UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES

ARAGÓN

IMPLEMENTACIÓN DE UN SISTEMA DE ALTA DISPONIBILIDAD PARA APLICACIONES DE MISIÓN CRÍTICA EN EL DEPARTAMENTO DE ADMINISTRACIÓN DE SERVIDORES DE LA DGSCA

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A N:

AGUSTÍN REYES VILLEGAS
SERGIO CANTE MARTÍNEZ

ASESOR:
ING. LILIANA HERNÁNDEZ CERVANTES

MÉXICO **2005**

Agradezco profundamente a cada uno de los miembros de mi familia por su apoyo incondicional a lo largo de mi carrera y de la realización de este proyecto.

Gracias a la Universidad Nacional Autónoma de México por haberme dado la oportunidad de forjarme profesionalmente.

Agustín

A mis padres que con su humildad me enseñaron que todo se logra a base de trabajo. A mis hermanos por todo su apoyo en momentos difíciles.

Buddy y Anali por encontrar en ustedes a mis mejores amigos. Gracias a ti por motivarme a concluir este proyecto.

Sergio

Prefacio

Este trabajo simboliza un esfuerzo por presentar un panorama general sobre los diferentes niveles de disponibilidad de los sistemas de cómputo y la aplicación de algunos de ellos, específicamente sobre el servidor WEB de la Universidad Nacional Autónoma de México. Este texto pretende proporcionar información básica suficiente para cualquier Administrador del sistema operativo UNIX - Solaris que desee enfocar su sistema a un esquema de alta disponibilidad.

Esperamos que el presente trabajo pueda ser de utilidad, ayudando a cubrir la ausencia de material introductorio relacionado con la disponibilidad de los sistemas de cómputo, y que sirva como referencia para futuros trabajos, incluso sobre el propio servidor WEB de la universidad más importante de México.

Los Autores

Contenido

Introducción

En la industria de hoy, un número significativo de clientes está adoptando el modelo de cómputo cliente/servidor, en el cual las computadoras personales y estaciones de trabajo de escritorio tienen acceso, y con frecuencia reposan sobre servicios provistos por sistemas servidores especializados. De la misma manera, muchos clientes corporativos están haciendo reestructuraciones de su tecnología de información, para explotar la flexibilidad de cliente/servidor y acceder a productos de hardware y software de bajo costo.

A medida que estos clientes ponen a funcionar soluciones cliente/servidor, las empresas empiezan a preocuparse tanto por la confiabilidad como por la efectividad por costos de la curva de crecimiento de estos sistemas en el futuro; estos atributos son críticos para soportar a la base de usuarios y para el éxito de negocios. La tecnología de Alta Disponibilidad para plataformas UNIX esta diseñada precisamente para enfrentar estos problemas mejorando la disponibilidad, la escalabilidad y la administración de datos y de servicios claves dentro de un ambiente cliente/servidor LAN.

En la actualidad el internet está transformando nuestra vida en muchos aspectos entre ellos el de los negocios, gobierno y educación por citar algunos; cada vez se desarrollan procesadores mas rápidos, manejadores de bases de datos mas estables, discos duros cada vez mas inteligentes, hoy por hoy quien quiera competir en el mercado y no cuente con redundancia en sus equipos para incrementar la disponibilidad de sus datos y servicios estará prácticamente sin posibilidades de éxito.

Las redes de información se han convertido en parte central de toda empresa sea cual sea el ámbito en el que se desarrolla, el número de aplicaciones que se ejecutan son cada vez mas grandes y a medida que pasa el tiempo se convierten en críticas para la empresa, al mismo tiempo éstas aplicaciones hacen que la empresa se expanda y con ello toma mayor relevancia el manejo de la información, por consiguiente se hace necesaria la implementación de la Alta Disponibilidad en los sistemas de cómputo, haciendo que la implementación de ésta sea una prioridad para la empresa.

Las pérdidas mas obvias cuando no está disponible el sistema se ven en la pérdida de la productividad, más sin embargo, las pérdidas monetarias varían dependiendo del servicio que se esté proporcionando, por ejemplo: si los usuarios son desarrolladores y el giro de nuestra

empresa es el desarrollo, estaremos experimentando pérdidas significativas, supongamos que el sueldo de un desarrollador varia entre 100 y 500 dólares al día y que nuestro equipo de desarrolladores es de 50 elementos, al estar abajo nuestro sistema por una semana que equivale a contar con un 98% de disponibilidad, al año estaríamos perdiendo 175,000 dólares esto sin contar que debemos de invertir una suma bastante considerable para que los desarrolladores recuperen el tiempo perdido, en muchas ocasiones estas cifras pueden llegar a triplicarse.

Imagine ahora, una empresa un poco mas robusta la cual se dedica a las ventas vía telefónica y por internet, suponga ahora usted que alguien desea comprar el reloj de moda que cuesta 68 dólares, y que su sistema está abajo por unos cuantos minutos, en esos minutos usted estará perdiendo las ventas por el reloj y peor aún estará perdiendo a un cliente que la próxima vez que busque un producto seguramente lo comprará con la competencia que sí tiene su sistema en línea, si multiplicamos esto por todas las llamadas que perdimos y a esto le sumamos los clientes que teníamos en el WEB, las pérdidas se incrementan aún más, realmente es difícil cuantificar cuanto perdimos en unos cuantos minutos. Si nuestro sistema no estuvo totalmente abajo pero experimenta lentitud, nuestro cliente dirá a todas sus amistades que brindamos un mal servicio y de este modo estaremos perdiendo potenciales compradores, es por esto, que se debe definir que es downtime, esta palabra cobrará mayor o menor relevancia dependiendo de la criticidad de la aplicación que se ejecute en el servidor.

La Universidad Nacional Autónoma de México no está a expensas de estos cambios, nuestro servidor de WEB, debe estar disponible los 365 días año, las 24 horas del día (a medida que nos vayamos adentrando en la tesis nos daremos cuenta que en la vida real no existe un disponibilidad real del 100%), esto es debido a la relevancia académica que tiene la UNAM en nuestro país, dentro de este portal se publican comunicados del rector, de investigadores, de académicos, etc., que son de mucha importancia para la comunidad universitaria.

Estructura de este trabajo

Este trabajo ha sido estructurado de tal forma que antes de realizar las actividades prácticas, se tengan las bases teóricas para poder desarrollar dichas actividades.

Capítulo 1. En este capítulo se mencionan algunos conceptos básicos referentes a disponibilidad y las fallas de los sistemas de cómputo, también se da una explicación general de la arquitectura de la plataforma Sun Ultra-Enterprise-10000 ya que los dos servidores que

utilizamos a lo largo de este trabajo se encuentran en dos plataformas de este tipo, por tal motivo también mencionamos la configuración de esas dos plataformas.

Capítulo 2. Se dan las bases teóricas sobre todo lo que engloba la elaboración de respaldos, se tratan conceptos y recomendaciones, posteriormente se analiza el esquema de respaldos actual del servidor WEB y se diseña y pone en práctica un nuevo esquema de respaldos de información.

Capítulo 3. El capítulo tres está enfocado a lo que es el manejo de los datos, se habla de las diferentes tecnologías de discos disponibles en la actualidad y de tres paquetes de software para manejar discos a través de los cuales se puede lograr redundancia en ellos. Posteriormente se analiza la configuración actual de los discos del servidor web y se plantea e implementa un nivel de disponibilidad de datos mas elevado que el actual mediante la redundancia en los discos.

Capítulo 4. En el desarrollo de este capítulo, se llega a un esquema de replicación de datos del servidor web, esto se logra utilizando el mismo servidor web y un servidor adicional que se llama newton que se menciona en el capitulo 1 y al cual se mandan los datos. Por supuesto antes de esta implementación se dan las bases teóricas sobre la replicación de datos.

Capítulo 5. Finalmente, en el capítulo 5 hablamos de las conclusiones de este trabajo, además dejamos las bases establecidas para la implementación de un cluster, por si en algún momento dado es posible ponerlo en práctica. Es importante mencionar, que la implementación de dicho cluster no se pudo llevar a cabo porque para ello necesitábamos hardware y software adicional, lo cual implicaba una buena cantidad de dinero, desafortunadamente la universidad en estos momentos no cuenta con el presupuesto suficiente para realizar esta actividad, por lo que solo realizamos el planteamiento de forma teórica.

Objetivo

Presentar la alta disponibilidad como un requerimiento ineludible para la disposición de la información de cualquier empresa, aplicándola directamente al servidor web de la Universidad Nacional Autónoma de México. Analizar tanto el esquema de respaldos actual, así como la configuración de discos, y proponer e implementar nuevas soluciones implicando mejoras. Proporcionar un trabajo que pueda ser utilizado como base para la implementación de un sistema de Alta Disponibilidad.

Capítulo 1. Fundamentos

En este capítulo se introduce al lector al tema de disponibilidad y algunos conceptos relacionados con la misma. Se pretende que la disponibilidad y los conceptos que aquí mencionamos sean comprendidos de una manera general y que el entenderlos facilite la comprensión de capítulos posteriores. De la misma manera se da una explicación de cómo está compuesta la plataforma Sun Ultra-Enterprise-10000 ya que el servidor WEB de la UNAM y un servidor adicional que será mencionado en el capítulo 4 para replicación de datos están implementados en una máquina de este tipo, así mismo se muestra a grandes rasgos la configuración de estos dos últimos servidores.

Definición y medición de la disponibilidad

En su modo más simple la disponibilidad es una medida del tiempo en que un sistema de cómputo está funcionando normalmente. La disponibilidad se puede calcular con la siguiente fórmula:

$$A = \frac{MTBF}{MTBF + MTTR}$$

Donde A es el grado de disponibilidad expresado en un porcentaje, MTBF (Mean Time Between Failures) es el Tiempo Medio Entre Fallas, y MTTR (Maximun Time To Repair) el Máximo tiempo Para Reparar o resolver un problema en particular.

Si el MTTR se aproxima a cero, A se aproximará a 100 %; si el MTBF tiende a ser muy largo, MTTR tiene un menor impacto sobre A y por lo tanto la disponibilidad A se incrementa.

Por ejemplo, supongamos que un sistema cualquiera tiene un MTBF de 100 000 horas, y un MTTR de 1 hora, sustituyendo estos valores en la ecuación:

$$A = \frac{100\,000}{100\,000 + 1} = \frac{100\,000}{100\,001} = 0.99999 = \mathbf{99.999\ \%}$$

Entonces podemos decir que el sistema de éste ejemplo tiene una disponibilidad del **99.999 %.**

Si lográramos disminuir el MTTR a 6 minutos, la disponibilidad se incrementaría un nueve más:

$$A = \frac{100\,000}{100\,000 + 0.1} \qquad \frac{100\,000}{100\,000.1} = 0.999999 = \textbf{99.9999 \%}$$

Para lograr este nivel de disponibilidad de **99.9999 %,** solo deberíamos de permitir 6 minutos de tiempos muertos en 11.4 años, lo cual en la práctica suena muy difícil de conseguir y es prácticamente algo imposible. Tiempos sin funcionar de 10 minutos por año (alrededor de 99.998 de disponibilidad) es algo más real de alcanzar, aunque es muy difícil mejorar esta cifra.

Definición y causas de tiempos muertos

Para mucha gente el definir un concepto como este es algo difícil ya que los proveedores de equipo y soluciones de Alta Disponibilidad aun no se han puesto de acuerdo en una definición exacta. Haciendo caso a las definiciones sencillas podemos decir que un "**tiempo muerto**" es el lapso de tiempo en que un sistema de cómputo deja de funcionar por causa de alguna interrupción.

A su vez una **interrupción** se da cuando un sistema deja de funcionar por causa de factores internos o externos conocidos como fallas. Ejemplos de factores internos pueden incluir errores de especificación y de diseño, errores de configuración de software; defectos de manufactura, defectos de componentes de hardware, y componentes obsoletos. Ejemplos de factores externos pueden incluir radiación, interferencia electromagnética, errores de operación, y desastres naturales.

Todas las interrupciones de sistema caen dentro de dos principales categorías:

Interrupciones de sistemas no planeadas e interrupciones de sistema planeadas.

Interrupciones de sistema no planeadas (fallas)

Las interrupciones no planeadas son el resultado de fallas de sistema aleatorias e incontrolables asociadas con defectos que ocurren en el hardware, componentes de software y su configuración o acontecimientos naturales del propio ambiente. Interrupciones no planeadas son las más costosas, con los más altos beneficios logrados cuando se toman medidas para evitarlas. Las interrupciones no planeadas pueden ser minimizadas a través de redundancia física de componentes de hardware.

Fallas de Hardware

El componente más complejo de cualquier sistema de cómputo es el servidor. El servidor consiste de decenas de componentes: CPU's, discos, memoria, fuentes de poder, ventiladores, tarjetearía, huecos de expansión, etc.; todos ellos a su vez constan de muchos subcomponentes.

Algunos de estos componentes pueden fallar, contribuyendo así a que el servidor falle, es entonces cuando estaremos hablando de una falla de hardware.

Aunque este tipo de falla origine caídas de los sistemas solo en un diez por ciento, cuando una caída ocurre en lo primero que pensamos es en una falla de hardware. Los componentes que más comúnmente ocasionan fallas de este tipo son aquellos que son móviles o intercambiables de posición, especialmente aquellos asociados con alta velocidad y baja tolerancia a su uso. En este contexto los discos son los primeros candidatos a sufrir fallas. Los discos también tienen tarjetas controladoras o cables que pueden romperse o fallar.

Los manejadores de cintas y librerías, especialmente las librerías de cintas de tipo DLT (Digital Lineal Tape), tienen muchas partes móviles, motores que paran e inicializan y que tienen una tolerancia extremadamente baja. También tienen tarjetas controladoras y muchos de los componentes internos que usan son los mismos que utilizan los discos, incluyendo cierto tipo de memoria.

Los ventiladores son los otros componentes con partes móviles. La falla de un ventilador no ocasionará que el sistema falle inmediatamente, pero cuando el sistema completo de enfriamiento falle los efectos pueden ser impredecibles, cuando los CPU´s y los circuitos integrados de memoria se sobrecalienten el sistema tendrá irremediablemente un mal funcionamiento.

De todos los componentes de un servidor los ventiladores y las fuentes de poder son los que tienen el peor MTBF. Las fuentes de poder pueden fallar drástica y rápidamente, resultando simplemente en un tiempo muerto, pero también pueden fallar gradualmente. La falla gradual de una fuente de poder puede ser un problema muy serio, y puede ser una sutil causa de una falla en los CPU´s, memoria o tarjetas principales. Las fallas de las fuentes de poder son causadas por muchos factores, incluyendo variaciones de voltaje y el estrés de ser encendidas y apagadas constantemente.

