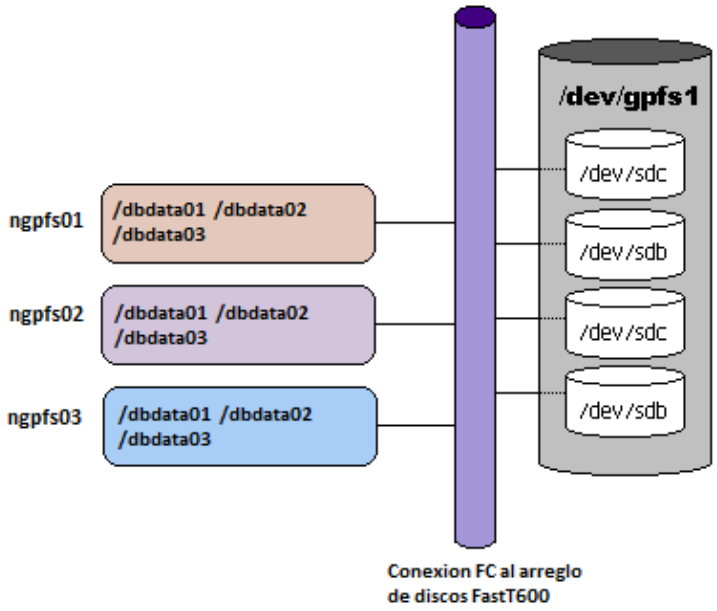


Figura 5.6 Disponibilidad de filesystems con los tres nodos

Al mostrar los filesystems vía comando df –kt (*disk free*) en cada uno de los servidores, se podía observar los mismos definidos en cada servidor, tal cual se detalla a continuación. En color rojo se observan los filesystems compartidos GPFS mientras en color negro los filesystems que residían en los discos internos.

```
Filesystem    1024-blocks      Used      Free %Used Mounted on
/dev/hd4           131072     26928    104144   21% /
/dev/hd2          1703936   1211216    492720   72% /usr
/dev/hd9var        131072     56520     74552   44% /var
/dev/hd3          1310720    111780   1198940    9% /tmp
/dev/hd1           131072     81032     50040   62% /home
/proc                   -         -         -    -  /proc
/dev/parche        4587520   1617900   2969620   36% /parche
/dev/delbank73     6291456   1981724   4309732   32% /delbank73
/dev/oralv        10223616   9458728    764888   93% /oracle
/dev/appslv        6291456   1306868   4984588   21% /apps
/dev/dbdata01    213384192  92741632 120642560   44% /dbdata01
/dev/dbdata02    142081024   3088384 138992640    3% /dbdata02
/dev/dbdata03     71040000  19283968  51756032   28% /dbdata03
```

Figura 5.7 Salida del comando df –kt en el nodo gpfs01

```
Filesystem    1024-blocks      Used      Free %Used Mounted on
/dev/hd4           131072     98316     32756   76% /
/dev/hd2          1441792    924756    517036   65% /usr
/dev/hd9var        131072     72112     58960   56% /var
/dev/hd3           917504    234140    683364   26% /tmp
/dev/hd1           131072       548    130524    1% /home
/proc                   -         -         -    -  /proc
/dev/oralv         9043968   4811708   4232260   54% /oracle
/dev/dbdata01    213384192  92741632 120642560   44% /dbdata01
/dev/dbdata02    142081024   3088384 138992640    3% /dbdata02
/dev/dbdata03     71040000  19283968  51756032   28% /dbdata03
```

Figura 5.8 Salida del comando df –kt en el nodo gpfs02

```
Filesystem    1024-blocks      Used      Free %Used Mounted on
/dev/hd4           131072     88060     43012   68% /
/dev/hd2          1441792    923848    517944   65% /usr
/dev/hd9var        131072     70428     60644   54% /var
/dev/hd3           917504    192580    724924   21% /tmp
/dev/hd1           131072       532    130540    1% /home
/proc                   -         -         -    -  /proc
/dev/oralv         9043968   6863828   2180140   76% /oracle
/dev/dbdata01    213384192  92741632 120642560   44% /dbdata01
/dev/dbdata02    142081024   3088384 138992640    3% /dbdata02
/dev/dbdata03     71040000  19283968  51756032   28% /dbdata03
```

Figura 5.9 Salida del comando df –kt en el nodo gpfs03

Al momento simular una caída de uno de los tres nodos ya sea desconectándolo de la red GPFS o apagándolo, el quorum se mantenía en un 66%, esto es arriba del 51% mínimo para mantener la información disponible. Una simulación de caída del nodo ngpfs02 se muestra a continuación:

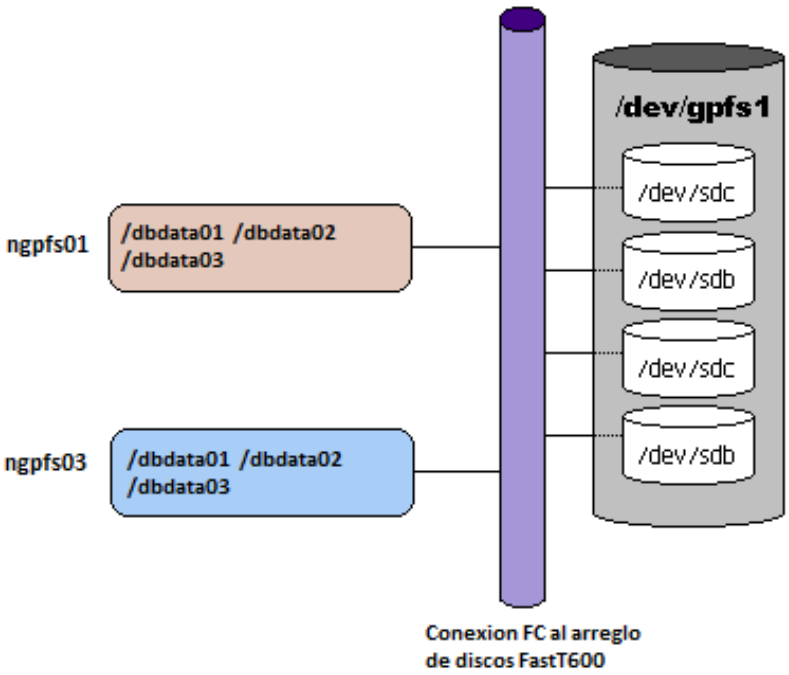


Figura 5.10 Cluster GPFS activo con uno de los tres nodos caídos

Con esto, al ejecutar nuevamente el comando **df –kt,** se podía comprobar que los filesystems internos o del GPFS estaban disponibles en los dos nodos restantes ngfps01 y ngpfs03.

5.1.3. Conflictos existentes con versiones de arreglo de discos

En el capítulo 3 se indicó que IBM tuvo que asumir el costo de un servidor pSeries para poder completar el quorum GPFS con al menos dos servidores disponibles y no

uno. El principal problema radicaba era que para ese entonces la versión 8.3 del *IBM Storage Manager*, aún no permitía el correcto manejo concurrente de discos compartidos por parte de más de un servidor pSeries.

Las versiones anteriores proporcionaban una capacidad muy básica para reserva de discos a través de comandos de “liberación” y “reservación” enviados por servidores conectados concurrentemente. Esta reserva y liberación se mantenía en base a quórums y fue pensada originalmente para instalaciones grandes que incluían muchos servidores; como es de fácil de imaginar, casi nunca tenían que soportar caídas de muchos nodos a la vez y tener que manejarse con uno o dos de ellos.

En nuestro país la realidad era un poco diferente, pues la capacidad de procesamiento era mucho menor y clusters de dos o tres nodos eran muy comunes y por lo tanto tuvimos que añadir nodos extras para compensar la pérdida de quorum. IBM tuvo que solucionar este problema con las nuevas versiones del *Storage Manager*, empezando con la 8.4 y mejorándola para evitar los conflictos existentes; algunas de estas mejoras fueron:

- Definición de un modelo bien definido para reservar discos a través de múltiples hosts y puertos destino (***target ports***).
- Mejoras en los niveles de control de acceso, por ejemplo lecturas compartidas, escrituras y lecturas exclusivas o dedicadas.
- Posibilidad de consultar información del sistema de almacenamiento a través de cualquiera de los caminos (**paths**) o puertos registrados y reservados.
- Mejoraras en el algoritmo para evitar la pérdida de quorum y para mantener la información grabada ante el eventual caso de pérdida de energía en el sistema de discos y mantenerse operativo ante la pérdida de N-1 nodos del cluster.

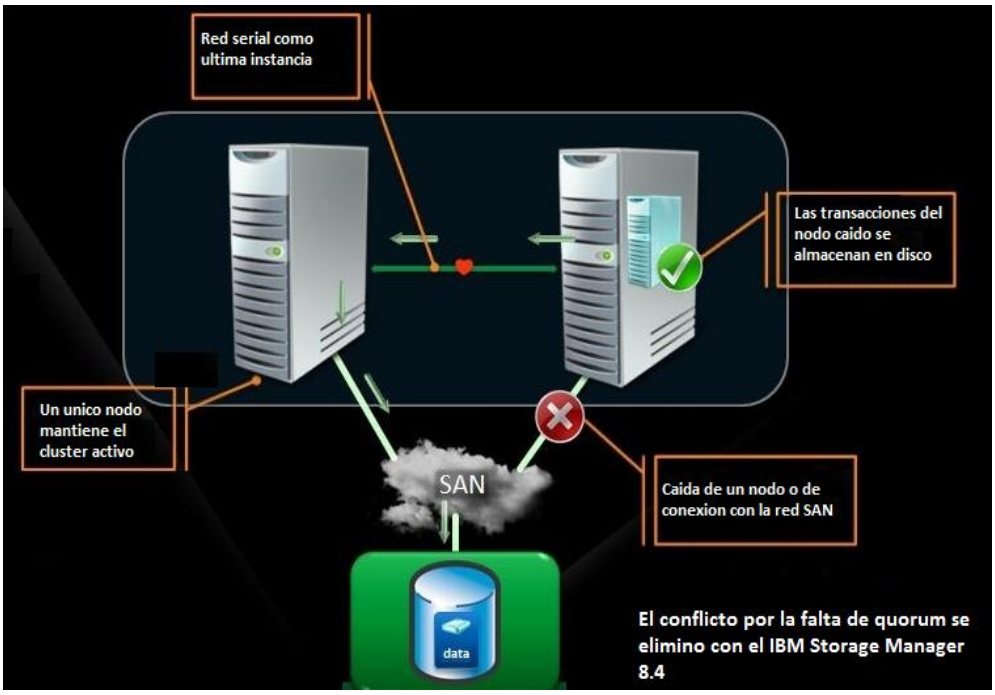


Figura 5.11 Solución al conflicto por falta de quórum (tomado de [7])

Las Reservas de discos se configuraban y gestionaban a través del software de servidor de clúster y preservaban las reservas de las unidades lógicas (LUNs) y registros para prevenir que otros hosts accedan a la unidad o unidades mencionadas. Además, se permitía que una unidad lógica principal tener espejamiento remoto.

La versión 8.4 además permitió el uso de unidades lógicas secundarias. Si una unidad lógica no tenía ningún tipo de reserva cuando haya sido designada como una unidad lógica secundaria, la unidad lógica principal detectaba un conflicto en la reserva en su primera solicitud de escritura al secundario y la unidad lógica y se borraba de la reserva automáticamente. Con esto, las solicitudes posteriores para realizar una reserva en la unidad lógica secundaria, se rechazaban. Finalmente, cabe aclarar que la versión 8.4 del Storage Manager sólo admitía FAStT600 Turbo,

FAStT700 y T900. Banco Delbank adquirió la versión 8.4 del Storage Manager en el año 2005 y utilizaron el nodo extra como nodo de producción.

5.2. Configuración y particionamiento de redes

Aunque las redes internas GPFS y Oracle 9iRAC no fueron particionadas por ser solo para la interconexión y transporte de datos entre nodos y el subsistema de discos, a nivel de usuarios fue necesario la instalación de un firewall para proteger la red de usuarios los cuales compartían información con otras sucursales en otros países y mantenían accesos a la internet.

La red de usuarios no fue duplicada por el costo que significaba instalar otro switch para definir la nueva red de respaldo. En caso de falla de red de usuarios, el cliente procedería a remplazar el switch con problemas con uno que ellos adquirieron justamente para casos de emergencias. Pero entre la red de usuarios locales y los servidores IBM pSeries, el personal de Banco DelBank instaló un firewall Cisco para prevenir posibles accesos no deseados de usuarios a los servidores de base de datos o aplicaciones.

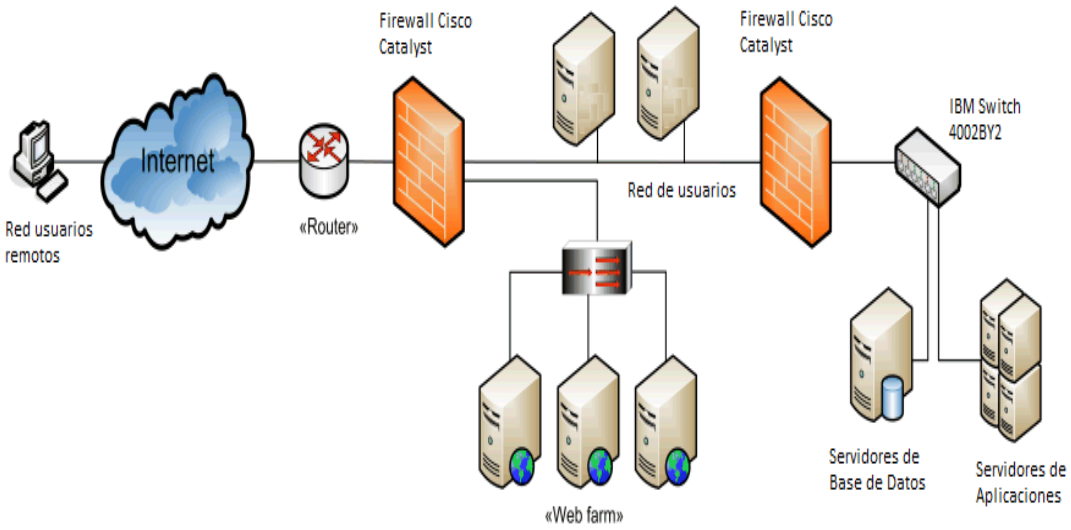


Figura 5.12 Diagrama de la configuración final de las redes locales (tomado de [5])

Finalmente se instaló un segundo firewall entre la red LAN y el internet para controlar los accesos de clientes remotos que se encontraban en España y accesos no deseados por parte de potenciales hackers.

5.2.1. Configuración de adaptadores ethernet

El rendimiento de las redes internas como de usuarios depende del hardware elegido, los adaptadores de red y de su posición en los slots de cada servidor. Para obtener un mejor rendimiento a nivel de adaptadores ethernet, antes que nada tuvimos que colocarlos en las ranuras de bus de E/S que eran las más adecuadas. En vista a que los servidores pSeries 615 y 630 soportaban tecnología de 64 bits, los adaptadores de red se colocaron en las ranuras de 64 bits, las que ofrecían la mayor velocidad de ciclo de reloj, en nuestro caso 133 MHz.

Otro parámetro que tuvimos que configurar fue el firmware del sistema pues es el responsable de configurar varios parámetros clave para cada adaptador PCI incluyendo los ethernet. Con un nivel de firmware actualizado garantizamos una correcta transmisión de datos entre los diferentes buses de E/S del sistema. Para confirmar el tipo de plataforma y el nivel de firmware del sistema tuvimos que ejecutar el comando **lscfg -vp|grep -p " ROM"** el mismo que mostraba lo siguiente:

```
System Firmware:
       ROM Level (alterable).......M2P030828
       Version.....................RS6K
       System Info Specific.(YL)...U0.1-P1/Y1
     Physical Location: U0.1-P1/Y1

     SPCN firmware:
       ROM Level (alterable).......0000CMD02252
       Version.....................RS6K
       System Info Specific.(YL)...U0.1-P1/Y3
     Physical Location: U0.1-P1/Y3

     SPCN firmware:
       ROM Level (alterable).......0000CMD02252
       Version.....................RS6K
       System Info Specific.(YL)...U0.2-P1/Y3
     Physical Location: U0.2-P1/Y3

     Platform Firmware:
       ROM Level (alterable).......MM030829
       Version.....................RS6K
       System Info Specific.(YL)...U0.1-P1/Y2
     Physical Location: U0.1-P1/Y2
```

Figura 5.13 Comando lscfg mostrando niveles de firmware de adaptadores

El ajuste predeterminado para adaptadores ethernet bajo AIX es "**Auto Negociación**", es decir que negocia la configuración de velocidad para las más altas velocidades de datos posibles. Para lograr que esto funcione tuvimos que configurar también los puertos del otro extremo del cable como switch u otros adaptadores para el caso de redes punto a punto. Los adaptadores de Ethernet podían ser configurados bajos los siguientes modos:

10_Half_Duplex

10_Full_Duplex

100_Half_Duplex

100_Full_Duplex

Auto_Negotiation

Si por ejemplo, un adaptador se ajustaba manualmente a una velocidad especifica cómo modo dúplex, en el otro extremo también se debía configurar manualmente con la misma velocidad y modo dúplex para mantener un buen performance de la red. Pero en definitiva, tanto para las redes internas como de usuarios, utilizamos el modo de auto negociación.

Otro parámetro que tuvimos que configurar en todos los adaptadores de a red física ethernet, fue el tamaño de la unidad del medio de transmisión **MTU**. Este es el tamaño máximo de un paquete (o **frame**) que se puede enviar en el medio o cable ethernet. Aunque los adaptadores soportan diferentes tamaños de MTU, nosotros NO utilizamos el tamaño de MTU predeterminado de ethernet que es 1500 bytes sino para paquetes extra grandes de 9000 bytes.

Cuando un adaptador de red recibe una trama mayor que su MTU, los datos se fragmentan en pequeños frames o se cae. Un paquete ethernet mayor de 1500 bytes se llama **Jumbo Frame** y en AIX éste puede ser activado vía menú del SMIT, la herramienta de administración de sistema operativo.

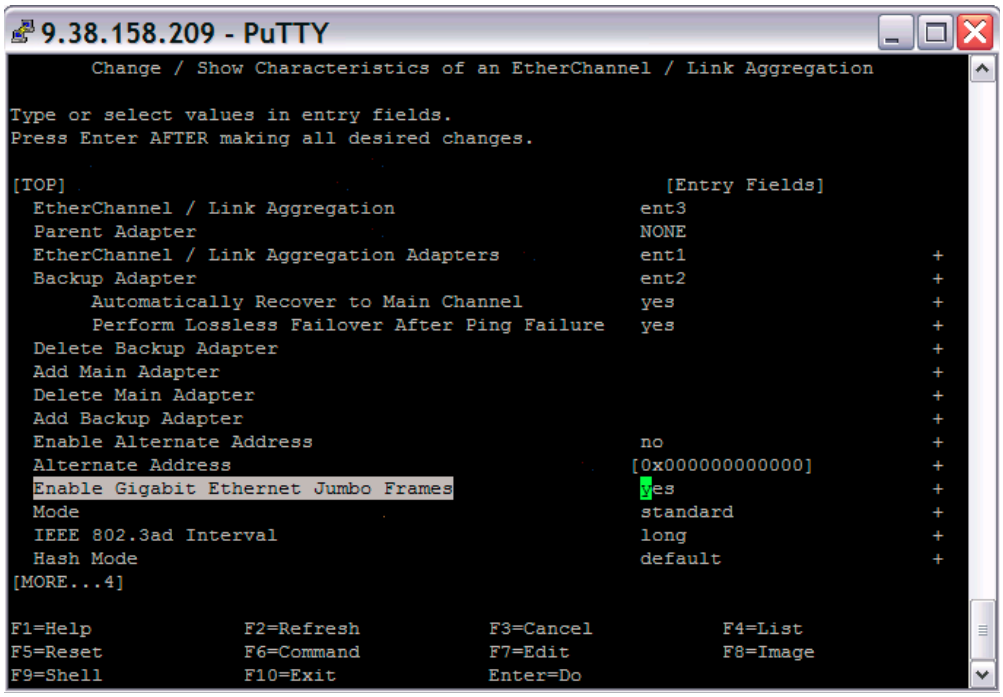


Figura 5.14 Pantalla de menús del SMIT para configurar adaptadores ethernet

Nosotros activamos la opción de Jumbo Frame, como parte de una recomendación para obtener mejor rendimiento en las redes ethernet **Gigabit**, descrita en uno de los manuales de instalación. El menú **SMIT** (System management Interface Tool) que muestra cómo cambiar este parámetro se muestra en la figura 5.14. Luego de la configuración a nivel de adaptadores ethernet, se pudo configurar la interface de red a nivel de software TCP utilizando el comando **no** (Network Options). Una salida típica de los valore TCP se pueden mostrar con el comando **ifconfig.**

```
# ifconfig en0
en0:
flags=5e080863,c0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,G
ROUPRT,64BIT,CHECKSUM_OFFLOAD,PSEG,CHAIN>
inet 192.0.0.1 netmask 0xffffff00 broadcast 192.0.0.255
tcp_sendspace 262144 tcp_recvspace 131072 rfc1323 1
```

Figura 5.15 Detalles de la configuración de un adaptador de red en0

5.2.2. Solución de problemas por incompatibilidades con adaptadores de red

Cuando se pasó a la etapa de pruebas de GPFS y HACMP en los tres nodos de base de datos y en los dos de aplicaciones, tuvimos un problema con el montaje de los filesystems GPFS y que estaban relacionados directamente con el Oracle 9iRAC. Oracle no reconocía las redes del **InterConnect**, es decir las redes formadas por los adaptadores Gigabit conectadas al switch ethernet IBM y se empezaron a reportar errores que reportaban la caída de los enlaces ethernet cuando se desconectaba uno de los adaptadores de la red interna de Oracle o de GPFS. Estos errores se muestran a continuación y se repetían cada tres minutos:

```
LABEL:       GOENT_LINK_DOWN
IDENTIFIER:  DED8E752

Date/Time:       Tue May 18 22:57:18 WEST
Sequence Number: 6765
Machine Id:      000FC75A4C00
Node Id:         deldb02
Class:           H
Type:            TEMP
Resource Name:   ent2
Resource Class:  adapter
Resource Type:   14106902
Location:        U0.1-P2-I6/E1
VPD:
        Product Specific.(  ).......10/100/1000 Base-TX PCI-X Adapter
        Part Number.................00P3056
        FRU Number..................00P3056
        EC Level....................H11635A
        Manufacture ID..............YL1021
        Network Address.............000255533C76
        ROM Level.(alterable).......GOL001

Description
ETHERNET DOWN

Probable Causes
CABLE
CSMA/CD ADAPTER

Failure Causes
LINK TIMEOUT

       Recommended Actions
       CHECK CABLE AND ITS CONNECTIONS
```

```
Detail Data
FILE NAME
line: 180 file: goent_intr.c
PCI ETHERNET STATISTICS
0000 0216 0063 081B 0000 0004 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0005 B548 0000 0000 02B5 909E 0000 0000 0004 FEB8 0000 0000
0000 0000 0005 A3CE 0000 0000 039D F131 0000 0000 0000 0000 0000 0001
0000 0000 0000 0000 0000 0000 0000 0004 0000 0000 0000 0004 0000 0000
DEVICE DRIVER INTERNAL STATE
2222 2222 0000 0000 0000 0000
SOURCE ADDRESS
0002 5553 3C76
```

Figura 5.16 Detalles del error obtenido al configurar la opción Jumbo Frame

Este problema nos tuvo paralizados casi dos semanas. Analizamos el tamaño de los paquetes ethernet enviados en la red interna y notamos muchos de ellos excedían el tamaño de 1500 bytes, que era el estándar para este tipo de redes. Lo complicado del caso era que se siguió paso a paso el libro rojo **SG24-6954-00** "*Deploying Oracle 9i RAC On IBM Eserver Cluster 1600 with GPFS*" y todos los pasos fueron ejecutados en base a las recomendaciones dadas allí. Como resultado del análisis de red, se logró identificar que justo cuando un adaptador de red recibía paquetes extra grandes, mayores a 9000 bytes, las redes internas y por ende los adaptadores colapsaban generando en ese preciso momento el error **GOENT_LINK_DOWN** descrito anteriormente. La recomendación decía textualmente lo siguiente:

We also recommend that you enable the Jumbo Frame (Maximum Transmission Unit - MTU=9000) for the Gigabit Ethernet adapters. You can use the following SMIT panel procedures to enable the Jumbo Frame:

smitty eadap
Change / Show Characteristics of an Ethernet Adapter
(select the Gigabit Ethernet Adapter)
Transmit Jumbo Frames: yes

Figura 5.17 Procedimiento erróneo en el manual de instalación

Cambiamos el parámetro **Transmit Jumbo Frames** a NO e inmediatamente el Oracle InterConnect subió sin problemas. Luego se procedió a realizar las pruebas

correspondientes, es decir simulando caídas de los adaptadores Gigabit y caídas de poder en un nodo.