Para cubrir estos factores, los sistemas modernos tienen ventiladores y fuentes de poder extras, y diagnósticos de hardware superiores que proveen la detección e identificación de un problema tan rápido como sea posible. Muchos sistemas pueden inclusive estar configurados para "llamarnos a casa". Cuando un componente falla el sistema automáticamente llama al centro de servicio para hacer un requerimiento de mantenimiento.

Por supuesto que las fallas también pueden ocurrir en la memoria y en los CPUs. Otra vez, algunos sistemas modernos son capaces de poner fuera de la configuración del sistema un componente que ha fallado sin necesidad de reiniciar el equipo.

Existen otros componentes de hardware que también están propensos a fallas, aunque es menos frecuente que suceda. Estos incluyen las tarjetas principales, tarjetas de sistema y los gabinetes de discos.

Fallas de software y de configuración

En cuanto a las fallas en el software existe una gran diversidad, puede tratarse desde el simple daño de un archivo ya sea de sistema operativo o de alguna aplicación, o inclusive de toda una base de datos. Puede tratarse de algunos parámetros de un archivo que tienen valores diferentes a los óptimos y que por lo mismo están mal configurados, o de la falta de algún "parche". Un parche es un archivo o conjunto de archivos o programas que tiene como propósito corregir ciertos errores del software existente.

Fallas de Operación

Como humanos que somos no estamos exentos de cometer errores durante la operación o administración de los sistemas, errores como el borrado de archivos de configuración, errores "de dedo", de programación, o de una mala planeación de actividades,

son fallas que comúnmente hacen más robusta la gama de interrupciones de los sistemas de cómputo y por lo mismo son causa de tiempos muertos.

Fallas en la Red

Las redes son naturalmente susceptibles a fallas porque contienen muchos componentes y son afectadas por la configuración de cualquiera de ellos. ¿Dónde exactamente está la red?, ¿En los cables?, ¿En las tarjetas del servidor?, ¿En los concentradores de cables?. Algunos de estos componentes físicos puede romperse, teniendo como resultado tiempos muertos o en el mejor de los casos fallas de red intermitentes.

Las redes son también afectadas por problemas de configuración. Incorrecta información de las direcciones lógicas de los servidores, nombres de servidores duplicados, o máquinas que interpretan mal las direcciones. También influyen las conexiones de red redundantes así como las conexiones de red en múltiples puntos. Cuando esa redundancia se rompe o su configuración es mal interpretada, la red estará susceptible a sufrir fallas.

Fallas físicas y fallas originadas por el ambiente

Existen muchos componentes en el ambiente que pueden originar tiempos muertos en los sistemas de cómputo, aunque estos son raramente considerados como puntos de falla potenciales. El problema más obvio es una falla de potencia. Las fallas de potencia pueden provenir de los suministros de electricidad (UPS´s), o puede ser un problema mucho más local. El sistema de enfriamiento del centro de cómputo puede fallar causando sobrecalentamiento masivo en todos los sistemas que allí se encuentren y potencialmente la falla de los mismos.

En algunos centros de cómputo, aunque suene raro hay nidos de ratas entre los cables, abajo del piso falso y detrás de los gabinetes de los servidores o de los discos. Los cables pueden ser mordidos, rotos y obviamente volverse inservibles y ocasionar alguna falla.

En otros centros de cómputo existen sistemas de protección contra el fuego. Existen problemas que se presentan cuando el fuego es real. El agua u otros agentes extintores pueden originar residuos que afectan a los servidores al grado de dejarlos fuera de operación.

Otro problema potencial del ambiente son las fallas estructurales de los componentes de soporte, como pueden ser los de los gabinetes de los servidores, los cuales pueden colapsarse o venirse abajo cuando las estructuras que los van a soportar no están construidas apropiadamente.

Finalmente, existen verdaderos desastres ambientales que pueden afectar nuestros centros de cómputo y por consiguiente nuestros sistemas de cómputo: los desastres naturales, como son terremotos, tornados, inundaciones, bombas y otros actos de guerra y terrorismo.

Interrupciones de sistema planeadas (mantenimientos)

Una interrupción planeada debe ser programada para tener un mínimo impacto de disponibilidad en el sistema. Las interrupciones planeadas son el resultado de eventos de mantenimiento para reparar o reemplazar hardware, respaldar información, o actualizar software y operaciones. Las reparaciones tienen el propósito de remover y sustituir componentes dañados y llevar al sistema a un estado funcional. Los respaldos se hacen con la finalidad de preservar datos críticos en medios de almacenamiento magnéticos (discos o cintas) para evitar pérdida de información cuando los sistemas experimentan una falla de almacenamiento. Las actualizaciones son realizadas para reemplazar el hardware o software actual con nuevas versiones.

Un buen mantenimiento implica tener un plan de trabajo previamente distribuido en el cual se incluyan tiempos, actividades, responsabilidades y personal involucrado, en muchas ocasiones si es posible es de gran ayuda contar con un plan de contingencia el cual nos ayudará a regresar a un punto en que el sistema funcionaba correctamente en caso de que por alguna razón el mantenimiento no tuviera éxito.

Pérdidas monetarias por interrupciones del sistema

Las afectaciones mas obvias cuando no esta disponible el sistema se ven en la pérdida del dinero, mas sin embargo las perdidas monetarias varían dependiendo del servicio que se esté proporcionando, por ejemplo: si los usuarios son desarrolladores y el giro de nuestra empresa es el desarrollo, estaremos experimentando perdidas significativas, supongamos que el sueldo de un desarrollador varia entre 100 a 500 pesos al día y que nuestro equipo de desarrolladores es de 50 elementos, el hecho de que nuestro sistema deje de funcionar por una semana que equivale a contar con un 98% de disponibilidad al año traería como consecuencia la pérdida de 175,000 pesos en un año, esto sin contar que debemos de invertir una suma bastante considerable para que los desarrolladores recuperen el tiempo perdido.

Imaginemos ahora, una empresa un poco más robusta la cual se dedica a la venta de artículos vía telefónica y por Internet, suponga ahora usted que alguien desea comprar el reloj de moda que cuesta 1800 pesos, y que su sistema deja de funcionar por unos cuantos minutos, en esos minutos usted estará perdiendo las ventas por el reloj y peor aun estará perdiendo a un cliente que la próxima vez que busque un producto seguramente lo comprará con la competencia que sí tiene su sistema en línea o funcionando, si multiplicamos esto por todas las llamadas que se perdieron y a esto le sumamos los clientes que teníamos en el WEB, las perdidas se incrementan aun más, realmente es difícil cuantificar cuanto perdiste en unos cuantos minutos. Si nuestro sistema no estuvo totalmente sin funcionar pero experimenta lentitud, nuestro cliente dirá a todas sus amistades que brindamos un mal servicio y de este modo estaremos perdiendo compradores potenciales.

Estos ejemplos, son solo eso, ejemplos, pues en la realidad existen empresas que el hecho de tener solo algunos minutos de tiempo muerto de su sistema, ocasiona la pérdida de cantidades de dinero estratosféricas.

Niveles de Disponibilidad

En general la disponibilidad de los sistemas de cómputo se divide en cuatro niveles, cada uno de ellos tiene sus características propias y abarca varios puntos que se deben considerar.

Nivel 1. Disponibilidad Normal

La protección más básica es esencialmente aquella que no te da ninguna protección, valga la redundancia, no existen dispositivos que nos brinden protección a los sistemas o discos. El primer nivel puede incluir respaldos de información y nada más. Además en este nivel no existen planes que pretendan disminuir los tiempos en que nuestras aplicaciones no prestan el servicio. Normalmente este nivel es suficiente para que trabajen muchas aplicaciones, pero puede dejarnos en algunas ocasiones fuera de línea por varios días, inclusive se puede experimentar pérdida en la integridad de los datos y en el peor de los casos la pérdida total. Definitivamente este nivel no es recomendable para quien desea mantener sus aplicaciones el mayor tiempo posible en línea.

Nivel 2. Disponibilidad incrementada

El nivel 2 es en esencia similar al nivel 1, con la diferencia de que se implementa cierta protección a los datos, esto quiere decir, que se implementa tecnología de manejo de discos RAID (Redundant Array of Independent Disk) o Arreglos Redundantes de discos Independientes, está tecnología será tratada mas a fondo en el capítulo tres; muy frecuentemente se utiliza el espejeo de discos, que no es otra cosa que una copia de datos adicional, pero también existen configuraciones que emplean RAID-5, el cual utiliza una parte de disco denominada paridad que permite en caso de ser necesario reconstruir los datos. Normalmente en este nivel no existen perdidas de datos por fallas en los discos ya que estos utilizan la tecnología RAID como protección; si a esto le añadimos que los respaldos se realicen correctamente se reduce aun más la probabilidad de que la información se pierda por daño en los discos, aunque normalmente los respaldos en este nivel se ocupan para protegernos de los propios usuarios que accidentalmente borran los datos. Hasta este nivel nos hemos protegido de los componentes en nuestro sistema que tienden a fallar mas: los discos, pero aun no estamos protegidos de las fallas en otros componentes de nuestro sistema, los cuales nos pueden ocasionar tiempos muertos de hasta varios días.

Nivel 3. Alta disponibilidad

El nivel 3 es comúnmente conocido como Alta Disponibilidad o HA por sus siglas en ingles (High Availability), en este tipo de configuración contamos con dos servidores, los cuales trabajan como si fueran un solo sistema, cada máquina tiene su propio sistema operativo y ambos trabajan con las mismas aplicaciones y además tienen acceso a los mismos discos duros, a esta configuración se le conoce como cluster. Uno de los dos servidores es llamado servidor primario y es el que tiene el control, y el otro es el servidor secundario el cual está en espera de que una falla en el servidor primario ocurra para tomar el control. El cambio de control puede ser manual o automático dependiendo de cómo se configure, y del software de manejo del cluster utilizado.

Algunos tiempos muertos pueden ocurrir en las configuraciones de alta disponibilidad, pero en la mayoría de los casos los tiempos que estamos fuera de línea son limitados. Los niveles de disponibilidad exceden el 99.98%, incluso el nivel 3 pueden tener mayores niveles de disponibilidad si implementamos protecciones extras a nuestros sistemas, una buena configuración de HA debe incluir una red redundante, auditoría a los sistemas, esto incluye seguridad y configuración de los equipos. El añadir una protección de este tipo a nuestros sistemas incrementa la disponibilidad de estos, pero también incrementa la complejidad de nuestro sistema y por ende la dificultad de administrarlo y los costos de implementación y de mantenimiento.

Estrechamente ligada al cluster está la replicación de datos, la cual también hace uso de dos servidores solo que en lugar de que ellos accedan a los mismos discos como en el cluster, en la replicación cada servidor debe tener su propio conjunto de discos de datos, y cada conjunto de discos la misma información, con lo cual tenemos dos copias de la misma información y por lo tanto tenemos mas protección en caso de corrupción de datos, aunque esta implementación también eleva los costos de manera considerable.

Nivel 4. Recuperación ante desastres

Este nivel es el más alto nivel de protección y en consecuencia es el más caro. Cuando se implementa "Recuperación Ante Desastres" (mejor conocido por DR por sus siglas en ingles), se está protegiendo a la empresa de una pérdida total, ya sea del centro de cómputo o del edificio; este tipo de solución implica el tener otro centro de cómputo alterno, a una distancia considerable del centro de cómputo principal, lo cual quiere decir, que debemos contar con la infraestructura de hardware y software necesarios para que nuestras operaciones puedan trabajar sin problema alguno en el centro de cómputo alterno, es por esto que, normalmente se debe tener exactamente el mismo equipo en ambos centros de cómputo.

Sistemas Tolerantes a Fallas.

Los sistemas tolerantes a fallas son sistemas muy caros que rara vez se implementan, un sistema tolerante a fallas es un sistema que cuenta con doble, triple o hasta cuádruple redundancia de sus componentes de hardware. Si un componente falla otro tomará su lugar sin interrumpir su funcionamiento, y el componente que falló puede ser reparado o reemplazado en línea.

Si un sistema tolerante a fallas esta bien configurado, ningún error de hardware interrumpirá el servicio. El único componente de nuestro sistema que puede ser capaz de interrumpir el servicio es el software. Alguna falla en una aplicación crítica o en el sistema operativo puede causar que nuestro sistema deje de funcionar y por ende interrumpir el servicio. Actualmente un sistema de este tipo esta valuado en varios millones de dólares, es por esto que cuando se requiera implementar HA para nuestros sistemas, antes debemos evaluar cuáles son nuestras verdaderas necesidades y de esta forma utilizar el nivel de disponibilidad adecuado.

Aunque el nivel tres nos dice que la alta disponibilidad es la implementación de un cluster, en realidad la alta disponibilidad puede ir desde un buen esquema de respaldos, hasta la

configuración de un sistema tolerante a fallas, pasando por la redundancia de los discos, la implementación de un cluster, la replicación de datos y la recuperación ante desastres.

Características Generales de la plataforma Ultra-Enterprise-10000

El servidor WEB de la UNAM y el servidor adicional que utilizaremos en este trabajo específicamente en el capítulo 4, están configurados en una plataforma SUN Ultra-Enterprise-10000, por este motivo mencionaremos de manera general las características principales de esta plataforma así como de su sistema de control (SSP).

Arquitectura

La plataforma SUN Ultra-Enterprise-10000 es un sistema de hardware escalable y multiprocesamiento que trabaja con Sistema Operativo UNIX en su variante de Solaris. Este sistema es ideal para aplicaciones de propósito general y servidores de datos basados en el modelo cliente/servidor tales como procesos de transacciones en línea, de almacenamiento de datos, servicios de comunicaciones o servicios multimedia. Para darnos una idea de las proporciones físicas debemos mencionar que el gabinete, de la Ultra-Enterprise-10000 es una caja que tiene 1.8 m de altura, 1.3 m de largo y 1 m de ancho, y que completamente llena de hardware tiene un peso aproximado de 700 kg. Esta máquina tiene una parte frontal y una parte trasera cada una de las cuales se conoce como "cara", cada cara tiene capacidad para la misma cantidad de hardware. Dentro de una plataforma de este tipo podemos tener configurados hasta 16 servidores, cada uno con sus propios recursos en cuanto a CPUs, memoria y tarjetas de IO se refiere y ejecutando su propia copia de sistema operativo, a cada uno de estos servidores se le conoce como dominio.