5.3. Ejecución de aplicaciones Oracle en los diferentes servidores

Para la instalación de la base de datos Oracle 9iRAC fue necesario instalar varios paquetes de software. Tal como se hizo para GPFS, la ejecución del software de Oracle se la hizo desde un sistema de archivos GPFS, creando así una sola copia de la imagen binaria. Esta opción era la más conveniente porque toda la configuración y mantenimiento de SW se podía realizar desde cualquier nodo en el clúster.

En cada nodo, creamos el directorio **/var/opt/oracle** cuyo dueño y grupo **oracle** y **dba** respectivamente. Durante la ejecución de los instaladores se creaba un archivo llamado **srvConfig.loc**, el mismo que se utilizaba para especificar el destino de la aplicación de administración de servidor de Oracle (**srvctl).**

Nosotros estandarizamos el directorio donde irían a residir los archivos ejecutables de Oracle como **/oracle**. Este era un filesystem de 10GB y estaba definido en los discos internos de todos los nodos, tanto de aplicaciones como de base de datos. A este filesystem se le asignó al usuario **oracle** y grupo **dba** con el comando **chown -R oracle.dba /oracle** Luego se configuró para cada vez que el usuario oracle ingrese al sistema, cargue las variables de ambientes, tal cual lo hacía el “autoexec.bat” del DOS. Para esto hizo falta crear el archivo **$HOME/.profile** y dentro de éste las respectivas configuraciones y variables de ambiente tales como ORACLE_SID, TMPDIR y ORACLE_HOME.

Para empezar la instalación de Oracle RAC, iniciamos la sesión como usuario oracle en el nodo ngpfs01 e iniciamos el instalador de Oracle Universal ejecutando el comando **runInstaller**, cuya ventana de bienvenida que se muestra a continuación:

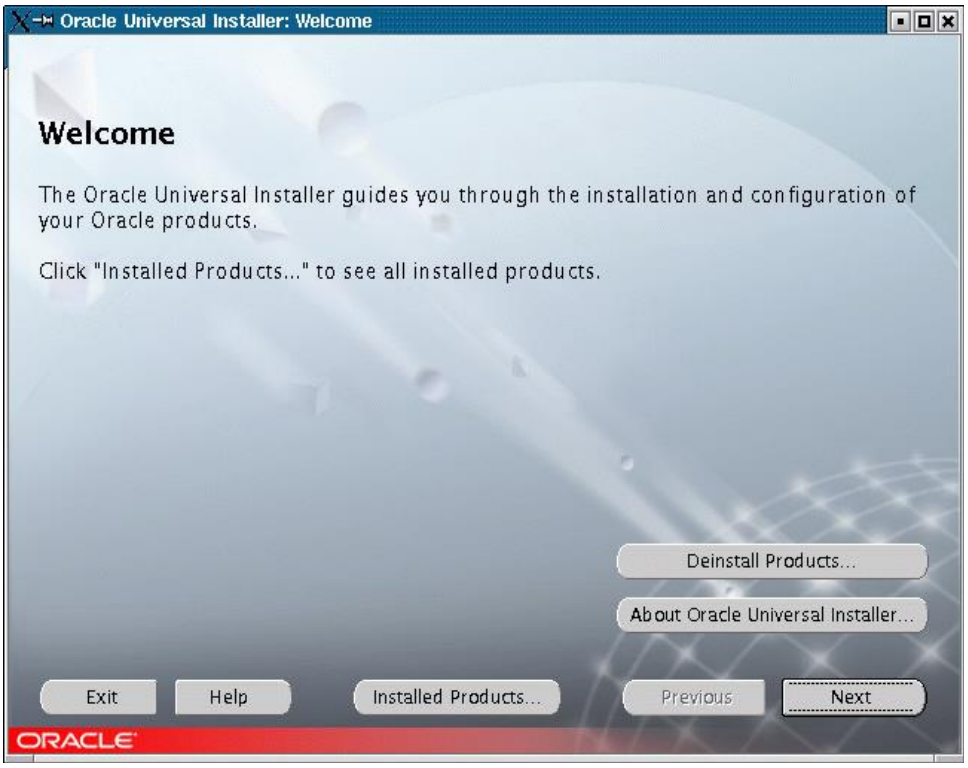


Figura 5.18 La ventana de bienvenida de Oracle 9iRAC (tomado de [2])

Luego de presionar el botón NEXT se presentaron un conjunto de pantallas donde se pedía definir los valores para Oracle. La primera de ellas hacía referencia al directorio raíz de Oracle que ya lo habíamos definido previamente como ***/oracle***; en la segunda definimos el directorio de inventarios en ***/oracle/orainventory***.

Después se definió el grupo de trabajo oracle como **dba** y luego procedimos a la definición de los nodos participantes en el cluster GPFS y para el Oracle 9iRAC. En la

lista de nodos participantes ya aparecían los nodos previamente creados con GPFS ngpfs01, ngpfs02 y ngpfs03. En cada uno de estos nodos se definió el lugar donde iban a residir los archivos que en nuestro caso fue **/oracle/product/9.2.0**.

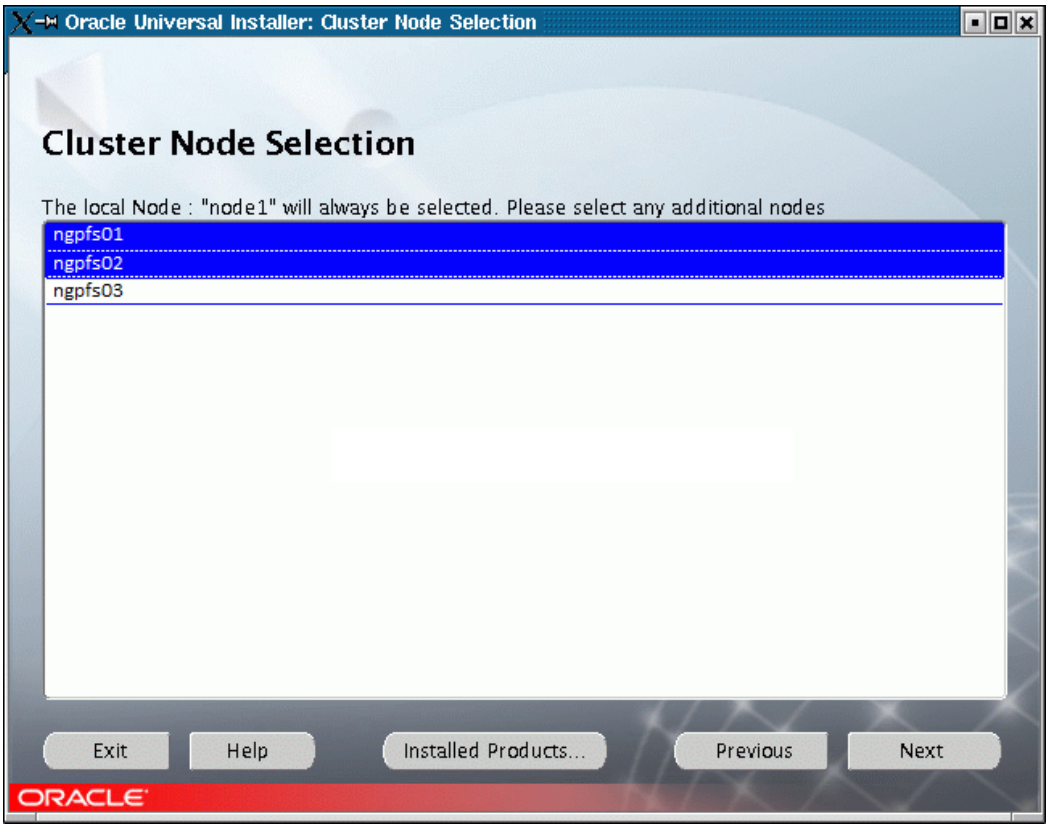


Figura 5.19 La ventana para seleccionar nodos participantes (tomado de [2])

Una vez que se había elegido el respectivo ambiente para Oracle, se procedió a seleccionar los productos disponibles; elegimos instalar la base de datos 9.2.0.1.0 y en una nueva pantalla seleccionamos el tipo de instalación de entre:

Enterprise Edition

Standard Edition

Custom. ←

Elegimos Custom, para poder seleccionar los componentes de software de la base de datos que aparecen en la siguiente pantalla, asegurándonos de elegir al menos

Oracle9i 9.2.0.1 ←

Oracle9i Application Cluster 9.2.0.1 ←

Luego de definir donde se deberían instalar los componentes de software elegidos, se procedió a entregar privilegios de sistema operativo AIX para el grupo **dba** de Oracle, para que de esta manera el Administrador de base de datos pudiese ejecutar tareas de respaldo, creación de archivos o accesar a ciertos directorios de sistema operativo; tareas que implicaban tener ciertos privilegios de AIX. Finalmente, se mostraba una pantalla final donde se hacía un sumario de los valores seleccionados previa a la instalación, para finalmente proceder a la instalación de la base de datos en forma paralela en todos los nodos del cluster.

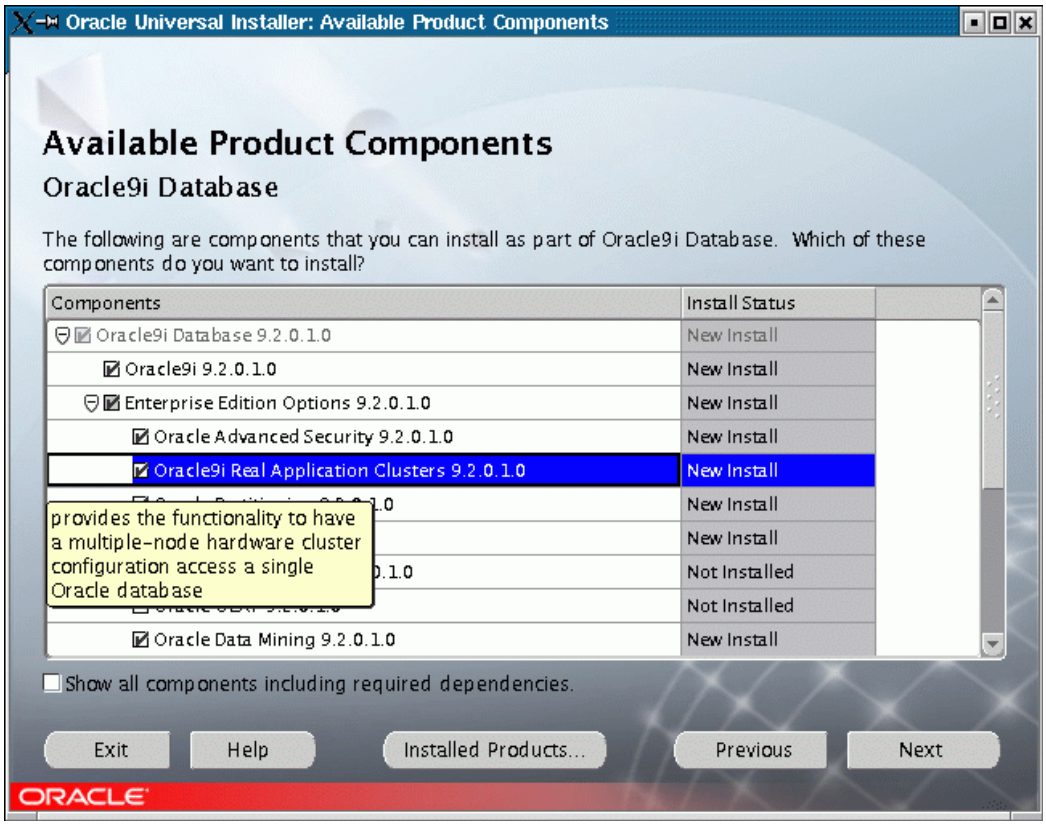


Figura 5.20 Selección de los componentes Oracle RAC (tomado de [2])

El instalador universal de Oracle9i RAC creaba un archivo de registro de la sesión de instalación en el directorio $ORACLE_BASE/oraInventory y se podía comprobar los mensajes de instalación como posibles errores ejecutando el comando

tail-f /oracle/oraInventory/logs/installactions.log

En vista a que esta instalación fue ejecutada en todos los nodos al mismo tiempo, tomaba más de 45 minutos en instalarse, pero al final de la misma se podía revisar el estatus de la misma la sentencia SQL:

sqlplus (sqlplus "/ as sysdba")

Si el "SQL>" aparecía al final de la ejecución de este comando, significaba que la instalación se había realizado correctamente.

Luego de la instalación, se hizo necesario actualizar parches de Oracle para tener la versión 9.2.0.3 en vez de la 9.2.0.1. Finalmente el DBA tuvo que empezar la configuración inicial de los servicios de Oracle Net para crear los componentes de red básicos, para permitir a los clientes y usuarios conectarse a las instancias de base de datos.

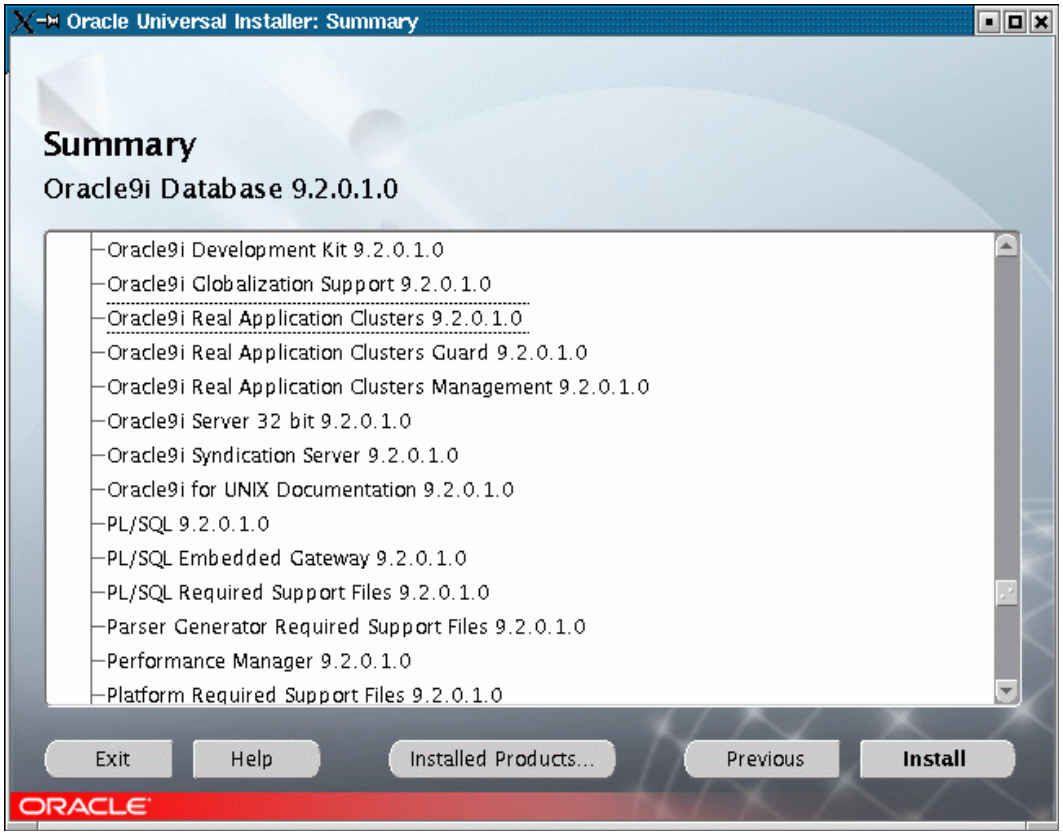


Figura 5.21 Valores seleccionados previa la instalación (tomado de [2])

Una de las tareas decisivas del DBA, fue la planeación del almacenamiento de las diferentes áreas de almacenamiento para Oracle. A nivel de AIX, nosotros creamos tres

filesystems de 213GB, 142GB y 71GB cuyos puntos de montajes eran /dbdata01, /dbdata02 y /dbdata03 respectivamente. Sobre estas definiciones se crearon las bases de datos Oracle. Estaba claro que la limitante del tamaño máximo de una base o conjunto de bases, estaba dado por el tamaño del filesystem; era posible que las bases reservasen todo el tamaño pero estar sin datos.

5.4. Instalación del HACMP para eliminación de puntos de falla

En vista a que HACMP fue instalado en únicamente en los dos servidores de aplicaciones, utilizamos el utilitario de configuración **xclconfig**, que era una aplicación de X-Windows que simplificaba la tarea de configurar un clúster HACMP de dos nodos. Con esta herramienta pudimos automatizar una de las cinco configuraciones predefinidas para un clúster de dos nodos.

Tal como se mencionó en el apartado 3.2.2, definimos que el tipo de configuración del cluster a utilizar era cascada, en donde el nodo con la más alta prioridad controlaba el grupo. Además se seleccionó el recurso tipo *mutual takeover* con un arreglo de recursos en cascada con un conjunto de nodos teniendo la misma prioridad.

Antes de invocar el utilitario de configuración rápida, habíamos completado las tareas de requisitos previos y que ambos nodos del clúster permitían acceso **root** remoto para el configurador. Para invocar el utilitario utilizamos el comando xclconfig, el cual mostraba la siguiente pantalla:

Figura 5.22 La pantalla de configuración HACMP (tomado de [8])

Al presionar el botón continuar, se mostraba una pantalla donde se pedía seleccionar la respectiva configuración de cluster, que para nuestro caso era del tipo cascada. Luego de esto, procedimos a definir los diferente roles de cada uno de los adaptadores del sistema y su respectivo adaptador de respaldo para poder solventar posibles caídas en uno de ellos. Para cada red, definimos un nombre único de adaptador, el tipo (ethernet o rs232) y el atributo (público, privado y serial) tal cual se definió en la tabla 7 del capítulo 4.3.

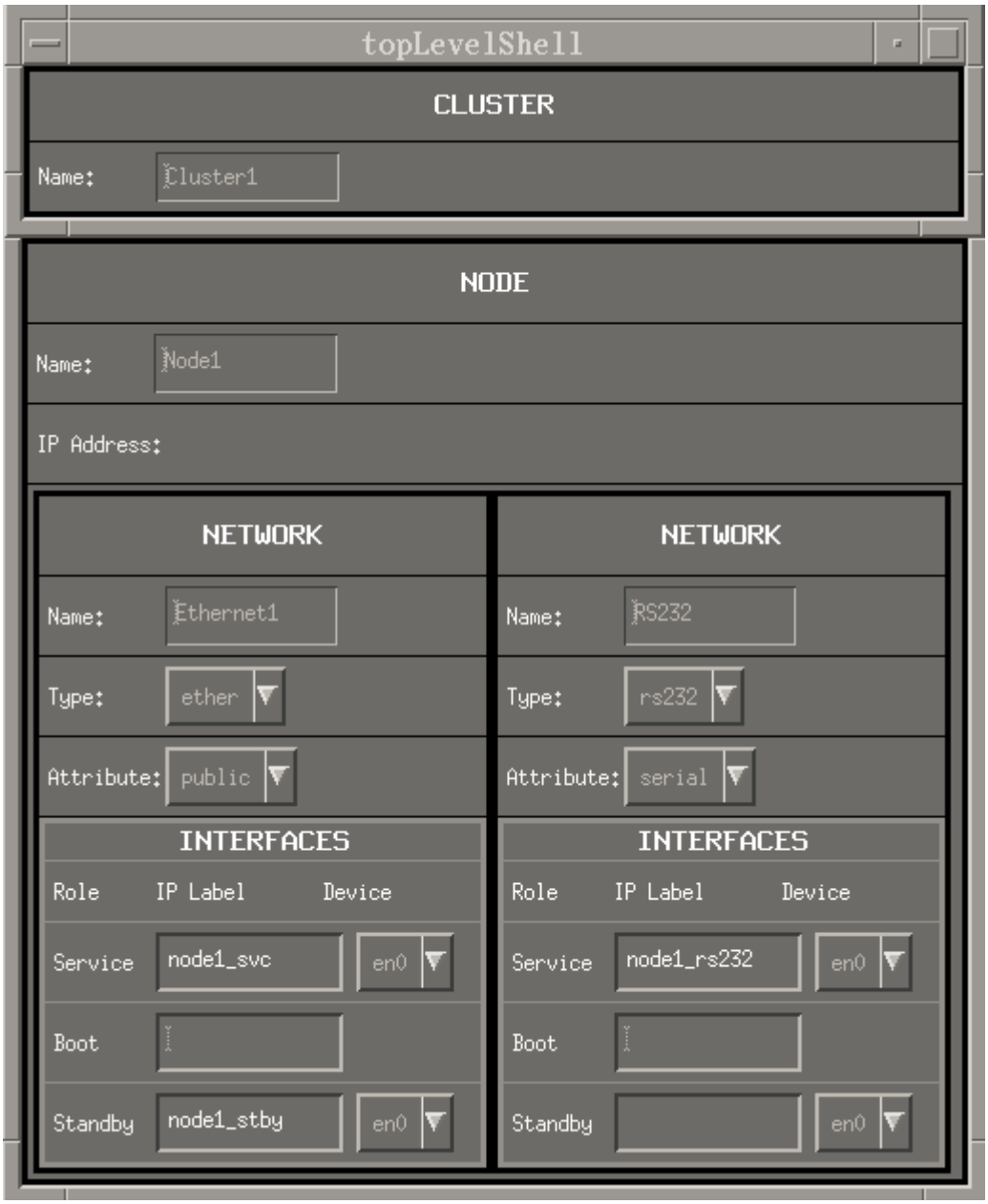


Figura 5.23 Definición de atributos para cada interface de red (tomado de [8])

Una vez definidos los atributos, se procedió a definir los tipos de recursos, con el nombre de los nodos participantes y la etiqueta de definió la dirección IP de los adaptadores que van a estar en producción. Para nuestro caso, no utilizamos ninguno

de los recursos compartidos como filesystems, VGs ni aplicaciones tal cual se muestra en la gráfica a continuación:

Figura 5.24 Definición de recursos compartidos (tomado de [8])

Con esto, se procedió a definir las direcciones IP para cada uno de los adaptadores de servicio como de boot en ambos servidores de aplicaciones. Con estos pasos, HACMP controlaba y eliminaba los siguientes puntos de fallas:

Posible falla en el adaptador de servicio

Posible falla en uno de los servidores pSeries

Posible falla de un disco de sistema operativo.

Dejando a las conexiones seriales rs232 punto a punto, como verificador de última instancia para determinar si un nodo estaba caído y determinar si necesitaba ser

remplazado por el segundo. Si eso llegaba a ocurrir, el nodo sobreviviente debía acarrear la carga del nodo caído.

5.5. Instalación de parches del software disponible

HACMP necesitaba que se le instale el parche U487607, mientras que GPFS los siguientes: IY36782, IY37744, IY37746, IY36626. Finalmente, a nivel del software de base de datos Oracle los siguientes parches: U488744, U488745, U488746, U488747. Adicionalmente, se podía obtener el paquete de mantenimiento AIX 5L 5200-03, el cual incluía todos los parches arriba mencionados y otros propios de sistema operativo, los mismos que estaban disponibles en la web y podían ser bajados a un computador en formato archivo comprimido o en formato CD o DVD. Para instalar los parches se ejecutó el siguiente comando: smitty update_all

El mismo que mostraba una pantalla del SMIT (System Management Interface Tool)

```
Update Installed Software to Latest Level (Update All)

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
[Entry Fields]
* INPUT device / directory for software                /dev/cd0
* SOFTWARE to update                                   _update_all
PREVIEW only? (update operation will NOT occur)       no +
COMMIT software updates?                              no +
SAVE replaced files?                                  no +
AUTOMATICALLY install requisite software?             yes+
EXTEND file systems if space needed?                  yes+
VERIFY install and check file sizes?                  no +
DETAILED output?                                      no +
Process multiple volumes?                             yes+
ACCEPT new license agreements?                        yes+
Preview new LICENSE agreements?                       no +_______
```

Figura 5.25 Pantalla del SMIT para instalar parches de sistema operativo

Los campos con “*” son obligatorios y el signo “+” implicaba que habían más opciones disponibles. Era recomendable no hacer un COMMIT del nuevo parche hasta después de un tiempo prudencial de prueba, caso contraria pueda ser eliminado. Finalmente, se podía hacer una verificación de la instalación con el comando **lppchk –v.**

CAPÍTULO 6

6. EJECUCIÓN DE PRUEBAS Y RESULTADOS

En este capítulo se presentan los resultados de las pruebas descritas en la sección 4.5 y que se realizaron con el fin de verificar el funcionamiento de la alta disponibilidad de algunos componentes críticos de la solución presentada a banco DelBank. Estos resultados incluían la conexión en tiempo de balanceo de carga y failover, así como la conmutación de la aplicación luego del error.