Toda la plataforma incluyendo todos los dominios que tenga configurados es controlada, administrada y monitoreada por una máquina de menor proporción llamada SSP (System Service Procesor) la cual está conectada físicamente a la plataforma. Una plataforma puede tener hasta 2 SSP pero solo una podrá tener el control. Adicionalmente conectados a la plataforma podemos tener discos o arreglos de discos para almacenar la información.

A continuación presentamos una imagen de la Ultra-Enterprise-10000 y de su SSP.

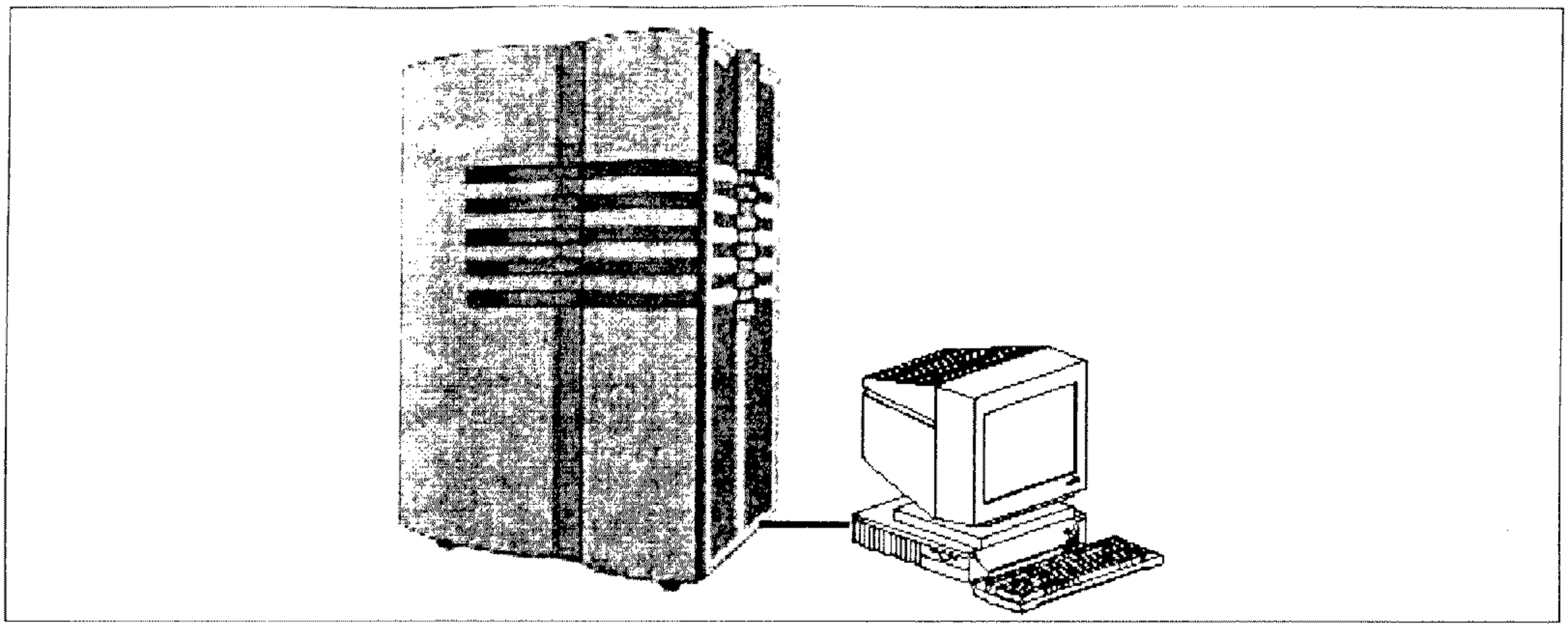


Figura 1.1. Vista exterior de una plataforma Sun Ultra-Enterprise-10000 y su SSP

Ya vimos que la Ultra-Enterprise-10000 está compuesta de un gabinete, una SSP y opcionalmente de arreglos de discos. A continuación veremos como está constituido el interior de la plataforma.

De manera general el gabinete contiene los siguientes componentes principales:

Tarjeta de sistema. Contiene en su interior procesadores, memoria, subsistemas de entrada y salida de datos, tarjetas de red, tarjetas para conexión de discos y convertidores de potencia. La Ultra-Enterprise-10000 puede tener hasta 16 tarjetas de sistema, 8 en cada cara. Cada tarjeta de sistema puede tener hasta 4 procesadores, 4 GB de memoria RAM y hasta 4 tarjetas de red o de conexión a discos.

Tarjeta central. Es una tarjeta que proporciona interconexión de datos y direcciones a todas las tarjetas del sistema. Esta tarjeta se encuentra ubicada en la parte central del gabinete y contiene huecos que permiten la conexión de las tarjetas de sistema.

Tarjeta de control. Se encarga del manejo de la comunicación entre la tarjeta central y los demás componentes de las tarjetas de sistema a través de un protocolo llamado JTAG, también tiene como función controlar los ventiladores, la potencia, la frecuencia de reloj y funciones de red. Esta máquina puede tener hasta 2 tarjetas de control.

Tarjeta de soporte de la tarjeta central. Provee el protocolo JTAG, también provee funciones de control de componentes.

Sistema de potencia de 48 volts. Este sistema se compone de los siguientes subsistemas:

Módulos de entrada de corriente alterna. Recibe 220 volts de corriente alterna, los monitorea y después los pasa a las fuentes de potencia.
Fuentes de potencia. Reciben la corriente de los módulos de entrada de corriente alterna.
Panel de Interrupción de Potencia. Interrumpe la potencia a varios componentes.

Secuenciador de potencia de corriente alterna. Recibe y monitorea 220-volts AC y los transmite a los dispositivos. Convierte la corriente alterna a corriente directa para los dispositivos. Puede tener 1 o más.

Subsistema de enfriamiento. El subsistema de enfriamiento contiene dos tarjetas de ventiladores, cada tarjeta de ventiladores puede tener 8 bandejas de ventiladores, y a su vez cada bandeja puede contener 2 ventiladores, por lo tanto esta máquina puede tener en total hasta 32 ventiladores.

A continuación se muestran una vista interna de la parte frontal de la plataforma SUN Ultra-Enterprise-10000 y una vista de su parte trasera.

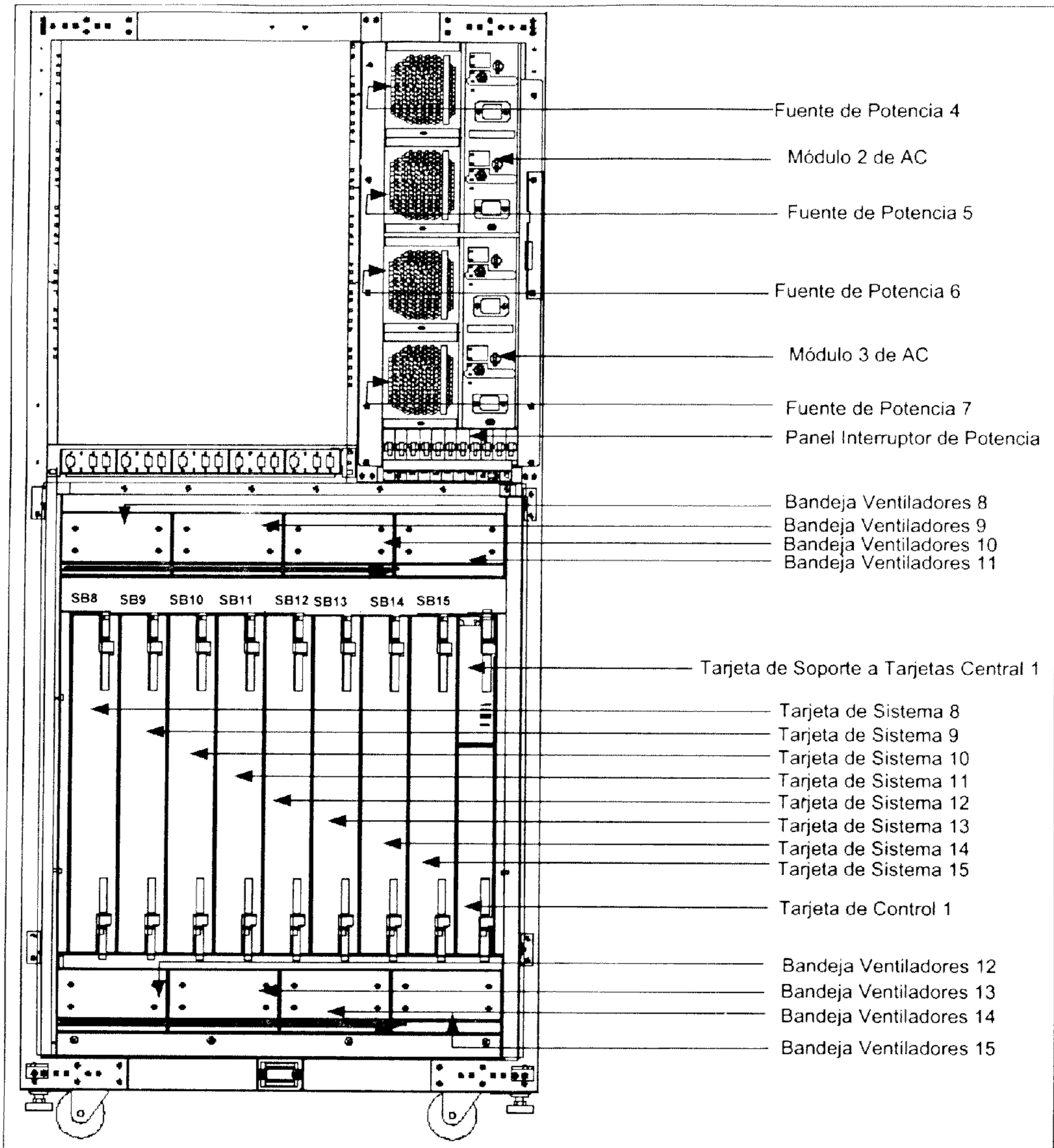


Figura 1.2. Vista frontal de la plataforma SUN Ultra-Enterprise-10000

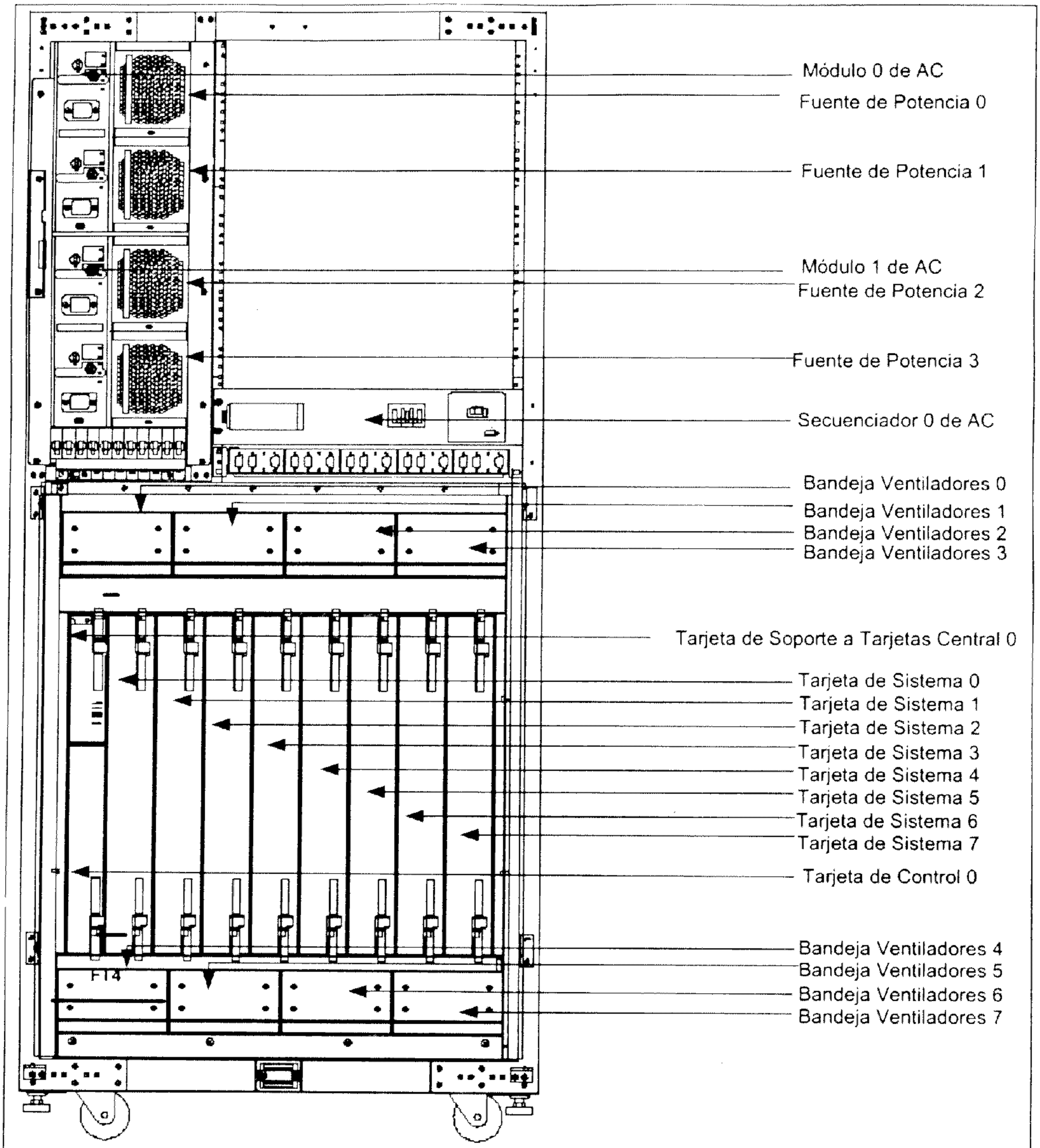


Figura 1.3. Vista trasera de la plataforma SUN Ultra-Enterprise-10000

Existe una parte a la que hicimos referencia en el inicio de este apartado y que no hemos visto en las graficas anteriores, esta parte es la tarjeta central y la razón por la que no la hemos podido apreciar es porque se encuentra exactamente a la mitad de la máquina y en ese lugar donde van insertadas las tarjetas de sistema, por lo tanto una vista frontal o trasera no nos permite verlo, pero si una vista superior del gabinete. A continuación se muestra una vista superior de la Ultra-Enterprise-10000, hay que notar que solo tiene 4 tarjetas de sistema y los demás huecos están vacíos. Las tarjetas de sistema que tiene a su vez nos muestran los módulos de memoria, procesadores y tarjetas de entrada/salida de datos.