Todas la configuraciones de los equipos y del software de sistema operativo, GPFS y Oracle fueron configurados adecuadamente con el fin de producir los resultados correctos.

También consideramos el efecto que estas fallas podrían ocasionar al cliente en un futuro cuando las maquinas estuviesen en producción y recomendamos la contratación de servicios de software para prevenir o corregir potenciales problemas en cualquiera de los componentes de la plataforma, tanto a nivel de sistema operativo, hardware o base de datos.

Las pruebas ejecutadas en este capítulo pudieron llevarse a cabo de diferente manera y lograr el mismo resultado, pero estaba claro que con estas pruebas, cumplíamos con los requisitos que simulaban un ambiente real de contingencia.

6.1. Simulación de fallas de adaptadores de la red interna

En esta sección se analizan algunas condiciones de error relacionados con la interconexión del clúster utilizado por Oracle9i RAC y por GPFS y la comunicación entre instancias de Oracle.

Aquí se puede observar la salida de los diferentes comandos dentro de los scripts creados anteriormente y con los cuales podremos comprobar cómo se produce la recuperación de instancias una vez que la interface de red asociada con la interconexión del clúster falla y como el tráfico cambia a la interconexión secundaria. Además se puede comprobar cuál es el comportamiento de la sesión cuando su interface falla y como sigue activa una vez que se interconecta a la red secundaria.

6.1.1. Pruebas con los adaptadores de la red interna GPFS

Esta simulación se llevó a cabo generando consultas sobre la tabla DATA descrita en la sección 5.1 En nuestra configuración, la red primaria para la interconexión de GPFS utilizaba la interface de red **en3** y la secundaria la interface **en2** que también hacia de red de contingencia del InterConnect de Oracle. Ambas redes estaban etiquetadas como "privadas" tal cual se muestra en la tabla 7.

En la figura 4.19 de la sección 4.7.1, se muestra el shell script **database_access.sh** que utilizamos para demostrar que una sesión de cliente corriendo un query, no es afectada por la falla de la red primaria del GPFS y añadimos en la sección de inicio del código de falla, las siguientes líneas para la simulación correspondiente:

```
echo "$(date) Shutting down GPFS interface ..."
rsh deldb01 -n "echo \"$(date) $(chdev -l en3 -a state=down) ...\" "
rsh deldb01 -n "echo \"$(date) $(df -kt \" "
```

Figura 6.1 Script que simula caída del adaptador en3 de red primaria GPFS

En la sección de inicio de restauración de la falla añadimos lo siguiente:

```
echo "$(date) Starting up GPFS interface ..."
rsh deldb01 -n "echo \"$(date) $(chdev -l en3 -a state=up) ...\" "
echo "$(date) Attention: please clean up processes and ipc ..."
echo "$(date) GPFS NOT started automatically ..."
echo "$(date) Listener NOT started automatically ..."
echo "$(date) Instance NOT started automatically .."
```

Figura 6.2 Código que simula la restauración del adaptador en3

Al momento de la caída de la red primaria de cluster GPFS, por unos segundos los filesystems del sistema GPFS (/dbdata01, /dbdata02 y /dbdata03) no estaban disponibles, pero inmediatamente sucede la conmutación a la tarjeta en2, recuperando los filesystems antes mencionados. La salida del programa ejecutándose se muestra a continuación:

```
Mon Jun 9 18:52:47 EDT 2004 Test will connect to rac1f ...
Mon Jun 9 18:52:47 EDT 2004 Starting test ...
Mon Jun 9 18:52:47 EDT 2004 Starting query ...
Mon Jun 9 18:52:47 EDT 2004 Waiting for 30 seconds ...
Mon Jun 9 18:53:17 EDT 2004 Shutting down GPFS interface ...
Mon Jun 9 18:53:17 EDT 2004 en3 changed ...
Mon Jun 9 18:53:20 EDT 2004 en2 changed ...
Mon Jun 9 18:56:22 EDT 2004 Starting up GPFS interface ...

Mon Jun 9 18:53:27 EDT 2004

Filesystem    1024-blocks      Used      Free %Used Mounted on
/dev/hd4           131072     26928    104144   21% /
/dev/hd2          1703936   1211216    492720   72% /usr
/dev/hd9var        131072     56520     74552   44% /var
/dev/hd3          1310720    111780   1198940    9% /tmp
/dev/hd1           131072     81032     50040   62% /home
/proc                   -         -         -    -  /proc
/dev/hd10opt       131072     11208    119864    9% /opt
/dev/parche       4587520   1617900   2969620   36% /parche
/dev/delbank73    6291456   1981724   4309732   32% /delbank73
/dev/oralv       10223616   9458728    764888   93% /oracle
/dev/appslv       6291456   1306868   4984588   21% /apps
```

```
Mon Jun 9 18:53:27 EDT 2004 Waiting for query to finish ...
Mon Jun 9 18:56:34 EDT 2004 Query finished …

SQL> @time_instance

CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
09-JUN-03 06.56.49.787107 PM -04:00 rac1
SQL> select /*+ noparallel */ count(*) from data;

COUNT(*)
----------
5200000

SQL> @time_instance
CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
09-JUN-03 06.56.51.361454 PM -04:00 rac2
SQL> spool off
Mon Jun 9 18:56:52 EDT 2004 Starting up GPFS interface ...
Mon Jun 9 18:56:52 EDT 2004 en3 changed ...
Mon Jun 9 18:56:52 EDT 2004 GPFS NOT started automatically ...
Mon Jun 9 18:56:52 EDT 2004 Listener NOT started automatically ...
Mon Jun 9 18:56:52 EDT 2004 Instance NOT started automatically ...
Mon Jun 9 18:56:52 EDT 2004 Test finished ...
```

Figura 6.3 Simulación de la caída del adaptador en3 del GPFS

Resultado de la prueba:

El resultado de la prueba era el correcto y como se esperaba, Oracle9i se bloqueó por pocos segundos cuando sus datos almacenados en los filesystems GPFS estaban inaccesibles. Inmediatamente, el sistema de archivos GPFS se recuperó montando los filesystems a través de la red que utilizaba el adaptador en2. La consulta que empezó en el nodo ngpfs01 (rac1) terminó en la instancia rac2 localizada en el nodo ngpfs02. Una vez que la interface de red en3 se reconectó, en2 automáticamente entregó el control de la red GPFS al en3.

6.1.2. Pruebas con los adaptadores de red interna de Oracle 9iRA

Para esta prueba se generó carga en todas las instancias por medio del envío de consultas SQL contra la base de datos, con el fin de generar tráfico sobre la red del

InterConnect de Oracle. En nuestra configuración, la red primaria para la interconexión utilizaba la interface de red **en1** y la segunda red utiliza la interface **en2**. Ambas redes estaban etiquetadas como "privadas" tal cual se muestra en la tabla 7. Con todas las interfaces con tráfico entre ellas, procedimos a observar la carga de red con el comando **topas**, como se muestra en los siguientes ejemplos:

```
Network       KBPS   I-Pack   O-Pack    KB-In    KB-Out
en1         232.8      686      672    610.0     554.0   ← red primaria ON
en0          78.2     1929     1318    192.0     199.0
en3          51.6      423      457     54.0     204.0
lo0           2.8      127      127      7.0       7.0
en2           1.0       15       17      2.0       3.0   ← red respaldo OFF
```

Figura 6.4 Red con carga y todas las redes conectadas

Tras la desconexión del cable de red del adaptador **en1** el tráfico del InterConnect cambió a la red secundaria **en2**, como se muestra en el ejemplo siguiente:

```
Network       KBPS   I-Pack   O-Pack    KB-In    KB-Out
en2         344.2      945     1073    778.0     943.0   ← red respaldo ON
en0          89.0     2198     1555    221.0     224.0   ← red primaria GPFS
en3          89.0     1052     1098    138.0     307.0
lo0           2.8      133      133      7.0       7.0
en1           0.0        0        0      0.0       0.0   ← red primaria OFF
```

Figura 6.5 Red con carga pero con red primaria desconectada

Si luego de desconectar el cable de **en1** también desconectamos el cable en **en2**, el InterConnect de Oracle se conectó a otra red interna disponible, la del GPFS cuyo adaptador es el **en0.** Esto queria decir que la red primaria del GPFS absorbía todo el tráfico de las dos redes: Oracle y GPFS.

```
Network       KBPS   I-Pack   O-Pack    KB-In    KB-Out
en0         295.4     1912     1662    648.1     829.1   ← red GPFS primaria
en3          59.4      574      612     73.0     224.0
lo0           1.8       92       92      5.0       4.0
```

```
en1            0.0         0          0      0.0       0.0   ← red primaria OFF
en2            0.0         0          0      0.0       0.0   ← red respaldo OFF
```

Figura 6.6 Red con carga pero con redes del InterConnect desconectadas

Al reconectar el cable rj-45 en el adaptador **en1**, el tráfico de la InterConnect se revertía y regresaba a esta red:

```
Network       KBPS   I-Pack   O-Pack   KB-In   KB-Out
en1          216.2      578      607   522.0    559.0   ← red primaria ON
en3           58.0      589      624    75.0    215.0
en0           55.6     1340      891   136.0    142.0
lo0            1.2       67       67     3.0      3.0
en2            0.0        0        0     0.0      0.0   ← red respaldo OFF
```

Figura 6.7 Red con carga con la red secundaria desconectada

Si se reconectaba el cable en en2, la red secundaria del Oracle regresaba a la red en forma activa pero no primaria.

```
Network       KBPS   I-Pack   O-Pack   KB-In   KB-Out
en1          205.2      544      622   463.0    563.0   ←red primaria ON
en0           65.4     1552     1031   159.0    168.0
en3           49.4      442      460    59.0    188.0
lo0           14.0      429      429    35.0     35.0
en2            1.0       19       18     3.0      2.0   ← red respaldo ON
```

Figura 6.8 Red con carga y con todas las redes conectadas luego de prueba

Comportamiento de sesión durante la simulación

Una vez más utilizamos el shell script **database_access.sh** de la figura 4.19 para demostrar que una sesión de cliente realizando consultas, no es afectada por la falla de la red primaria del InterConnect. Tal cual se indicó en esa sección, utilizamos una tabla DATA que contenía 5.2 millones de filas y ejecutamos búsquedas que tardaban aproximadamente 100 segundos.

En esa misma sección, se indicó que el código del script que simulaba la falla, variaba dependiendo del tipo de recurso. En este caso, como estamos simulando la caída de un adaptador, el código utilizado fue el siguiente:

```
echo "$(date) Shutting down primary interconnect ..."
rsh deldb01 -n "echo \"$(date) $(chdev -l en1 -a state=down) ...\" "
```

Figura 6.9 Código de simulación de falla de adaptador en1

Así mismo, el código para restaurar el adaptador caído es el siguiente:

```
echo "$(date) Starting up primary interconnect ..."
rsh deldb01 -n "echo \"$(date) $(chdev -l en1 -a state=up) ...\" "
```

Figura 6.10 Código de simulación para restaurar adaptador en1

Una vez que corrimos el mencionado script desde un cliente, con las respectivas caídas o restauraciones podemos observar la siguiente salida:

```
Tue Jun 10 15:40:14 EDT 2004 Test will connect to rac1f ...
Tue Jun 10 15:40:14 EDT 2004 Starting test ...
Tue Jun 10 15:40:14 EDT 2004 Starting query ...
Tue Jun 10 15:40:14 EDT 2004 Waiting for 30 seconds ...
Tue Jun 10 15:40:44 EDT 2004 Shutting down primary interconnect
Tue Jun 10 15:40:44 EDT 2004 en1 changed ...
Tue Jun 10 15:40:44 EDT 2004 Waiting for query to finish ...
Tue Jun 10 15:42:18 EDT 2004 Query finished ...
SQL> @time_instance
CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
10-JUN-03 03.40.14.832135 PM -04:00 rac1
SQL> select /*+ noparallel */ count(*) from data;
COUNT(*)
----------
5200000
SQL> @time_instance

CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
10-JUN-03 03.42.18.243060 PM -04:00 rac1
SQL> spool off
Tue Jun 10 15:42:18 EDT 2004 Starting up primary interconnect ...
Tue Jun 10 15:42:18 EDT 2004 en1 changed ...
Tue Jun 10 15:42:18 EDT 2004 Test finished ...
```

Figura 6.11 Resultado de la prueba de caída de un adaptador en1

Resultado de la prueba:

Como se puede ver en los resultados, la consulta ha terminado en la misma instancia y en el mismo nodo donde se inició, pues no existió cambio de nodo sino solo de adaptador, entregando el resultado correcto. La diferencia de tiempo de ejecución desde el inicio de la caída del adaptador en1 es de aproximadamente 1 minuto con 34 segundos y la conmutación por error del InterConnect fue cuestión de unos pocos segundos.