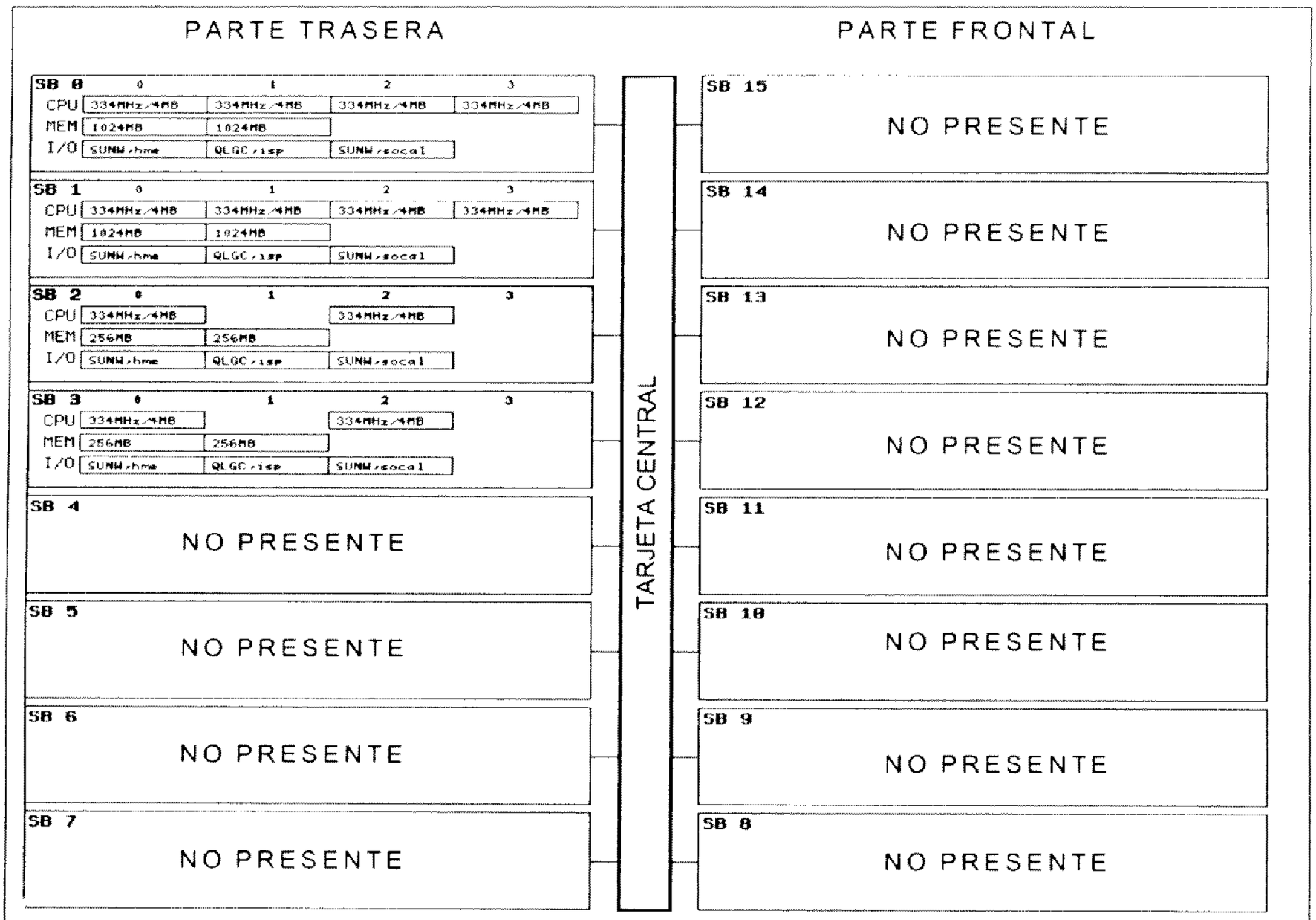


Figura 1.4. Vista superior de la plataforma SUN Ultra-Enterprise-10000

Tarjetas de sistema

Como ya vimos las tarjetas de sistema se componen de subcomponentes como procesadores, memoria RAM y tarjetas de entrada/salida de datos. Cada tarjeta de entrada salida que se inserta en una tarjeta de sistema se introduce en una ranura específica, cada tarjeta de sistema tiene cuatro ranuras, las dos primeras ranuras corresponden a un canal de datos y las dos siguientes a otro canal, para facilitar el entendimiento de esto numeremos a cada ranura desde el número 0 al 3, así la primer ranura empezando de arriba para abajo le asignaremos el número 0, a la segunda el 1, a la tercera el 2 y a la cuarta ranura el número 3. Esta nomeclatura aplica para cualquier tarjeta de sistema. Veamos la siguiente gráfica.

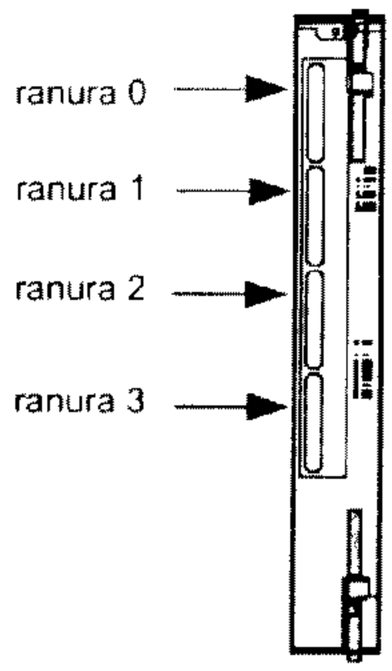


Figura 1.5. Ranuras de entrada/salida de datos de una tarjeta de sistema

Segmentación en dominios

La plataforma SUN Enterprise-10000 debe de ser identificada con un nombre para efectos de administración. Esta máquina tiene la capacidad de dividirse en varios sistemas separados o lo que es lo mismo en varios servidores, cada uno de los cuales tiene sus propios recursos de software y de hardware, cada servidor separado es llamado dominio. Cada dominio es administrado y manejado de manera independiente. Los dominios son implementados por características especiales que soportan tanto el hardware de la Enterprise-10000 como el software de la SSP. Cuando se crea un dominio se le asigna sus propias tarjetas de sistema y por ende su propia memoria, sus propios procesadores, tarjetas de red, su propio sistema operativo y sus propios discos.

Los dominios pueden ser creados y removidos sin interrupción de otros dominios dentro de la misma plataforma. Pueden crearse tantos dominios como queramos, aunque solo 16 de ellos pueden estar funcionando a la vez.

Configuración actual del WEB de la UNAM

Después de haber visto las características principales de la plataforma SUN Ultra-Enterprise-10000, veremos la forma en que están configuradas las dos plataformas de la UNAM sobre las que trabajaremos. La primera plataforma que analizaremos es la que contiene el servidor central de WEB de la UNAM.

Actualmente el servidor web de la Universidad Nacional Autónoma de México se encuentra configurado en una plataforma SUN Ultra-Enterprise-10000, la cual trabaja con Sistema Operativo UNIX en su variante de Solaris. Es importante mencionar que además del servidor web se encuentran configurados otros dos servidores o dominios dentro de la misma. Físicamente el equipo del que hablamos está ubicado en Ciudad Universitaria en el Centro de Cómputo de la Dirección General de Servicios de Cómputo Académico (DGSCA).

Como ya vimos con anterioridad una plataforma de este tipo debe tener un nombre, en este caso el nombre con que se denomina a la plataforma donde está el servidor WEB es silicio, veamos sus características.

Nombre de plataforma	Silicio
Número de dominios o servidores	3
Tarjetas de sistema	4
Procesadores	12 a 334 MHZ
Tarjetas de entrada/salida de datos	12
Capacidad de memoria RAM	5 GB
Tarjetas de control	1
Tarjetas de soporte	2
Fuentes de potencia	5
Ventiladores	18

Tabla 1.1. Características de hardware de la plataforma silicio

Está plataforma solo cuenta con una SSP, la cual es una Ultra-5, y tiene las siguientes características:

Nombre de la SSP	litio
Sistema operativo	Solaris versión 2.7
Versión de software de SSP	SSP 3.3.0

Tabla 1.2. Características de la SSP de la plataforma silicio

A su vez la plataforma llamada silicio tiene configurados en su interior 3 dominios o servidores:

Nombre Dominio	Tarjetas de sistema	Posición de las tarjetas de sistema	Sistema Operativo	Número de Procesadores	Memoria RAM	Tarjetas de Entrada/Salida
kripton	1	0	Solaris 2.7	4	2 GB	3
dragón	1	1	Solaris 2.6	4	2 G B	3
cobalto	2	2 y 3	Solaris 2.7	4	1 GB	6

Tabla 1.3. Características de los dominios de la plataforma silicio

A continuación se muestra una gráfica de cómo está distribuida la plataforma silicio. Como podemos observar la plataforma no está llena de hardware, además la gráfica nos indica en que tarjetas de sistema están configurados los dominios que acabamos de mencionar. Como vimos en las tablas anteriores el servidor web se encuentra configurado en la tarjeta de sistema número 1 y tiene conectados un arreglo de discos D1000 y 2 arreglos de discos A5000. En el capítulo 3 veremos más a detalle las características de estos arreglos, por ahora solo basta decir que un arreglo de discos A5000 o D1000 tiene varios discos duros en su interior.

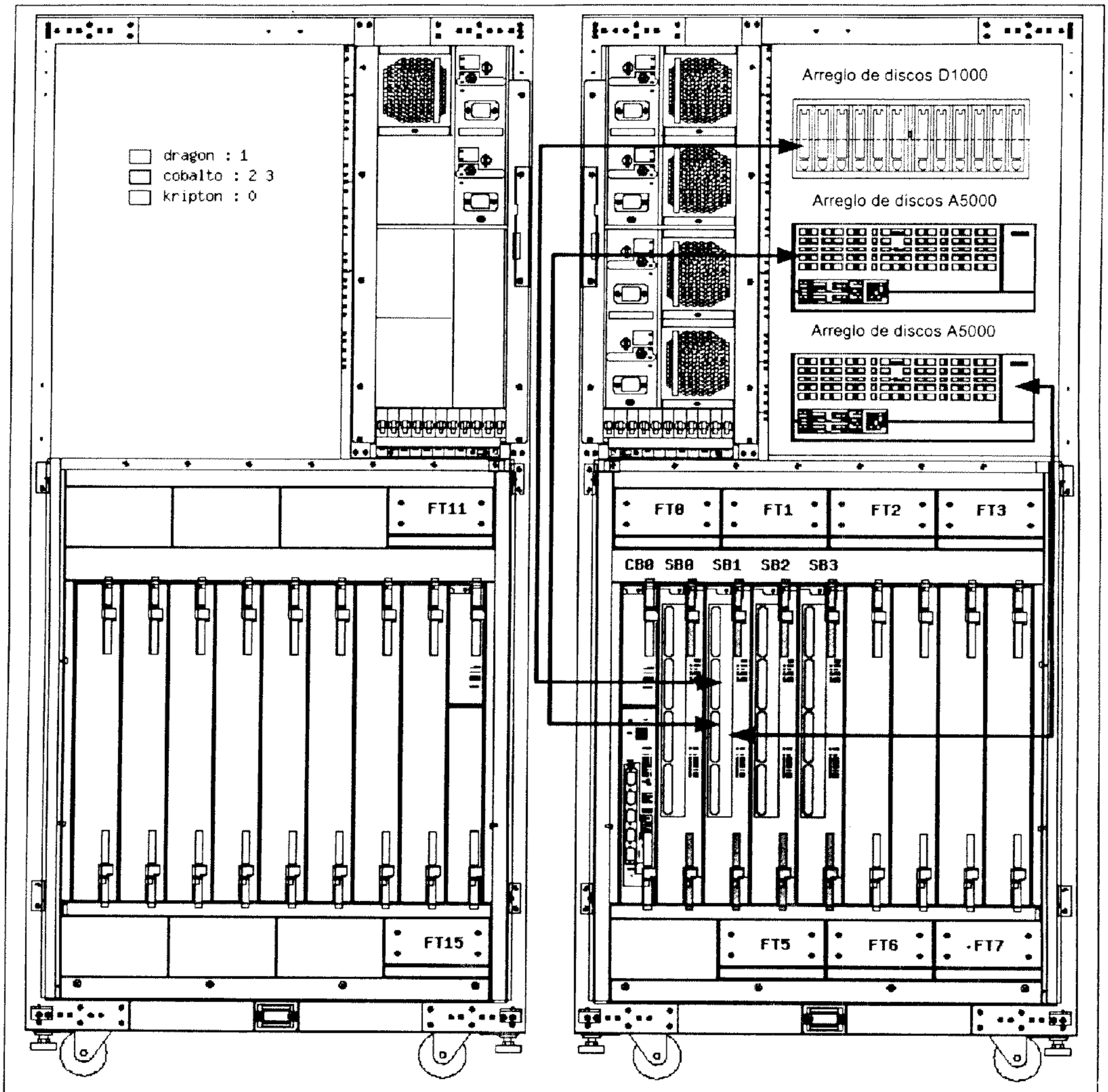


Figura 1.6. Vista frontal y trasera de la plataforma silicio

Configuración del servidor de replicación de datos

La otra plataforma que utilizaremos en este trabajo, sobre todo en el capítulo 4 de replicación de datos es también una máquina SUN Ultra-Enterprise-10000. Esta plataforma esta dedicada más que nada a la investigación de algunas áreas de la UNAM, sin embargo existen recursos que se desperdician, en particular en el servidor llamado newton, por lo que trataremos de aprovechar esto haciendo replicación de datos de archivos del servidor web hacia éste último. Está plataforma está localizada físicamente en el centro de computo de Pitágoras en la colonia del Valle del D.F.

Sus características son las siguientes:

Nombre de plataforma	científicos
Número de dominios o servidores	3
Tarjetas de sistema	3
Procesadores	12 a 400 MHZ
Tarjetas de entrada/salida de datos	12
Capacidad de memoria RAM	6 GB
Tarjetas de control	1
Tarjetas de soporte	2
Fuentes de potencia	5
Ventiladores	18

Tabla 1.4. Características de hardware de la plataforma científicos

Está plataforma también tiene solo una SSP y es una máquina Ultra-5, tiene las siguientes características:

Nombre de la SSP	Pitágoras
Sistema operativo	Solaris versión 2.7
Versión de software de SSP	SSP 3.3.0

Tabla 1.5. Características de la SSP de la plataforma científicos

Pitágoras también tiene tres dominios configurados:

Nombre Dominio	Tarjetas de sistema usadas	Posición de las tarjetas de sistema	Sistema Operativo	Número de Procesadores	Memoria RAM	Tarjetas de Entrada/Salida
Newton	1	0	Solaris 2.7	4	2 GB	4
Einstein	1	1	Solaris 2.7	4	2 G B	4
Euler	1	2	Solaris 2.7	4	2 GB	4

Tabla 1.6. Características de los dominios de la plataforma científicos

A continuación se muestra la gráfica correspondiente a la plataforma cientificos, en esta plataforma el dominio que nos interesa es newton, el cual, utiliza un arreglo de discos D1000 y 2 arreglos A5200. De esta forma ya tenemos un panorama general de lo que es la disponibilidad y de los servidores que vamos a utilizar.

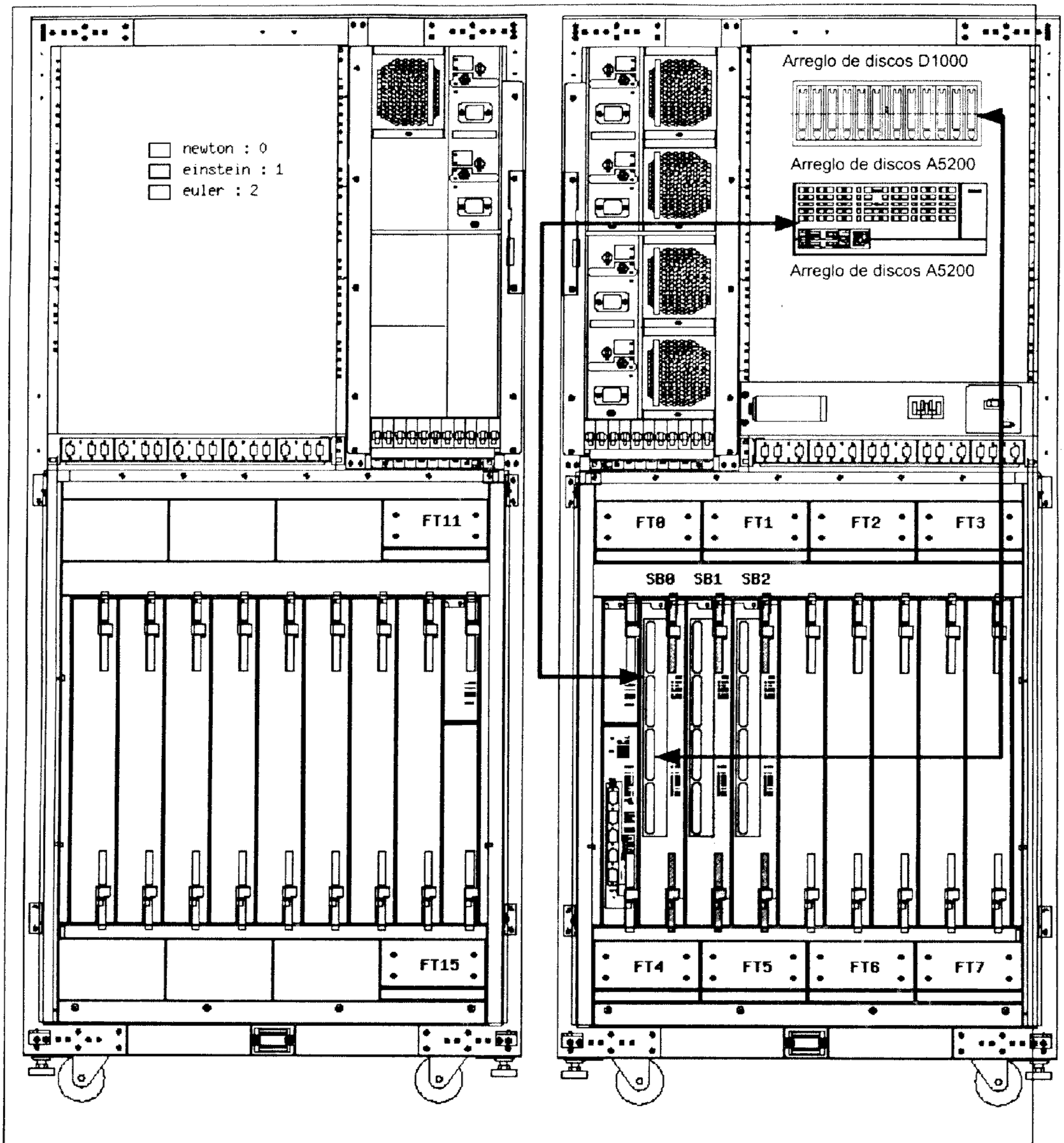


Figura 1.7. Vista frontal y trasera de la plataforma científicos

Capítulo 2. Elaboración de Respaldos

Los respaldos de información son el corazón de cualquier diseño de sistemas de misión crítica. Realizados de forma correcta representan la última opción de recuperación de información contra desastres naturales o borrado accidental de datos o de cualquier otro tipo. En muchas ocasiones los respaldos son necesarios simplemente para reiniciar una máquina que ha sufrido corrupción de datos y que no puede reanudar operaciones.

En este capítulo trabajaremos con los respaldos de información, comenzaremos con el concepto de respaldo, y veremos los tipos de respaldos que se utilizan hoy en día, así como los calendarios para respaldar. También hablaremos de los dispositivos y medios de respaldo y algunas recomendaciones para respaldar datos. Finalmente analizaremos la forma en que se hacen los respaldos en el servidor web de la UNAM y en el servidor que utilizaremos para replicar datos que se utilizará en el capítulo cuatro, y haremos una propuesta e implementación de un nuevo esquema de respaldos.

Concepto de Respaldo

Existen múltiples formas de perder información. Los errores en el software frecuentemente corrompen archivos. Los usuarios de los sistemas borran los archivos accidentalmente. Los intrusos y usuarios resentidos borran información intencionalmente. Adicionalmente, los problemas de hardware y los desastres naturales pueden poner fuera de operación sistemas completos. En la mayoría de las empresas la información almacenada en los sistemas de cómputo es mucho más valiosa que los sistemas mismos, además es mucho más difícil de recuperar. Por lo tanto una de las tareas más importantes y también más tediosas de los administradores de sistemas es la realización de respaldos de la información.

Un **respaldo** significa hacer una copia de datos o archivos en un medio magnético como pueden ser cintas o discos.

Realizar adecuadamente los respaldos permite al administrador recuperar un sistema de archivos o una parte de él al estado en que se encontraba en el momento en que se hizo el último respaldo. De ahí la importancia de realizar esta actividad de manera adecuada.

Los respaldos están relacionados con conceptos como: particiones de disco, sistemas de archivos y cintas. En seguida definimos de manera breve cada uno de estos conceptos.

Partición de disco. En el sistema operativo Solaris, un disco duro se puede dividir hasta en 8 partes, a cada una de esas partes se le llama partición de disco.

Sistema de archivos. Un sistema de archivos es una partición de disco que tiene definida una estructura de archivos y directorios.
Cinta. Es un medio magnético que sirve para hacer respaldos de información.

Tipos y Niveles de Respaldos

Existen principalmente tres tipos de respaldos: completo, incremental acumulativo e incremental diferencial.

Respaldo Completo

Un respaldo completo es aquel en el que cada bit de los datos que están en disco es copiado a un dispositivo magnético. Aunque son extremadamente confiables los respaldos completos pueden ser muy lentos al copiar todos los datos, a menos de que se respalden solo los datos que han cambiado desde el última copia completa. Un respaldo incremental solo copia los archivos o datos que han cambiado desde el último respaldo realizado.

En realidad hay dos diferentes clases de respaldos incrementales:

Respaldo incremental acumulativo

En este tipo de respaldo, se copian solo los datos que han cambiado desde la última vez que se hizo un respaldo completo. Al respaldar menos datos que un respaldo completo obviamente es más rápido.

Respaldo Incremental Diferencial

En este tipo de respaldo, se copian solo los datos que han cambiado desde la última vez que se hizo un respaldo incremental acumulativo. Esta clase de respaldo es mas rápido que el respaldo completo y que el incremental acumulativo ya que copia menos datos.

Niveles y Calendarios de Respaldos

Un nivel de respaldo es aquel que nos sirve para determinar que información se deberá incluir al copiar nuestra información a cinta o a disco. Con comandos del sistema operativo UNIX en su variante de Solaris podemos tener hasta diez niveles de respaldos: 0 al 9. El nivel 0 es siempre un respaldo completo y es el nivel más bajo de respaldos, por consiguiente el nivel más alto es el 9. Del nivel 1 al 9 son respaldos incrementales los cuales respaldan los archivos que han cambiado desde el respaldo mas reciente en un nivel mas bajo. Un calendario de respaldos nos sirve para ilustrar cómo vamos a respaldar nuestra información, es decir nos muestra los dias, las semanas, los meses y los tipos de respaldos que estamos utilizando al respaldar nuestros datos.

Para entender mejor lo anterior, veamos algunos ejemplos.

1.- Calendario de respaldos modelo 9 a 5. Respaldos incrementales acumulativos diarios e incrementales acumulativos semanales.

MENSUAL	0

	Lunes	Martes	Miércoles	Jueves	Viernes
Semana 1	9	9	9	9	5
Semana 2	9	9	9	9	5
Semana 3	9	9	9	9	5
Semana 4	9	9	9	9	5

Tabla 2.1. Primer ejemplo de un calendario de respaldos.

Para llegar al calendario que se muestra arriba, partimos de la idea de hacer un respaldo completo, es decir de nivel 0, al inicio de cada mes. Pero el hacer un respaldo cada mes no nos garantiza el poder recuperar información de manera confiable, es decir, si en el transcurso de la segunda semana se crean ciertos datos y si en la tercera semana se borran accidentalmente y queremos recuperarlos no podremos hacerlo porque el respaldo que hicimos al inicio del mes no contiene dichos datos, así que optamos porque cada viernes se hiciera un respaldo incremental acumulativo, de esta manera para todos los viernes tendríamos un respaldo de todos los datos acumulados que han cambiado desde el respaldo de nivel mas bajo, en este caso usamos el 5 para los viernes (podríamos haber utilizado cualquier número entre 1 y 8), el respaldo de nivel mas bajo es el 0 que se hizo al inicio del mes, con esto podríamos solucionar un problema como el que mencionamos anteriormente. Pero qué pasa si por ejemplo un martes, creamos ciertos datos y en el transcurso de uno o dos

días se pierden por algún motivo?, al querer recuperar nuestros datos perdidos no lo podríamos hacer, simplemente porque nuestro respaldos semanal y mensual no los contienen, así que para evitar eso se definieron respaldos incrementales acumulativos diarios, utilizamos el 9 para que el respaldo que se haga todos los lunes respalde también los datos del nivel mas bajo que es el 5, es decir el respaldo del lunes incluirá los datos que cambiaron del viernes a ese momento.

Con este calendario se logra lo siguiente:

Cada día de la semana se acumulan los archivos que cambiaron desde el final de la semana anterior o desde el nivel 0 inicial de la primera semana.
El respaldo de cada viernes contiene los archivos que cambiaron desde el primer nivel 0.
Para cada viernes el nivel bajo mas cercano es el nivel 0 realizado al principio del mes. Por tanto cada viernes el respaldo contiene todos los archivos que cambiaron durante el mes hasta ese momento.

2. Calendario de respaldos modelo 9 a 2-3-4-5. Respaldos incrementales acumulativos diarios e incrementales diferenciales semanales.

MENSUAL	0

	Lunes	Martes	Miércoles	Jueves	Viernes
Semana 1	9	9	9	9	2
Semana 2	9	9	9	9	3
Semana 3	9	9	9	9	4
Semana 4	9	9	9	9	5

Tabla 2.2. Segundo ejemplo de un calendario de respaldos

Con este calendario:

Cada día de la semana se acumulan todos los archivos que se modificaron desde el inicio de semana (o para la primera semana desde el nivel más bajo, el nivel 0).
El respaldo de cada viernes contiene todos los archivos que cambiaron durante esa semana.

Alternativas de Respaldos

Respaldos fiscos y lógicos

Dentro del amplio mundo de las soluciones de respaldos encontramos dos grandes vertientes: los respaldo físicos y los respaldo lógicos. Los respaldos físicos respaldan bit por bit una base de datos completa, una partición de disco o un sistema de archivos al medio de respaldo que generalmente es una cinta. Los respaldos lógicos se preocupan por respaldar la estructura lógica de una base de datos o de archivos individuales de un sistema de archivos. Ambos tipos de respaldos tienen sus ventajas y desventajas.

Los respaldos físicos son mucho más rápidos que los respaldos lógicos, ya que la información se respalda secuencialmente y por consiguiente, se aprovecha la velocidad del dispositivo de respaldo. Su principal desventaja radica en que lo que es respaldado únicamente es visto como una sola entidad, es por esto, que en un ambiente de bases de datos no es muy utilizado este tipo de respaldos, por ejemplo, en determinado momento no se podría restaurar una sola parte de una base de datos. Normalmente este tipo de respaldos son utilizados para hacer un respaldo completo de un sistema de archivos, pero se debe tener especial cuidado con la integridad de los datos que serán respaldados.

En contraste, un respaldo lógico se preocupa por leer la estructura lógica de una entidad, una base de datos por ejemplo, estas estructuras lógicas son leídas una a la vez, a diferencia de los respaldos físicos, este tipo de respaldos son lentos, pero obtenemos el beneficio que mediante este tipo de respaldos sabremos cuando fue la última fecha de modificación de cada archivo y así sabremos si un archivo fue o no modificado desde que el último respaldo se hizo. Los respaldos lógicos solo pueden ser más rápidos que los respaldos físicos, cuando se implementan esquemas de respaldos incrementales. Los respaldos lógicos requieren de cierto conocimiento sobre el contenido y la estructura de lo que se está respaldando.

Respaldos en caliente

Lo que se está usando actualmente para minimizar los tiempos de realización de respaldos, es hacer respaldos en caliente, este tipo de respaldos no necesitan interrumpir el servicio para llevarse a cabo. Son ampliamente utilizados en respaldos de bases de datos y en una menor medida en sistemas de archivos. El gran problema que han solucionado los respaldos en caliente, es garantizar la consistencia de los datos.