6.2. Simulación de fallas de adaptadores de la red de usuarios

Para esta prueba se deshabilitó la interface de red en0 en el nodo de aplicaciones delsrv02. Un usuario ejecutaba la consulta en una instancia desde ese nodo. Luego desconectamos físicamente el cable del adaptador en0 para simular la falla de caída del adaptador de red.

En vista que en los nodos de aplicaciones existían dos tarjetas de red de 10/100 Mbps (en0 y en1) incluidas en el mainboard de los pSeries, HACMP manejaba la conmutación entre los dos adaptadores rápidamente (en cuestión de uno o dos segundos), restaurando las sesiones de usuarios casi inmediatamente sin causar mayor problema a las consultas hechas a las instancias de la base de datos. En base a esto, una prueba válida implicaba trabajar sólo con un adaptador verificando que la consulta finalice sin importar el tiempo; elegimos utilizar el en0 y desconectamos físicamente el cable rj45 del en1, para comprobar que las consultas esperasen el tiempo necesario hasta que en0 vuelva a recuperarse, es decir no tuvimos dos adaptadores disponibles para esta prueba. Utilizamos dos pruebas, con sesión inactiva y activa.

Prueba con sesión inactiva:

Tal como se indicó en la sección 4.5.1, para la realización de las pruebas se utilizaron sesiones de usuarios inactivas y activas. La primera prueba que hicimos fue con sesión inactiva y se quería verificar si la sesión podía volver a conectarse a otra instancia y si era posible realizar consultas desde la misma sesión después de la falla. En vez de utilizar un query, utilizamos el comando de AIX **sleep 120** dentro del programa **database_access.sh**, el mismo que al correrlo, obtuvimos los siguientes resultados:

```
_____________________________________________________
Thu Jun 5 11:29:30 EDT 2004 Test will connect to rac1f ...
Thu Jun 5 11:29:30 EDT 2004 Starting test ...
Thu Jun 5 11:29:30 EDT 2004 Starting query ...
Thu Jun 5 11:29:30 EDT 2004 Waiting for 45 seconds ...
Thu Jun 5 11:30:15 EDT 2004 Shutting down interface ...
Thu Jun 5 11:30:15 EDT 2004 en0 changed ...
Thu Jun 5 11:30:16 EDT 2004 Waiting for query to finish ...
Thu Jun 5 11:43:56 EDT 2004 Query finished ...
SQL> @time_instance
CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
05-JUN-03 11.29.30.845725 AM -04:00 rac1
SQL> !sleep 120
SQL> @time_instance
CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
05-JUN-03 11.43.56.724708 AM -04:00 rac2
SQL> spool off
Thu Jun 5 11:43:56 EDT 2004 Starting up interface ...
Thu Jun 5 11:43:56 EDT 2004 en0 changed ...
Thu Jun 5 11:43:57 EDT 2004 Test finished ...
_____________________________________________________
```

Figura 6.12 Simulación por falla de la red de usuarios. Modo inactivo

Resultado de la prueba con sesión Inactiva:

De acuerdo con la salida del ejemplo anterior, la sesión del **sqlplus** inicia en la instancia rac1 en el nodo ngfps01 y es reconectada en la rac2 en ngpfs02. El total de tiempo recorrido del query fue de 14 minutos en vez de 2 minutos que el comando sleep 120 podría necesitar si hubiese otro adaptador en1. Esta diferencia fue debido a los periodos de tiempos de espera del TCP/IP pues la reconexión del cable fue manual.

Prueba con sesión Activa:

Esta simulación se la hizo ejecutando una consulta mientras la interface de red del nodo donde residía la instancia que procesaba la mencionada consulta, era desconectada o deshabilitada.

```
Thu Jun 12 16:47:31 EDT 2004 Test will connect to rac1f ...
Thu Jun 12 16:47:31 EDT 2004 Starting test ...
Thu Jun 12 16:47:31 EDT 2004 Starting query ...
Thu Jun 12 16:47:31 EDT 2004 Waiting for 30 seconds ...
Thu Jun 12 16:48:01 EDT 2004 Shutting down interface ...
Thu Jun 12 16:48:01 EDT 2004 en0 changed ...
Thu Jun 12 16:48:01 EDT 2004 Waiting for query to finish ...
Thu Jun 12 17:02:05 EDT 2004 Starting up interface ...
Thu Jun 12 17:02:05 EDT 2004 en0 changed ...

Thu Jun 12 17:02:05 EDT 2004 Query finished ...
SQL> @time_instance

CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
12-JUN-03 04.47.31.311525 PM -04:00 rac1
SQL> select count(*) from
2 ( select * from dba_source
3 union
4 select * from dba_source
5 union
6 select * from dba_source
7 union
8 select * from dba_source
9 union
10 select * from dba_source);
select count(*) from
*
ERROR at line 1:
ORA-03113: end-of-file on communication channel

SQL> @time_instance
CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
12-JUN-03 05.02.05.550570 PM -04:00 rac2

SQL> spool off
Thu Jun 12 17:02:06 EDT 2004 Test finished ...
```

Figura 6.13 Simulación por falla en la red de usuarios. Modo Activo

Resultado de la Prueba:

La consulta que se ejecutó no es completada si la interface de red en0 del usuario falla, pues para estas pruebas, en4 fue desconectada previamente. La sesión sin embargo, es reconectada a la siguiente instancia rac2 y continúa luego de que el adaptador en0.

6.3. Simulación de caídas de nodos del cluster

En esta prueba, se simulaba la caída del nodo ngpfs01 mientras la instancia ejecutaba la respectiva consulta de prueba. Los códigos de fallo y reparación se muestran a continuación:

```
# --- failing full node test
# --- code to be included in test script
#
# --- failure code:
#
echo "$(date) Failing node ..."
rsh deldb01 -n "touch trigger"
#
# --- repair code:
#
echo "$(date) Attention: node halted ..."
```

Figura 6.14 Script utilizado para hacer fallar un nodo

En vez de apagar directamente el nodo utilizando el comando **rsh shutdown**, optamos por colocar un archivo de activación en el nodo que iba a fallar. En este nodo se creó un pequeño lazo infinito que esperaba bucle que esperaba a que el archivo (que lo nombré **/trigger**) de activación apareciese para detener el nodo.

```
#!/bin/ksh
while sleep 2
do
if [ -f /trigger ]
then
  rm -f /trigger
  $1
```

```
  break
else
  print ".\c"
fi
done
```

Figura 6.15 Lazo infinito corriendo en el nodo a fallar

El script tomaba una cadena de caracteres como argumento y lo ejecuta como un comando desde el shell tan pronto el archivo llamado /trigger sea encontrado en el directorio raíz. El script se llama **wait_for_trigger** y lo ejecutábamos el comando **wait_for_trigger "halt -q"**, para que el nodo se detenga tan pronto como el archivo /trigger apareciera en el nodo. Con el fin de acortar el tiempo de conmutación por error de TCP/IP, se estableció el parámetro **tcp_keepidle = 240**, equivalente a un período de tiempo de espera de 2 minutos en vez del valor predeterminado de 14400 que correspondía a 2 horas, lo cual era demasiado espera para una prueba. La salida del query utilizando el script **database_access.sh** con las modificaciones correspondientes para simular la caída de un nodo, se muestran a continuación:

```
Thu Jun 12 16:17:50 EDT 2004 Test will connect to rac1f ...
Thu Jun 12 16:17:50 EDT 2004 Starting test ...
Thu Jun 12 16:17:50 EDT 2004 Starting query ...
Thu Jun 12 16:17:50 EDT 2004 Waiting for 30 seconds ...
Thu Jun 12 16:18:19 EDT 2004 Failing node ...
Thu Jun 12 16:18:19 EDT 2004 Waiting for query to finish ...
Thu Jun 12 16:21:12 EDT 2004 Query finished ...
SQL> @time_instance
CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
12-JUN-03 04.17.49.933615 PM -04:00 rac1
SQL> select count(*) from
2 ( select * from dba_source
3 unión
4 select * from dba_source
5 union
6 select * from dba_source
7 union
8 select * from dba_source
9 union
10 select * from dba_source);
COUNT(*)
----------
119662
SQL> @time_instance
CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
```

```
12-JUN-03 04.21.12.427132 PM -04:00 rac2
SQL> spool off
Thu Jun 12 16:21:12 EDT 2004 Attention: node halted ...
Thu Jun 12 16:21:12 EDT 2004 Test finished ...
```

Figura 6.16 Salida de la consulta cuando un nodo fallaba

Resultados de la prueba:

Esta prueba también produjo los resultados correctos tal cual se esperaba. Las consultas finalizaron sin errores y la sesión continuaba en la siguiente instancia disponible.

6.4. Análisis de la disponibilidad de datos en nodos disponibles

Una vez realizadas las respectivas pruebas de disponibilidad de datos, se pudo comprobar que el comportamiento de los tipos de consultas inactivas y activas, se realizaron con los resultados esperados.

En general, con todas las pruebas exceptuando la simulación de una caída de la interface de usuario ocurrieron en tiempos de espera muy cortos, aceptados por el personal de Banco DelBank.

Todos los nodos estuvieron disponibles luego de una falla de algún recurso y los datos accesibles en todo momento. Un compendio que resume la disponibilidad de los datos y que detalla si las consultas realizadas fueron completadas satisfactoriamente o no, se muestran en la siguiente tabla:

Componente Fallando	Recurso Caído	Recurso de Remplazo	Consulta completada	Resultado Satisfactorio
Red interna GPFS	Ethernet en3	Ethernet en2	SI	SI
Red interna Oracle	Ethernet en1	Ethernet en2	SI	SI
Red de usuarios	Ethernet en0	Ethernet en1	SI	SI
Red de usuarios	Ethernet en0	Ninguno	SI luego de reconectar cable en en0	SI
Nodo del cluster	ngpfs01	ngpfs02 ngpfs03	SI	SI

Tabla 12. Disponibilidad datos luego de las simulaciones

A nivel de software, pudimos comprobar lo que teóricamente se afirmaba:

- Que el failover o recuperación de la base de datos se la hizo sin pérdida de datos.
- Que las instancias sobre las cuales se recuperaron las consultas, pasaron a ser instancias primarias.
- Que la copia no requirió administración de la propia base de datos.
- La redundancia en CPU, componentes de red, fuentes de poder y almacenamiento de datos funcionó correctamente, asegurando la inversión Banco DelBank.
- Existió un mínimo impacto sobre la disponibilidad y desempeño de un sistema primario.
- Después de la falla, la solución habilitó un cambio de vuelta al sistema primario de manera automática.

A nivel de hardware:

- Pudimos comprobar la escalabilidad de un sistema tipo cluster, al agregarle nodos al clúster para un mejor desempeño.
- Debido a sus componentes redundantes, los cuales proveían servicio ininterrumpido, siempre en el evento de fallas de hardware, GPFS y Oracle9i RAC proveían la alta disponibilidad, permitiendo a los usuarios tener acceso a todo dato en donde haya un nodo disponible en un clúster. Esto significa que los datos estaban consistentemente disponibles.

6.4.1. Revisión de la información de la base de datos

Para la respectiva revisión y comprobación de los datos antes y después de la simulación de una falla de algún recurso en cada uno de los tres nodos de bases de datos, ejecutamos un query para verificar que el número de filas de la tabla DATA recuperadas, contenía el mismo número de filas, es decir deberían mostrarnos 5.2 millones. Una toma de datos antes de la falla, mostraba lo siguiente:

```
SQL> @time_instance

CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
11-JUN-03 03.33.14.832135 PM -04:00 rac1
SQL> select count(*) from data;
COUNT(*)
----------
5200000
SQL> @time_instance

CURRENT_TIMESTAMP                   INSTANCE_NAME
----------------------------------- ----------------
11-JUN-03 03.34.24.243060 PM -04:00 rac1
SQL> spool off
```

Figura 6.17 Query antes de la simulación de falla

El mismo query disparado desde un cliente luego de la falla de algún recurso mostraba:

```
SQL> @time_instance

CURRENT_TIMESTAMP                    INSTANCE_NAME
---------------------------------- ----------------
11-JUN-03 06.56.49.787107 PM -04:00 rac1
SQL> select /*+ noparallel */ count(*) from data;

COUNT(*)
----------
5200000

SQL> @time_instance
CURRENT_TIMESTAMP                    INSTANCE_NAME
---------------------------------- ----------------
11-JUN-03 06.59.58.361454 PM -04:00 rac2
```

Figura 6.18 Query después de la simulación de falla

En ambas prueba, se puede apreciar que la cantidad de filas contabilizadas fueron cinco millones doscientos mil. La única diferencia que existía fue en el tiempo, que por supuesto en los casos de las simulaciones era mayor.

Además, se verificó que el estado de la base de datos estaba operativo y con la instancia respectiva activa, con el comando:

```
SQL> select * from v$instance

INSTANCE_NUMBER  INSTANCE_NAME  HOST_NAME  VERSION  STARTUP_T  STATUS  PAR
---------------  -------------  ---------  -------  ---------  ------  ---
1                rac1           ngpfs01    9.2.0.3  21-JUN-03  OPEN    NO
```

Figura 6.19 Estado de la base de datos e instancias

Y Finalmente se verificó que la base de datos continuaba abierta luego de la simulación de alguna falla:

```
SQL> select status from v$instance

STATUS
------
OPEN
```

Figura 6.20 Estado de la base de datos

Existían muchos otros comandos de Oracle relacionados con las vistas y nos podían mostrar el estatus de la base de datos, pero en nuestro caso, estandarizamos la prueba con las tres pruebas anteriormente mostradas.

6.4.2. Comprobación de eliminación de puntos de falla

En las secciones anteriores se procedió a realizar un conjunto de tareas y procedimientos cuyo objetivo era eliminar puntos únicos de falla. Es decir eliminar cualquier componente individual integrado en el cluster que, en caso de fallar afectaba directamente la disponibilidad de la aplicación utilizada por los usuarios finales. Los puntos de falla que eliminamos por parte de IBM, fueron los acordados con el personal de Banco DelBank e incluían: los nodos del clúster, almacenamiento, adaptadores de red y redes.

Por parte del personal de Oracle, se configuró **Oracle9i RAC Cache Fusion**, el cual permitía que cualquier bloque de la base de datos que era mantenida por una instancia, pueda ser copiada a otra a través de la red interna de Oracle o InterConnect. Esto permitía que los datos sean compartidos entre diferentes instancias en nodos diferentes. La eliminación del punto de falla producido por una instancia, estaba a cargo de Oracle y fue transparente para nosotros, el proceso de recuperación de la base de datos. Nuestra responsabilidad recaía sobre las redes internas de Oracle, cuyos puntos de falla eliminamos con la instalación de adaptadores redundantes Ethernet Gigabit.

A nivel de GPFS, una falla de red en un nodo producía una falla en la instancia de la base de datos en ese nodo; al igual que en el caso anterior, para eliminar la red GPFS como un único punto de falla, utilizamos como redundancia, la red secundaria de Oracle con sus respectivos adaptadores.

A nivel de los tres nodos de la base de datos, comprobamos la caída de uno de los tres, logrando que los otros dos sean los nodos redundantes y soporten la carga hasta que el primero se recupere del estado de error y se reintegre al cluster.

A nivel de discos duros, tanto los internos de sistema operativo como los externos del arreglo FastT600, utilizamos espejamiento es decir RAID 1.

Los dos servidores de aplicaciones fueron configurados con HACMP donde cada uno era respaldo del otro y una caída de uno de ellos implicaba que el segundo soportaba las tareas y la carga hasta la reintegración del que falló. Igualmente, con los adaptadores de red para usuarios en estos dos nodos, HACMP controlaba fácilmente la falla de uno de ellos, pues cada servidor contaba con dos adaptadores de red ethernet.

A nivel de hardware, todos los nodos fueron configurados con fuentes de poder redundantes, eliminando así un posible punto de falla a este nivel.

Un detalle de las pruebas realizadas para la comprobación de que se eliminaron los puntos de falla solicitados por el personal de Banco DelBank se muestra a continuación.

Punto de Falla	Eliminado x recurso redundante	Utilizando	Recurso donde reside	Eliminado
Disco interno hdisk0	Disco interno hdisk1	Espejamiento por AIX	Todos los Nodos (5)	SI
Disco externo en el arreglo	Otro disco del arreglo	Espejamiento FastT600	Arreglo FastT600	SI
Instancia rac1	Instancia rac2 o rac3	Oracle Cache Fusion	Nodos base de datos (3)	SI
Adaptador en1	Adaptador en2	Oracle9i RAC	Nodos de base de datos (3)	SI
Adaptador en3	Adaptador en2	GPFS	Nodos base de datos (3)	SI
Adaptador en0	Adaptador en1	HACMP	Nodos de Aplicaciones (2)	SI
Adaptador en1	Adaptador en0	HACMP	Nodos de Aplicaciones (2)	SI
Nodo delsrv01	Nodo delsrv02	HACMP	Nodos de Aplicaciones (2)	SI
Nodo delsrv02	Nodo delsrv01	HACMP	Nodos de Aplicaciones (2)	SI
Nodo ngpfs01	ngpfs02 y ngpfs03	GPFS	Nodos bases de datos (3)	SI
Nodo ngpfs02	ngpfs01 y	GPFS	Nodos bases de datos (3)	SI

	ngpfs03			
Nodo ngpfs03	ngpfs02 y ngpfs01	GPFS	Nodos bases de datos (3)	SI
Fuente Poder	Otra fuente poder	Circuitos múltiples	Todos los Nodos (5)	SI

Tabla 13. Comprobación de eliminación de puntos de falla

6.5. Análisis de resultados

Una vez realizadas las pruebas de alta disponibilidad con los cinco servidores IBM pSeries, se pudo poner en práctica mucho de los conocimientos teóricos que se tenían hasta ese momento. Para el respectivo análisis de las pruebas ejecutadas, he dividido los resultados obtenidos de las mismas en tres diferentes áreas: hardware, software operacional y base de datos.

Análisis de resultados de Hardware:

Utilizamos recursos de hardware redundantes en las partes más críticas del sistema:

- Discos duros con sistema operativo y discos duros de la base de datos.
- Fuentes de poder
- Servidores IBM pSeries
- Adaptadores de redes ethernet
- Redes internas

Las causas de fallas en cualquiera de estos recursos pudieron estar relacionadas a diferentes razones, como cortes de energía, fallas de red, fallas de materiales, ambientales o desastres naturales. Por lo que el uso de recursos redundantes se

complementó con la implementación de un disciplinado proceso de respaldos en caliente y en frio cuyos medios reposaban lejos del centro de cómputo.

Para todos estos casos, una falla de un recurso implicaba que el recurso redundante tomaba el lugar del principal de una forma transparente y en cuestión de segundos, dejando al Administrador del sistema una ventana para poder realizar el remplazo correspondiente sin tener que quitar el servicio a los usuarios. Si el recurso fallido estaba dentro del equipo servidor pSeries, implicaba apagarlo para realizar el cambio respectivo, dejando que el servidor de respaldo atienda los requerimientos de usuarios.

Si el recurso fallido era un disco dentro del subsistema FastT600 la conmutación era mucho más transparente, pues implicaba identificar el disco, removerlo y remplazarlo por el nuevo; todo esto en línea, sin apagarlos.

Para el caso de las redes internas, el hecho de contar con doble tarjetería y doble switch de comunicación, hacía mucho más fácil el remplazo y una conmutación entre redes, de una forma transparente.

Con lo anteriormente expuesto, se logró mantener un sistema disponible las 24 horas con tiempos de inactividad casi nulos, full tolerable a errores.

Análisis de Software:

Desde el punto de vista de mi rol asignado a este proyecto como Ingeniero de Sistemas AIX, las aplicaciones IBM con las cuales tuve que desenvolverme fueron validadas y justificadas por el cliente. Tanto el software de sistema operativo AIX, HACMP y GPFS cumplieron con todas las expectativas propuestas a inicio del proyecto.

AIX cumplió el rol de manejar la disponibilidad de los servidores, la administración de recursos y el espejamiento de discos internos. Las pruebas realizadas permitieron probar su desempeño cada vez que AIX tenía que resolver condiciones de fallas de los diferentes recursos (inclusive nodos) y decidir remplazarlo por otro. Probamos así que el kernel de sistema operativo era completamente dinámico y era el pilar que soportaba otras aplicaciones tanto IBM como no IBM.

Las aplicaciones para lograr la transparencia en el remplazo de recursos, trabajaron acorde a lo requerido. HACMP por el lado de los servidores de aplicación, siempre controló las pruebas de caídas de servidores, adaptadores y discos, manejándose acorde a lo requerido para mantener un sistema altamente disponible. Incluso en los casos extremos, donde se simulaba la caída de uno de los nodos de cluster HACMP podía manejar el servicio con un solo servidor y manejar efectivamente la reintegración del nodo fallido al cluster con la asignación de las direcciones de booteo en los adaptadores de red redundantes.

Por el lado de los servidores de base de datos, el problema inicial por la falta de quorum requerido por GPFS, se solucionó con la compra de un tercer nodo. La condición de falla que quedó abierta fue si fallasen dos de los tres servidores de base de datos, pero este inconveniente fue subsanado en el año 2005, con la compra del software IBM Storage Manager 8.4 por parte de Banco Delbank.

En vista que Oracle 9iRAC no garantizaba el failover transparente, pues sólo garantizaba la disponibilidad de la base de datos, GPFS fue la solución ideal para complementar la disponibilidad cubierta por Oracle 9iRAC. Con esto puedo indicar que todas las pruebas con GPFS fueron satisfactorias y con tiempos de recuperación de los

adaptadores, discos compartidos y servidores dentro del rango aceptado en pruebas estándar similares.

Finalmente el cache Fusion de Oracle9i RAC hizo la parte más importante a nivel de base de datos, permitiendo que las consultas de la base de datos mantenidas en una instancia de un nodo, pueda ser copiada a otra a través de la red interna InterConnect, logrando la compartición de datos entre instancias en diferentes nodos. Con esto se pudo terminar de hacer las consultas en otros servidores cuando el original fallaba.
El resultado final en general fue satisfactorio. Los Ingenieros de IBM con los de Oracle cumplimos con todas las pruebas requeridas para lograr la entrega de un proyecto terminado y aceptado por Banco Delbank.

6.5.1. Tiempos de recuperación en casos de contingencia

Los tiempos de recuperación en base a las pruebas anteriores se basaron en la simulación de falla de cuatro recursos:

- La red GPFS
- La red InterConnect de Oracle
- Un nodo
- Red de usuarios

De estas cuatro tipos de fallas, la de la red de usuarios fue un poco atípica debido a que la prueba se la hizo utilizando un solo adaptador de red pues lo que se buscaba era comprobar que la consulta SQL terminara.

Para la red GPFS, el tiempo de recuperación promedio una vez que el adaptador en3 fallaba era de **03m:17s** (3 minutos y 17 segundos); ese era el tiempo que le tomaba a GPFS tomar control de la red, conectarse al nuevo adaptador en2 y finalizar la consulta SQL enviada por un usuario.

Para la red de Oracle, el tiempo de recuperarse de una falla del InterConnect cuando el adaptador en1 fallaba, se conmutaba al en2 y terminaba la consulta SQL era de **01m:34s** (un minuto y treinta y cuatro segundos).
Para la caída de un nodo, el tiempo de recuperación era de **02m:53s** este valor implicaba el tiempo que le tomaba al query en reconectarse a otra instancia en otro nodo.

Para la red de usuarios, un tiempo de conmutación entre adaptadores era de aproximadamente 1,7 segundos; un query tomaba **01m:31s** (un minuto y treinta y un segundos) en terminar en la misma instancia. Pero como indicamos al inicio de esta prueba, había más significado en controlar la terminación del query en vez de los tiempos de respuesta. Habría que recordar que lo que estábamos desconectando era el cable ethernet de uno de los dos nodos de aplicaciones y no de base de datos.

Finalmente, para una referencia de cuanto le tomaba a uno de los servidores realizar la consulta de los 5.2 millones de tablas sin ninguna falla, el tiempo del query era de **01m:10s**, tal cual se muestra en el la salida de la figura 6.17 de la sección 6.4.1.