Normalmente, durante un respaldo en caliente o también conocido como respaldos en línea, las bases de datos o los sistemas de archivos son puestos en un estado donde las escrituras a disco están bloqueadas y éstas transacciones para no ser perdidas son enviadas a un archivo de registro que guarda dichas transacciones. Una vez que el respaldo ha sido concluido, la información que se encuentra en el archivo de registro pasa a un proceso que se encarga de actualizar los datos en la base de datos o en el sistema de archivos. En algunas ocasiones cuando la información a respaldar es mucha, se experimenta algo de carga de trabajo en los sistemas. Es por esto que se deben planear muy bien los horarios de tiempo requeridos para cada área de información a respaldarse.

El problema de consistencia que se suscita al respaldar sistemas de archivos, se presenta al ocupar un esquema de dos fases, una primera fase donde se respalda la estructura propia de la información a la cual llamamos superbloque, es decir, el tamaño de archivo, última fecha de acceso, última fecha de actualización, etc. y una segunda etapa donde se respalda propiamente el contenido del archivo. Si el archivo cambia en el transcurso de estas dos etapas habrá una inconsistencia entre la información que se encuentra en el superbloque y la información que se ha respaldado y por lo consiguiente tendremos un respaldo defectuoso, el cual seguramente no podremos restaurar cuando sea necesario.

Almacenamiento Jerárquico

Una tarea interesante para un administrador, es el buscar en su sistema, archivos de usuarios que no han sido accedidos por lo menos una vez en dos semanas, en un mes, en tres meses, en seis meses y en un año por citar algunos periodos de tiempo. Considérese lo siguiente: estos archivos no han sido tocados en seis meses y más aún utilizan espacio, tiempo, y dinero para ser respaldados, ¿Qué podemos hacer para recapturar los recursos que estos archivos consumen?

De esto es de lo que se encargan las herramientas de almacenamiento jerárquico o HSM, por sus siglas en inglés (Hierarchical Storage Management), este tipo de herramientas ampliamente usadas en el mercado proveen al administrador de herramientas para catalogar la información. Un HSM procesa y examina los accesos más recientes a un archivo o a un disco, y basado en reglas que el administrador establece de acuerdo a sus necesidades, mueve los archivos de menor uso a medios menos caros, más permanentes y generalmente más lentos. Estos medios pueden ser unidades de cintas dedicadas, CDs regrabables o robots de discos magneto-ópticos.

Una vez movida la información, el usuario normalmente solo accederá a ella en casos excepcionales, cuando esto sucede el sistema automáticamente usa el dispositivo que contiene la información y la mueve nuevamente a un disco local, una vez que la información esta en el disco local el ciclo vuelve a repetirse.

2.3.4 Espejos Triples

Existen paquetes de software que se encargan de manejar los discos, los cuales trabajan con discos virtuales que se generan a partir de particiones de disco, los cuales contienen copias de datos llamadas espejos. Algunos fabricantes de software han implementado modelos de respaldos los cuales, involucran la creación de una tercera copia de datos, es decir la creación de un tercer espejo, en el cual se puede albergar una base de datos o un sistema de archivos. Cuando se requiere hacer un respaldo, el tercer espejo puede ser separado de los otros dos. Las transacciones continúan en el espejo principal, mientras que el respaldo se realiza utilizando el tercer espejo. Cuando el respaldo es concluido, el tercer espejo debe ser nuevamente sincronizado con las otras dos caras del mirror.

El objetivo principal de hacer un respaldo con esta tecnología radica en que el sistema de archivos o la base de datos no tiene actividad, por lo cual, se garantiza la integridad de la información a respaldar. La principal desventaja de los espejos triples es su costo, ya que se requiere disco adicional para cumplir con su objetivo. Para hacer un espejo normal se requiere de un 100% adicional de disco y para hacer un tercer espejo se requiere un 200% adicional, es decir, para hacer un espejo de un sistema de archivos de 60 Gigabytes requeriríamos de 120 Gigabytes de disco y para hacer un tercer espejo requeriríamos 180 Gigabytes de disco para cumplir con nuestro objetivo. Aunado a esto, el acto de sincronización requiere de una sobresaturación de los dispositivos de entrada/salida de datos principalmente de los discos y de los CPUs.

La otra variación implica la creación del tercer espejo antes de hacer el respaldo y destruirlo después de que este termine. Este variante también implica una gran carga en los dispositivos de entrada/salida y en los CPUs.

Algunos fabricantes de cajas de discos como la empresa estadounidense EMC, en sus arreglos de discos, utilizan herramientas para realizar la sincronización de los discos, eliminando el impacto de entrada/salida de datos y de CPUs en los servidores, así mismo, realizan más rápido la sincronización de la información.

Foto instantánea

Los sistemas de archivos no son más que una colección de bloques de datos en un disco, con apuntadores hacia ellos, los cuales son recuperados después de que se ha escrito a disco. En los respaldos tipo snapshot o de foto instantánea los apuntadores son copiados a localidades alternas y por lo tanto siempre apuntan a la información original, es decir, el nuevo conjunto de punteros siempre apuntan a una copia de datos consistente.

Como consecuencia de este proceso, nosotros podemos realizar respaldos de sistemas de archivos confiables y continuar recibiendo datos al mismo tiempo.

Dispositivos y Medios para Respaldar

Considerando que las fallas que se presentan en un sistema pueden afectar distintos componentes de hardware a la vez, los respaldos deben hacerse en algún tipo de medio removible. En general las organizaciones debieran conservar sus respaldos fuera del centro de cómputo, de tal forma que un siniestro no pueda destruir la información original y su respaldo.

Hoy en día existen en el mercado varios tipos de dispositivos o cartuchos de cinta en los cuales podemos respaldar información. A continuación se mencionan los más comunes.

Cartuchos de cinta QIC

El dispositivo o manejador QIC de cartuchos de cinta son normalmente encontrados en las computadoras llamadas estaciones de trabajo. Los cartuchos QIC (Quarter In Cardridge) son caros y son particularmente adecuados para hacer respaldos, puesto que aún un sistema de archivos pequeños se puede llevar varias cintas. Sin embargo las cintas QIC son empleadas con frecuencia para distribuir software, y son excelentes para intercambiar datos entre distintas plataformas. El cartucho QIC más común empleado con manejadores SUN era el QIC-150 el cual podía almacenar 150 MB. Recientemente a salido un modelo que puede almacenar hasta 2.5 GB.

Cartuchos de cinta de 8 mm

Existen varias marcas de dispositivos de cintas que graban a cartuchos de 8 mm (formato pequeño). El formato original almacenaba aproximadamente 2 GB y posteriormente 5 GB, sin embargo formatos mas recientes almacenan 7 GB y 20 GB. Utilizando compresión de datos se pueden almacenar hasta 14 GB y 40 GB.

Estos sistemas son relativamente rápidos y permiten realizar un respaldo completo sin intervención del operador debido a su gran capacidad de almacenamiento. Las cintas de 8 mm son compactas, reducen la necesidad de espacio para almacenaje físico y facilitan la tarea de guardarlas fuera del centro de cómputo.

La principal desventaja de las cintas de 8 mm es que los mecanismos de los dispositivos que leen y escriben en ellas tienden a desalinearse después de cierto tiempo de uso, requiriendo una reparación que no siempre es de bajo costo. Para limpiar esta unidad de cintas se debe utilizar un cartucho especial de limpieza que solo sirve para usarse un número limitado de veces.

Cartuchos de 4 mm

Muchos fabricantes suministran con el equipo de cómputo dispositivos que emplean cintas DAT (Digital Audio Tape). Técnicamente el estándar DAT para almacenamiento de datos es llamado DDS (Digital Data Storage). Las cintas DAT son los medios magnéticos más pequeños que se encuentran disponibles. Estos cartuchos almacenan 2 GB (DDS), 4 GB (DDS2) y 12 GB (DDS3); sin embargo empleando compresión de datos pueden almacenar 4 GB, 8 GB y 24 GB respectivamente. El acceso a los datos es mucho más rápido que en las cintas de 8 mm y los dispositivos que las manejan son más confiables. De igual forma que en el caso de las unidades de 8 mm se requiere un cartucho especial para realizar limpieza.
2.4.4 Cartuchos DLT

La tecnología DLT (Digital Lineal Tape) proporciona una solución ideal para respaldos de alta capacidad y de alto rendimiento con un nivel de integridad de datos excepcional. La diferencia fundamental de esta tecnología con relación a la tecnología que utilizan los cartuchos de 8 mm y los de 4 mm es que lee y escribe múltiples canales simultáneamente. Los dispositivos de manejo DLT dividen los medios magnéticos en pistas paralelas y horizontales y graban los datos pasando la cinta varias veces sobre una cabeza de lectura-escritura estacionaria.

Los cartuchos DLT almacenan típicamente 20 GB y 35 GB en los modelos disponibles de SUN Microsystems, sin embargo suponiendo una relación de 2 a 1 se pueden tener 40 GB y 70 GB almacenados en un solo cartucho.

Autocargadores de cintas

Adicionalmente a los dispositivos unitarios que se han mencionado hasta el momento, existen los llamados autocargadores de cintas. Un autocargador de cintas es simplemente una caja que contiene ranuras llamadas biblioteca en donde se guardan las cintas, un número determinado de dispositivos que se encargan de leer y escribir en ellas y un componente mecánico comúnmente llamado brazo que tiene como función mover las cintas entre las ranuras y los dispositivos de lectura-escritura para respaldar o recuperar información, además de permitir sacar de la caja cintas que ya no se usan y/o meter cintas nuevas. Los autocargadores de cintas están disponibles para varios tipos de medios, incluyendo 4 mm, 8 mm y DLT. Normalmente las cajas de cintas se venden acompañadas de un software de respaldos que permite el manejo del brazo.

La utilización de las cajas de cintas conduce a la posibilidad de almacenar cantidades tan grandes de información como 140 GB (280 GB con compresión) en un solo autocargador que tiene bibliotecas con 7 ranuras DLT o bien cantidades aún mayores al existir bibliotecas con hasta 100 ranuras para cintas DLT empleando cartuchos de 35 GB. Estas bibliotecas ciertamente son apropiadas para sistemas que respaldan grandes cantidades de información.

Recomendaciones en el manejo de cintas

Muchas clases de medios utilizan partículas magnéticas para almacenar datos. Estos medios son susceptibles a ser dañados por campos eléctricos y magnéticos. Algunos riesgos específicos que deben evitarse son:

Bocinas que contienen electroimanes grandes; no almacene cintas encima o cerca de ellas. Incluso las bocinas pequeñas para equipos multimedia pueden ser riesgosas.
Los transformadores son esencialmente electroimanes. Habitualmente se encuentran en paredes y suministros de energía.

Los discos duros y las unidades de cinta tienen motores y cabezas magnéticas, y sus gabinetes frecuentemente están sin blindaje. Los dispositivos de lectura-escritura en gabinetes metálicos son más seguros.
Los monitores utilizan transformadores y alto voltaje. Muchos monitores conservan una carga eléctrica aún después de ser apagados. Los monitores de color son los peores. Nunca almacene cintas sobre un monitor.
Los detectores de metal, especialmente aquellos que se encuentran en los aeropuertos, pueden afectar severamente y destruir la información. Si es necesario transportar cintas en un avión, es recomendable pasarlas consigo y someterlas a revisión manual.

La exposición prolongada a la radiación terrestre afecta los datos en medios magnéticos, acortando su vida útil y volviéndose imposible leerlos después de algún tiempo. La mayoría de los medios se conservan durante tres años aproximadamente, si se planea almacenar datos por un tiempo más largo es recomendable utilizar medios ópticos o bien volver a grabar la información.

Planeación de respaldos y reglas Básicas

Planeación de una arquitectura de respaldos

La planeación de la capacidad es un elemento sumamente importante al implementar una arquitectura de respaldos. En esta etapa una gran cantidad de variables deben ser tomadas en cuenta; algo importante en está etapa es minimizar el impacto de los cuellos de botella que pudiera tener nuestro sistema.

La persona encargada de verificar la capacidad integrada del sistema es responsable de la selección de hardware y software necesaria, para que las operaciones de respaldo y recuperación de información en el centro de cómputo se lleven de la mejor forma posible.

El primer punto a considerarse es la cantidad de información a respaldarse y debe ponerse especial atención a lo siguiente:

- Disponibilidad de los datos.
- Cómo van a ser respaldados los datos a través de la red.
- Políticas de respaldo.
- Requerimientos para realizar una recuperación de información.

El segundo punto debe considerar los requerimientos del servidor de respaldos:

- Red.
- Discos.
- Dispositivos de respaldo.

Reglas básicas para hacer respaldos

Existen infinidad de manuales y guías completas para hacer respaldos, pero a su vez también existen puntos sustanciales, los cuales siempre debemos tener en mente en el diseño de un ambiente correcto de respaldos y así sacar la mayor ventaja posible de los respaldos que nosotros hagamos:

Los espejos no reemplazan a los respaldos. En la práctica esto es un hecho y ha dejado de ser un mito. Los espejos nos protegen contra las fallas en los discos, pero de ninguna manera nos protegen contra un archivo borrado, si un archivo es borrado (o está corrupto) en un lado del espejo también será borrado del otro lado del espejo. La forma más común de recuperar el archivo es mediante un respaldo.
Los respaldos no solo se utilizan después de una catástrofe. Las catástrofes ocurren, pero es más probable que algún usuario accidentalmente borre o dañe un archivo o inclusive un directorio completo.

Regularmente debemos probar nuestra habilidad para recuperar información. Los respaldos realmente son una herramienta maravillosa, pero si no podemos leer su contenido no sirven de nada. Si nos es posible se debe probar cada respaldo que nosotros hagamos y mejor aún cada respaldo debe probarse regularmente.

Las cabezas de lectura-escritura de los dispositivos de respaldos deben estar siempre limpias. Una cabeza de respaldo sucia puede mandarnos un mensaje de "respaldo exitoso", cuando en realidad respaldamos basura. Los fabricantes nos recomiendan que las cabezas deben ser limpiadas antes y después de haberse hecho un respaldo.

Atención con el MTBF de las cintas. Si el fabricante nos sugiere que la cinta se puede reciclar únicamente 3 veces, solo 3 veces debemos utilizarlas. Existen herramientas de respaldo que llevan una base de datos con las reutilizadas de una cinta y que nos informaran cuando debe reemplazarse.

Las cintas pueden no servir después de algunos años. No debemos asumir que una cinta hecha hace cinco años será confiable el día de hoy. El material magnético con el que son hechas puede degradarse.
Haz dos copias de respaldo críticos. Realmente es mucho más barato gastar dos o tres cintas mas, que pagarle a alguien para que trate de recrear los datos que se encuentran en las cintas, además esta segunda copia debe estar fuera de nuestro centro de cómputo en un lugar seguro.

El realizar un buen plan para protegernos de un desastre no es tarea fácil, ésta nos puede llevar meses e inclusive años para perfeccionarla. Existen seis pasos para diseñar un plan que nos proteja en caso de desastres y el orden en que se vayan ejecutando es muy importante:

1.- Define una pérdida aceptable para tu empresa. Antes de desarrollar tu esquema, debes definir cuánta información estás dispuesto a perder. Esto te ayudará a decidir cuanto tiempo y cuanto dinero estas dispuesto a gastar.

2.- Respalda todo. De ser posible debes asegurarte que has respaldado todo, absolutamente todo y esto incluye datos, procedimientos e instrucciones necesarias para regresar al mismo punto antes del desastre. Este es un punto muy difícil de llevar a cabo en la práctica ya que en las organizaciones donde existe un gran volumen de información los costos de las cintas y los dispositivos de cintas se incrementarán en gran escala y además el tiempo de realización de un respaldo completo se elevaría y con ello tendría lugar una degradación del funcionamiento de nuestro servidor.

3.- Organiza todo. Debes de organizar los respaldos que sean multivolumen.

4.- Protégete contra desastres. Mucha gente piensa que solo nos debemos proteger contra desastres naturales, si bien es cierto que los desastres naturales ocurren, también es cierto que debemos protegernos contra los posibles desastres de nuestra zona.

5.- Documenta todo lo que has hecho. Debes de documentar todo el plan para recuperarte de un desastre y así en caso de que éste ocurra puedes seguir los pasos que hayas establecido, existen diferentes formatos los cuales te pueden servir, por ejemplo: HTML, pdf, postcript, algún procesador de textos e inclusive fotocopias.

6.- Prueba todo. Un plan para recuperarte de un desastre y que no haya sido probado no es un plan, realmente a ninguno de nosotros nos gustaría estar en medio de un problema y descubrir que hemos olvidado algunos pasos críticos. El secreto para recuperarte de un desastre real es que hayas probado que tu plan trabaje correctamente, al hacer esto descubriremos posibles errores, los cuales pueden corregirse.

Recuperación de la información

Muchas de las cosas que podemos hacer para incrementar la velocidad de los respaldos ocasionarán que las recuperaciones de información sean más lentas. A continuación mencionaremos algunas tareas que podemos realizar para incrementar la velocidad de las recuperaciones, aunque algunas de ellas incrementarán la duración de los respaldos.

Hacer menos respaldos incrementales

Por supuesto el hacer menos respaldos incrementales significa hacer más respaldos completos, y más respaldos completos significan tiempos más largos. Probablemente la mejor recomendación es hacer incrementales acumulativos una vez a la semana o dos. De esta manera solo necesitaremos restaurar información de dos cintas después de que haya ocurrido una falla. Si nosotros alternamos entre dos y tres cintas de respaldos incrementales acumulativos, el costo de la pérdida de una de esas tres cintas también será reducido. Si ponemos todos nuestros incrementales en una cinta, si esa cinta se pierde o no es posible leerla, se perderá toda la información respaldada que contenga esa cinta.

Tener las cintas disponibles

Además de tener una copia reciente de la información fuera del centro de cómputo se debe tener otra copia en cinta cerca de las librerías de respaldos y además de preferencia deben estar ordenadas para que cuando se necesite alguna el tiempo de búsqueda y de inserción en los dispositivos sea el mínimo.

Incrementar la velocidad de escritura a disco

Algunos arreglos de discos utilizan memoria adicional a partir de discos físicos para capcurar largos bloques de datos antes de que éstos sean escritos a disco. Si las escrituras están basadas en discos, es recomendable utilizar un sistema de archivos para optimizar el funcionamiento.

Utilizar un camino más rápido de la cinta al disco

Redes dedicadas de alta velocidad moverán los datos más rápido a través de la red. Pueden ser conexiones directas Ultra Wide SCSI, 100 Base-T (100 megabits por segundo) o FCAL (100 megabytes por segundo).

Tratar de no recuperar toda la información a menos que sea necesario. El mejor camino para agilizar la recuperación de la información es no tener que restaurar todo.

Esquemas de respaldos de los servidores dragón y newton

Cómo vimos en el capítulo 1 los servidores que utilizaremos en este trabajo tienen asignado un nombre. En este punto vamos a analizar el esquema de respaldos del servidor WEB de la UNAM cuyo nombre es dragón, y también el esquema de respaldos del servidor que usaremos para replicar datos en el capítulo 4 y que se conoce como newton.

Esquema de respaldos del servidor WEB de la UNAM

Actualmente la información del servidor dragón, es respaldada a cartuchos de cinta DDS 3 de 4 mm a través de la red, a un dispositivo manejador de cintas del mismo tipo que se encuentra en una máquina Sun Enterprise-3500 en el centro de cómputo de la DGSCA, esto debido a que silicio, la plataforma donde se encuentra configurado dragón, no cuenta con ningún dispositivo de cinta para respaldar datos. Lo que se respalda a cinta son sistemas de archivos completos, podemos dividir estos sistemas de archivos en sistemas de archivos de sistema operativo y en sistemas de archivos de aplicaciones y de usuarios.

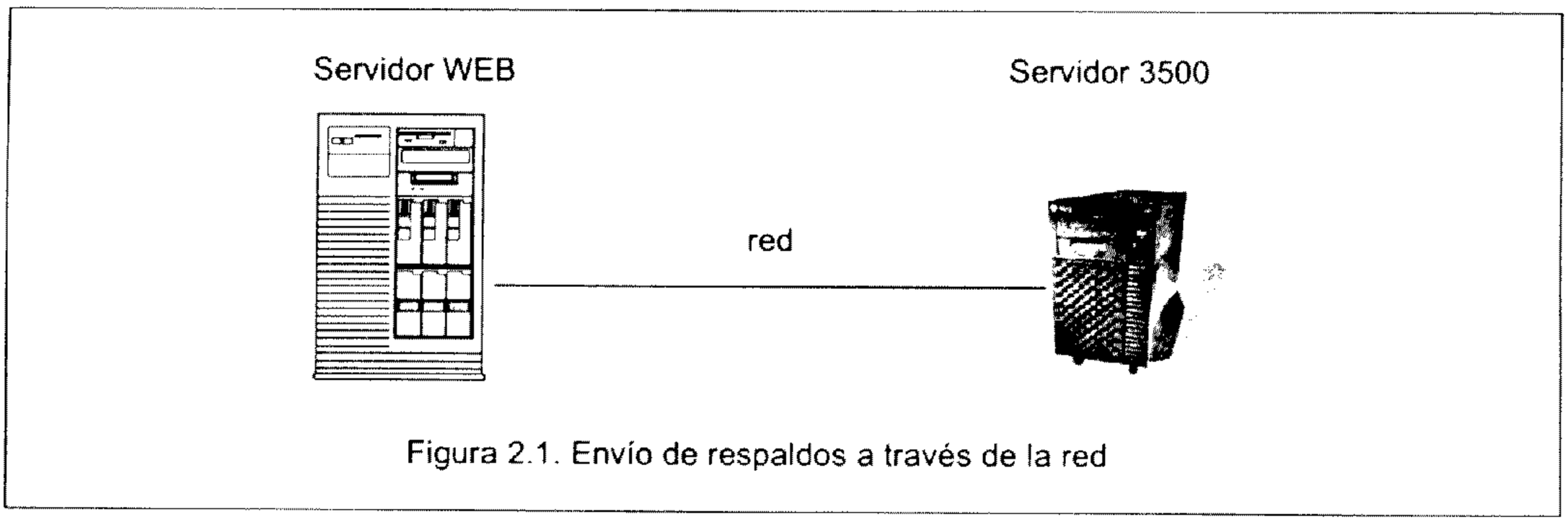


Figura 2.1. Envío de respaldos a través de la red

1. Sistemas de archivos de sistema operativo.

Estos sistemas de archivos contienen datos que son propios de sistema operativo e incluyen toda la información que se genera cuando se instala este último, archivos de configuración, estructura de los directorios, estructura de los discos, paquetes de software, y parches de software.

El sistema operativo de dragón está instalado en un disco que se encuentra divido en varias particiones a las cuales les corresponde un sistema de archivos determinado. Los sistemas de archivos que conforman el disco donde se encuentra instalado el servidor web de la UNAM y que se mandan respaldar se muestran a continuación, también se muestra su tamaño y el espacio utilizado.

Sistema de archivos	Tamaño en MB	Espacio utilizado en MB
/	617275	327903
/usr	4131866	2646600
/var	3099287	2162276
/opt	2056211	466484

Tabla 2.3. Sistemas de archivos de sistema operativo del servidor web de la UNAM

Para respaldar esta información se utiliza un sencillo programa escrito en shell de UNIX que se ejecuta de forma manual cada vez que se quiere respaldar la información, este hace uso de un comando de sistema operativo llamado ufsdump, veamos el contenido del script.

```
/usr/sbin/ufsdump  -0uf  /          132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /usr      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /var      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /opt     132.248.18.9:/dev/rmt/0n
```

Donde:

/usr/sbin/ufsdump — Es un comando de solaris para respaldar información.
0 ———————— Indica el nivel de respaldo, en este caso es de nivel 0.
u ———————— Actualiza un archivo de registro que contiene la fecha y el nivel de respaldo del sistema de archivo.

f ——————— Hace referencia al dispositivo al cual los archivos van a ser escritos, en este caso son mandados al dispositivo /dev/rmt/0n de la máquina Sun Enterprise-3500 que tiene la dirección IP 132.248.10.9

/, /usr, /var, /opt—— Son los nombres de los sistemas de archivos que van a ser respaldados.

2. Datos de aplicaciones y de usuarios

Además de los datos de sistema operativo existen también los datos de las aplicaciones, como son los del software que funciona como servidor web, los propios datos de las páginas web, así como, los datos de los usuarios y administradores de esas páginas. Por desgracia toda está información es demasiada para ser respaldada a unidades de cintas DDS, por lo que esta información no está siendo respaldada. Toda esta información se encuentra distribuída en los siguientes sistemas de archivos.

Sistema de archivos	Tamaño en MB	Espacio utilizado
/mirror/home	30945914	27291825
/home/log	30945914	12978909
/mirror/users00	6186810	4193040
/home/users01	6186810	3778354
/home/users02	6186810	3492786
/home/users03	6186810	5857917
/home/users04	3871954	2069039
/Sybase	6186810	2881346
/raid	3871954	1871249
/raid2	3871954	1100983
/usr/local/pgsql	1527116	1140770
/usr/Sybase	4743974	3615240
/home	30957590	28151616
/home/users00	6199998	3354000

Tabla 2.4. Sistemas de archivos de aplicaciones y usuarios del servidor web de la UNAM

El calendario de respaldos que se utiliza actualmente para respaldar los datos de sistema operativo de dragón es un respaldo de nivel 0 y se hace cada mes.

Calendario de respaldos modelo 0.
Respaldos completos mensuales.

MENSUAL	0

Esquema de respaldos del servidor newton

El servidor newton también respalda a través de la red a una máquina SUN Enterprise-3500 a cintas DDS-3 de 4 mm, ya que la plataforma "cientificos" tampoco tiene dispositivo de cintas para respaldar información. Como recordaremos este servidor se utiliza para algunas actividades de investigación de la UNAM y de cálculo esporádicamente. Dado que vamos a replicar datos del servidor web a este servidor, es de gran importancia tener un esquema adecuado de respaldos de información. Adicionalmente a los sistemas de archivos de sistema operativo solo se respalda un sistema de archivos más el cual contiene información de los usuarios y corresponde a /home como se puede ver en la siguiente tabla de datos.

Sistema de archivos	Tamaño en MB	Espacio utilizado en MB
/	1018191	91602
/usr	4131866	1447773
/var	4131866	1151518
/opt	1018191	225373
/home	2921895	1407124

Tabla 2.5 Sistemas de archivos de sistema operativo del servidor newton

Al igual que en dragón, en el servidor newton también hay información que por su tamaño no puede respaldarse a cinta. Para respaldar la información anterior se utiliza el siguiente programa, también se ejecuta de manera manual y el significado de su contenido es el mismo que el programa de dragón solo que varía el sistema de archivos /home y la máquina 3500 a la que se manda respaldar.

```
/usr/sbin/ufsdump  -0uf  /          132.248.10.21:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /opt     132.248.10.21:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /usr     132.248.10.21:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /var     132.248.10.21:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /home 132.248.10.21:/dev/rmt/0n
```

El calendario de respaldos que se utiliza actualmente para respaldar los datos de newton es también un respaldo de nivel 0 y se hace cada mes.

Calendario de respaldos modelo 0.
Respaldos completos mensuales.

MENSUAL	0

Problemática Actual

Analizando el esquema anterior, nos damos cuenta que tiene varias desventajas. Es cierto que se gasta muy poco dinero en cartuchos de cintas, pero al respaldarse la información cada mes la probabilidad de que no se puedan recuperar datos que se generaron y perdieron en ese lapso de tiempo entre un respaldo y otro se incrementa notablemente, y por lo tanto la disponibilidad de los datos es menor. Además como se está respaldando la información a otra máquina a través de la red, en el caso de que se corrompieran sistemas de archivos completos de sistema operativo o parte de ellos se llevarían un tiempo considerable en recuperar dicha información. Por lo tanto en el siguiente punto, tratamos de corregir esto mediante un nuevo esquema para hacer respaldos de información.

Implementación de un nuevo Esquema de Respaldos

Dada la problemática anterior, se propusieron e implantaron: un nuevo calendario de respaldos, la creación de un sistema de archivos para guardar respaldos y un disco de sistema operativo alterno. Es muy importante señalar que esto se hizo después de realizar la reestructuración de discos que se verá en el capítulo 3, ya que no hubiese tenido caso hacerlo antes pues no había espacio disponible y se tenían que hacer varios movimientos físicos de discos.

Nuevo Calendario de Respaldos

Sistemas de archivos de Sistema Operativo

Se diseñó e implantó un nuevo calendario de respaldos a cinta DDS-3 de 4 mm para los sistemas de archivos de sistema operativo de dragón y de newton.

Tomando en consideración que los datos de sistema operativo no cambian constantemente, optamos por realizar respaldos completos semanales tanto para dragon como para newton.