Con estos tiempos, se puede indicar que los tiempos de pruebas parten con consultas SQL que les toma 01m:10s en terminar sin ningún tipo de falla en

adaptadores o nodos. Este tiempo hay que restarle a los valores que tomaría una recuperación neta de los tres casos anteriores:

03m:17s – 01m:10s = **02m:07s** Tiempo en recuperarse en3 en en2 (GPFS)
01m:34s – 01m:10s = **00m:24s** Tiempo en recuperarse en1 en en2 (Oracle)
02m:53s – 01m:10s = **01m:43s** Tiempo en recuperarse el nodo ngpfs01

La razón por la cual la recuperación del InterConnect de Oracle sólo tomaba 24 segundos, es debido a que no hay pérdida del acceso a los datos localizados en los filesystems /dbdata01, /dbdata02 y /dbdata03. En cambio, para GPFS, la base de datos en los mencionados filesystems no estaba disponible y había que esperar mucho más hasta que los tres filesystems estén disponibles nuevamente pero a través del adaptador en2.

Para el caso de falla de un nodo, los clientes deberían esperar 1m:43s hasta que la consulta se recupere en otra instancia disponible, que obligatoriamente debía ser en otro nodo.

6.5.2. Comparación de tiempos de respuesta entre servidores

Para la última prueba con los tres servidores de base de datos, se realizaron diez consultas secuenciales de los cinco millones de registros para comprobar el rendimiento de los mismos. Aunque todas las pruebas se ejecutaron sin datos del cliente sino con programas provistos por Oracle y datos de prueba, era interesante observar el desempeño de los dos p630 versus el p615 de menor capacidad de

rendimiento. Los resultados de las pruebas se muestran en las tablas a continuación:

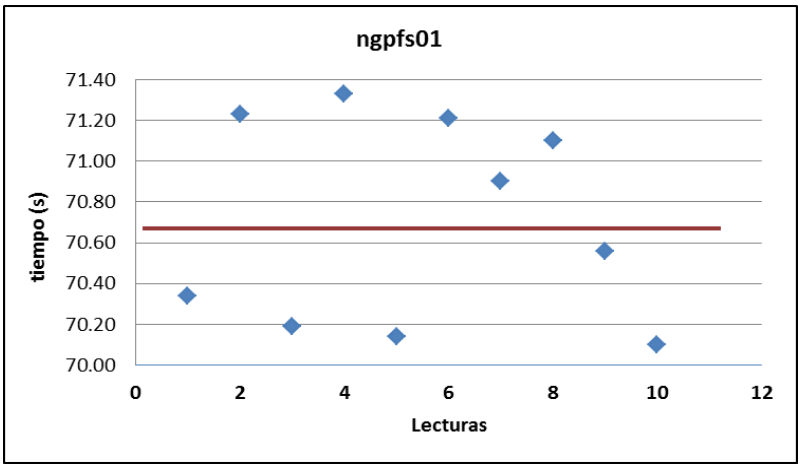


Figura 6.21 Tiempo de respuesta servidor pSeries 630 ngpfs01

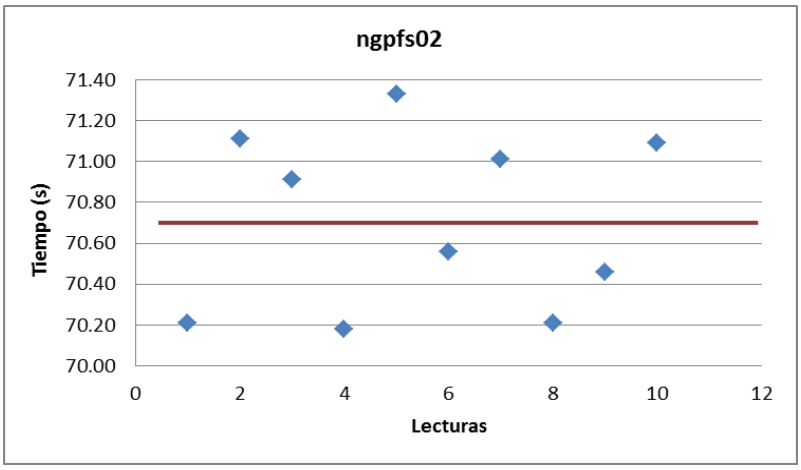


Figura 6.22 Tiempo de respuesta servidor pSeries 630 ngpfs02

La línea horizontal roja muestra el promedio de las 10 pruebas, que para los pSeries 630 era de 70.71 segundos, mientras que para el pSeries 615 el tiempo fue de 112.19 segundos. Se observaba claramente que éste último tenía una capacidad de rendimiento inferior a los dos anteriores, pues fue añadido para completar el quorum

requerido por GPFS, pero igualmente, estaba en capacidad de realizar tareas propias de base de datos pero con un balanceo de carga previo.

Estas pruebas de finales fueron ejecutadas en cada uno de los nodos, pero no en forma paralela repartiendo la carga entre los tres servidores.

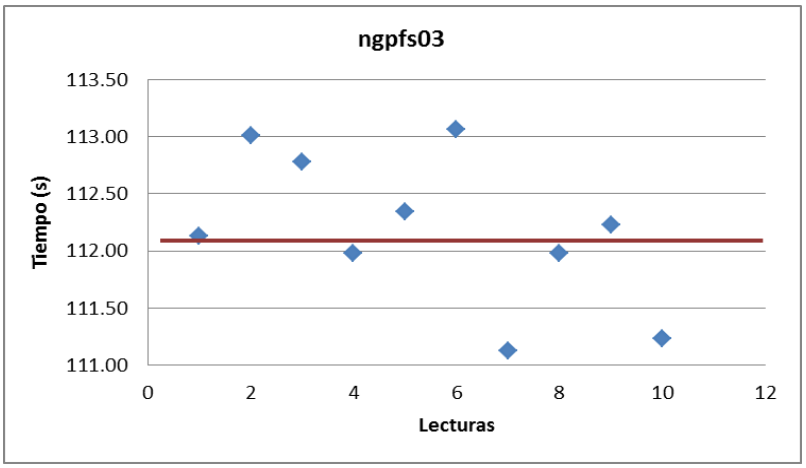


Figura 6.23 Tiempo de respuesta servidor pSeries 615 ngpfs03

Esas pruebas iban a estar a cargo de Oracle con el Banco Delbank una vez que se termina la fase de instalación e implementación de las aplicaciones Abanks. Banco DelBank realizó el correspondiente balanceo de la carga entre los tres servidores, con valores porcentuales de 40%, 40% y 20%.

CONCLUSIONES

1. De todos los proyectos en los cuales estuve involucrado mientras fui especialista de software AIX, el ejecutado en Banco DelBank fue en definitiva uno de los más importantes de todos, no solo por la complejidad del mismo, sino por el reto que el mismo implicaba pues en su momento no se había implementado una solución similar en América del Sur. Con esto, en Ecuador fuimos los pioneros en este tipo de instalaciones y fuimos los encargados de orientar a otros países en la preventa y postventa de soluciones similares.
2. Nuestro cliente comprendió la importancia de mantener infraestructuras tecnológicas de alta disponibilidad para proteger los sistemas de información y datos críticos.
3. Con la utilización de clusters altamente disponibles se logró estabilizar los servicios informáticos de una empresa en constante crecimiento. Una demostración de lo estable que ha quedado esta instalación, es que ha seguido funcionando ininterrumpidamente desde el año 2004. En el lapso de estos años, se han realizado cambios de discos, fuentes de poder y se han incrementado memoria a los servidores, pero nunca han tenido de paralizar la operación, demostrando en la práctica lo que en teoría debería pasar con este tipo de instalaciones, obteniendo valores mayores al 99,99% de disponibilidad del sistema.

4. La cantidad de carga se ha ido incrementando considerablemente en el transcurso de los años, pero el esquema de crecimiento de servidores en línea ha facilitado el ingreso de nuevos nodos y más subsistemas de discos al cluster inicial. El esquema inicial se lo sigue manteniendo y se lo ha replicado en otras sucursales y empresas.
5. A nivel competitivo contra compañías como SUN Microsystems o Hewlett Packard, IBM logró en ese entonces el 52% del mercado de servidores de rango medio en nuestro país. Con soluciones similares tuvimos que realizar "*benchmarks*" donde demostramos la fortaleza de GPFS contra soluciones similares como NFS (Network File Systems) ofrecidas por otros empresas.

RECOMENDACIONES

La tecnología de clusters con GPFS ha mejorado tecnológicamente en el transcurso de los años y sigue manteniéndose líder en el mercado mundial como responsable de la disponibilidad, seguridad y confiabilidad de la infraestructura en muchos clientes. Por lo anteriormente expuesto se recomendaría que:

1. En universidades y centros educativos superiores, podrían dictarse materias relacionadas con este tipo de soluciones, pues involucra un amplio grupo de proveedores tanto a nivel de sistemas operativos (AIX, Solaris, Linux, HPUX), base de datos (Oracle, DB2, Informix) y aplicaciones (Baan, JD Edwards, SAP, People Soft).
2. La instalación de laboratorios con clusters similares es posible, pues el costo del mismo se reduce considerablemente si se instala Linux en vez de AIX y finalmente empresas proveedoras de bases de datos están abiertas a ofrecer sus productos para pruebas en centros educativos.
3. El éxito de una instalación similar se logra manteniendo un solo proveedor en las diversas áreas de la infraestructura tecnológica. Muchos de los clientes en donde participamos con soluciones similares, mantenían centros de cómputo heterogéneos con decenas de servidores de diferentes marcas y sistemas operativos diferentes, haciendo casi imposible la administración de los mismos.

BIBLIOGRAFÍA

[1] Wan Hee Kim, Paulo Queiroz, Andrei Vlad. 2011. SG24-7844-00. Implementing the IBM General Parallel File System (GPFS) in a Cross-Platform Environment. First Edition (June 2011). This edition applies to IBM AIX 6.1 TL05, IBM Virtual IO Server 2.1.3.10-FP23, IBM General Parallel File System 3.4.0.1, RedHat Enterprise Linux 5.5. ibm.com/redbooks.

[2] Octavian Lascu, Vigil Carastanef, Lifang (Lillian) Li, Michel Passet, Norbert Pistoor, James Wang. 2003. SG24-6954-00 Deploying Oracle 9*i* RAC on IBM Eserver Cluster 1600 with GPFS. First Edition (October 2003); this edition applies to Version 5, Release 2, Modification 01 of AIX and Version 9.2.0.x of Oracle9i Real Application Clusters. ibm.com/redbooks.

[3] Pedro Diaz Robles. 2004. Oracle Real Application Cluster 10g. https://sites.google.com/site/mysitepeter/oracle-real-application-cluster. First Edition (March 2004).

[4] Octavian Lascu, Zbigniew Borgosz, Josh-Daniel S. Davis, Pablo Pereira, Andrei Socoliuc. 2003. SG24-6978-00. An Introduction to the New IBM Eserver pSeries High Performance Switch. First Edition (December 2003) applies to Version 5, Release 2, Modification 2 of AIX 5L (product number 5765-E62. ibm.com/redbooks.

[5] Jorge García Delgado. 2010. UT-06 Implantación de Soluciones de Alta Disponibilidad. Primera Edición (Agosto 2010).

[6] Rick Piasecki. 2002. Configuring the IBM General Parallel File System (GPFS) in the Oracle Real Application Cluster (RAC) Environment. First Edition (November 2002), Prepared by Rick Piasecki IBM eServer Technical Consultant.

[7] Abbas Farazdel, Robert Curran, Astrid Jaehde, Gordon McPheeters, Raymond Paden, Ralph Wescott. 2001. SG24-6035-00. GPFS on AIX Clusters: High Performance File System Administration Simplified. First Edition (August 2001)
This edition applies to Version 1 Release 4 of IBM General Parallel File System for AIX (GPFS 1.4, product number 5765-B95) or later for use with AIX 4.3.3.28 or later. ibm.com/redbooks.

[8] Judy Campos. 2002. SC23-4277-04. High Availability Cluster Multi-Processing for AIX Planning Guide Version 4.5. Fifth Edition (June 2002), This edition applies to HACMP for AIX, version 4.5 and to all subsequent releases of this product until otherwise indicated in new editions.

[9] Copyright International Business Machines Corporation. 2003. GC26-7574-00. IBM TotalStorage FAStT Storage Manager Version 8.4 Installation and Support Guide for AIX, HP-UX, and Solaris. First Edition (September 2003)