MENSUAL	0

De esta manera tendremos datos lo suficientemente consistentes para recuperarnos de algún problema, y además utilizaremos una cantidad mínima de cintas.

Sistemas de archivos de datos

En lo que respecta a los sistemas de archivos de datos de dragon se implementó un nuevo calendario de respaldos a cintas DLT, para ello hicimos uso del robot DLT-240 el cual tiene capacidad para albergar hasta 8 cartuchos DLT y un drive de lectura/escritura, el calendario de respaldos solo contempla los sistemas de archivos siguientes:

Sistema de archivos
/home/users01
/home/users02
/home/users03
/home/users04
/Sybase
/usr/Sybase
/home
/home/users00

El modelo de calendario de respaldos que se utilizó es el 3-4-5-6 a 2. Respaldos incrementales diferenciales diarios e incrementales acumulativos semanales.

MENSUAL	0

	Lunes	Martes	Miércoles	Jueves	Viernes
Semana 1	3	4	5	6	2
Semana 2	3	4	5	6	2
Semana 3	3	4	5	6	2
Semana 4	3	4	5	6	2

Tabla 2.6. Nuevo calendario de respaldos

Este calendario nos permite lo siguiente:

Se seguirán haciendo respaldos mensuales de nivel 0, pero para solucionar la pérdida de datos que pudiera darse en el lapso de tiempo entre un respaldo mensual y otro, propusimos respaldos incrementales de lunes a jueves e incrementales acumulativos semanales. De esta manera cada día de la semana el respaldo contiene solamente los archivos que se modificaron desde el día anterior, y el respaldo de cada viernes contiene todos los archivos que se actualizaron desde el nivel inicial 0 al principio del mes. Aunque parece que son muchas cintas las que se utilizan, en realidad no es así, recordemos que los respaldos incrementales al copiar solo la información que ha cambiado ocupan una menor cantidad de cintas que los respaldos completos.
Es importante mencionar que el anterior calendario de respaldos no aplica para el servidor newton ya que este último finalmente tendrá replicados los mismos datos que dragon.

Sistema de archivos de respaldos

Con el propósito de que la recuperación de información de sistema operativo fuese más rápida que de cinta, se creo un nuevo sistema de archivos de 17 GB llamado "respaldos" para copiar en él la misma información que se manda a cinta, este último fue creado en un disco local de los servidores, por lo tanto es más rápido respaldar y recuperar información a ese sistema de archivos que respaldarla o recuperarla de cinta a través de la red. Este sistema de archivos se creó tanto para dragón como para newton, en ambos servidores se llama igual y se utilizó el mismo procedimiento que es la creación de un volumen espejeado con volume manager. La creación de este sistema de archivos es parte del procedimiento de configuración de volume manager que se verá en el capítulo3, por lo tanto, solo hacemos referencia a él, más adelante se utilizará este sistema de archivos en los scripts de respaldos.

Se creó también este sistema de archivos para el servidor newton, el procedimiento fue el mismo que se menciona en el capítulo 3, lo único que cambia es el disco que en este caso fue el c2t85d0.

A este sistema de archivos que acabamos de crear se enviarán respaldos de los mismos sistemas de archivos que se mandan a cinta, solo que los respaldos se harán cada sábado a las 10 de la noche.

Automatización de respaldos

Respaldos a cinta

Para respaldar la información a cinta tanto de dragón como de newton, se modificaron los programas en shell de UNIX existentes, y ahora se ejecutará uno para cada día de la semana, ya que manejan diferentes niveles, se automatizó la ejecución de esos programas mediante una herramienta de sistema operativo que se llama cron. Un **cron** es un archivo que contiene la ruta y el nombre de un programa, el minuto, la hora, el dia, el mes y el año en que se ejecutará, y es frecuentemente usado para automatizar tareas que son repetitivas.

Haciendo caso al calendario de respaldos que implementamos, el respaldo mensual se ejecutará el día 1 de cada mes y para ello se utilizará un script o programa en de unix; los respaldos incrementales diferenciales se ejecutarán de lunes a jueves y se necesitan 4 scripts; todos los viernes se hará un respaldo incremental acumulativo, para ello se necesita un script; los scripts los guardaremos en el directorio /opt/shells.

Scripts del servidor dragon

Nombre de script: **/opt/shells/resp_completo_mensual_cinta.sh**

Contenido:

```
/usr/sbin/ufsdump  -0uf  /home      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /home/users00      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /home/users01      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /home/users02      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /home/users03      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /home/users04      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /home/users05      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /sybase            132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -0uf  /usr/sybase      132.248.18.9:/dev/rmt/0n
```

Nombre de script: **/opt/shells/resp_incremental_diferencial_lunes_cinta.sh**
Contenido:

```
/usr/sbin/ufsdump  -3uf  /home      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -3uf  /home/users00      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -3uf  /home/users01      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -3uf  /home/users02      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -3uf  /home/users03      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -3uf  /home/users04      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -3uf  /home/users05      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -3uf  /sybase            132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -3uf  /usr/sybase      132.248.18.9:/dev/rmt/0n
```

Nombre de script: **/opt/shells/resp_incremental_diferencial_martes_cinta.sh**
Contenido:

```
/usr/sbin/ufsdump  -4uf  /home     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -4uf  /home/users00     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -4uf  /home/users01     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -4uf  /home/users02     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -4uf  /home/users03     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -4uf  /home/users04     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -4uf  /home/users05     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -4uf  /sybase           132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -4uf  /usr/sybase     132.248.18.9:/dev/rmt/0n
```

Nombre de script: **/opt/shells/resp_incremental_diferencial_mierc_cinta.sh**
Contenido:

```
/usr/sbin/ufsdump  -5uf  /home     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -5uf  /home/users00     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -5uf  /home/users01     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -5uf  /home/users02     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -5uf  /home/users03     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -5uf  /home/users04     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -5uf  /home/users05     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -5uf  /sybase           132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -5uf  /usr/sybase     132.248.18.9:/dev/rmt/0n
```

Nombre de script: **/opt/shells/resp_incremental_diferencial_jueves_cinta.sh**
Contenido:

```
/usr/sbin/ufsdump  -6uf  /home     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -6uf  /home/users00     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -6uf  /home/users01     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -6uf  /home/users02     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -6uf  /home/users03     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -6uf  /home/users04     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -6uf  /home/users05     132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -6uf  /Sybase         132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -6uf  /usr/sybase     132.248.18.9:/dev/rmt/0n
```

Nombre de script: **/opt/shells/resp_incremental_acumalativo_viernes_cinta.sh**
Contenido:

```
/usr/sbin/ufsdump  -2uf  /home      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -2uf  /home/users00      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -2uf  /home/users01      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -2uf  /home/users02      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -2uf  /home/users03      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -2uf  /home/users04      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -2uf  /home/users05      132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -2uf  /sybase            132.248.18.9:/dev/rmt/0n
/usr/sbin/ufsdump  -2uf  /usr/sybase     132.248.18.9:/dev/rmt/0n
```

Para hacer automática la ejecución de estos scripts, agregamos el nombre y la ruta del directorio en el que están en el archivo cron del usuario root. Un archivo de cron se utiliza para hacer tareas de forma repetida, dentro de su contenido podemos encontrar 6 columnas, la primer columna se refiere al minuto en que se ejecutará un programa o comando y sus valores van desde 0 a 59; la segunda columna indica la hora en que se ejecutará un programa y sus valores van de 1 a 23; la tercer columna hace referencia al dia del mes en que se ejecutará el programa, sus valores pueden ser de 0 a 31; la cuarta al mes del año y sus valores van de 1 a 12; la quinta se refiere al día de la semana en que se ejecutará nuestro programa, sus valores van de 0 a 6, donde 0 es domingo, 1 el lunes, 2 el martes, tres el miércoles, 4 el jueves, 5 el viernes y 6 el sábado; y finalmente la quinta columna contiene la ruta y nombre del programa a ejecutarse. Para hacer referencia a todos los minutos, todas las horas, todos los dias del mes, todos los años.

En seguida se muestran las lineas que debe contener el cron del usuario root para automatizar la ejecución de respaldos.

```
#ident  "@(#)root       1.19    98/07/06 SMI"  /* SVr4.0 1.1.3.1      */
#
# The root crontab should be used to perform accounting data collection.
#
# The rtc command is run to adjust the real time clock if and when
# daylight savings time changes.
#

00 21 1 * *    /opt/shells/resp_completo_mensual_cinta.sh                    ## 1
00 22 * * 1    /opt/shells/resp_incremental_diferencial_lunes_cinta.sh       ## 2
00 22 * * 2    /opt/shells/resp_incremental_diferencial_martes_cinta.sh      ## 2
00 22 * * 3    /opt/shells/resp_incremental_diferencial_mierc_cinta.sh       ## 3
00 22 * * 4    /opt/shells/resp_incremental_diferencial_jueves_cinta.sh      ## 4
00 22 * * 5 /opt/shells/resp_incremental_acumalativo_viernes_cinta.sh        ## 5
```

Las líneas del cron anterior tienen el siguiente significado:

Línea 1:

00 Indica el minuto en el que se ejecutará el script, en este caso es minuto 0.
21 Indica la hora en que se ejecutará el script, en este caso a las 21 hrs.
1 Indica el día del mes, en este caso el 1 indica que el script se ejecutará el primer dia del mes.
* Indica el mes, en este caso el * indica para todos los meses del año.
* Indica el día de la semana, en este caso el * indica todos los día de la semana

De esta forma traduciendo toda la línea podemos decir que a las 21 hrs. con 0 minutos, del primer día de todos los meses del año cualquiera que sea el día de la semana, se ejecutará el script que hace un respaldo completo mensual.

Línea 2:

00 Indica el minuto en el que se ejecutará el script, en este caso es minuto 0.
22 Indica la hora en que se ejecutará el script, en este caso a las 22 hrs.
* Indica el día del mes, en este caso el * indica que el script aplicará para cualquier día del mes.
* Indica el mes, en este caso el * indica que el script aplicará para todos los meses del año.
1 Indica el día de la semana, en este caso el 1 que el script se ejecutará el día 1 de la semana que corresponde al lunes.

De esta forma traduciendo toda la línea podemos decir que a las 22 hrs. con 0 minutos, del día lunes de cada semana, durante todos los meses del año, se ejecutará el script que hace un respaldo incremental diferencial de nivel 3.

Línea 3:

00 Indica el minuto en el que se ejecutará el script, en este caso es minuto 0.
22 Indica la hora en que se ejecutará el script, en este caso a las 22 hrs.
* Indica el día del mes, en este caso el * indica que el script aplicará para cualquier día del mes.
* Indica el mes, en este caso el * indica que el script aplicará para todos los meses del año.
2 Indica el día de la semana, en este caso el 1 que el script se ejecutará el día 2 de la semana que corresponde al martes.

De esta forma traduciendo toda la línea podemos decir que a las 22 hrs. con 0 minutos, del día martes de cada semana, durante todos los meses del año, se ejecutará el script que hace un respaldo incremental diferencial de nivel 4.

Línea 4:

00 Indica el minuto en el que se ejecutará el script, en este caso es minuto 0.
22 Indica la hora en que se ejecutará el script, en este caso a las 22 hrs.
* Indica el día del mes, en este caso el * indica que el script aplicará para cualquier día del mes.
* Indica el mes, en este caso el * indica que el script aplicará para todos los meses del año.
3 Indica el día de la semana, en este caso el 1 que el script se ejecutará el día 2 de la semana que corresponde al miércoles.

De esta forma traduciendo toda la línea podemos decir que a las 22 hrs. con 0 minutos, del día miércoles de cada semana, durante todos los meses del año, se ejecutará el script que hace un respaldo incremental diferencial de nivel 5.

Línea 5:

00 Indica el minuto en el que se ejecutará el script, en este caso es minuto 0.
22 Indica la hora en que se ejecutará el script, en este caso a las 22 hrs.
* Indica el día del mes, en este caso el * indica que el script aplicará para cualquier día del mes.
* Indica el mes, en este caso el * indica que el script aplicará para todos los meses del año.
4 Indica el día de la semana, en este caso el 1 que el script se ejecutará el día 2 de la semana que corresponde al jueves .

De esta forma traduciendo toda la línea podemos decir que a las 22 hrs. con 0 minutos, del día jueves de cada semana, durante todos los meses del año, se ejecutará el script que hace un respaldo incremental diferencial de nivel 6.

Línea 6:

00 Indica el minuto en el que se ejecutará el script, en este caso es minuto 0.
22 Indica la hora en que se ejecutará el script, en este caso a las 22 hrs.
* Indica el día del mes, en este caso el * indica que el script aplicará para cualquier día del mes.
* Indica el mes, en este caso el * indica que el script aplicará para todos los meses del año.
5 Indica el día de la semana, en este caso el 1 que el script se ejecutará el día 5 de la semana que corresponde al viernes.

De esta forma traduciendo toda la línea podemos decir que a las 22 hrs. con 0 minutos, del día viernes de cada semana, durante todos los meses del año, se ejecutará el script que hace un respaldo incremental diferencial de nivel 2.

Respaldos a disco

Para los respaldos a disco el script que utilizamos varía un poco en relación con los scripts que respaldan a cinta y se muestra a continuación.

Nombre de script: **/opt/shells/resp_completo_semanal_disco.sh**

Contenido:

```
#!/bin/ksh

## Este script realiza un respaldo completo de Sistema Operativo en la ruta ## /respaldos/actual
y antes  de realizarlo valida si existe el espacio
## suficiente en disco. Tambien el script guarda una version comprimida del ## respaldo anterior
en la ruta  /respaldos/anterior.




usr_used=`df -k /usr | grep usr | awk '{print $3}'`
root_used=`df -k / | grep / | awk '{print $3}'`
var_used=`df -k /var | grep var | awk '{print $3}'`
opt_used=`df -k /opt | grep opt | awk '{print $3}'`
OS_actual=`expr $usr_used + $root_used + $var_used + $opt_used`
#### echo " el SO ocupa $OS_actual"

/usr/bin/gzip /respaldos/actual/*.dump

current_used=`du -k /respaldos/actual | awk '{print $1}'`
#### echo " el respaldo comprimido ocupa $current_used"
fs_espace=`df -k /respaldos | grep ufsdump | awk '{print $2}'`
#### echo " el espacio en el sistema de archivos es $fs_espace "
OS_plus_dump=`expr $OS_actual + $current_used`
#### echo " el SO mas el dump ocupa $OS_plus_dump"

if [ $OS_plus_dump -gt $fs_espace ]
   then

    echo " $DATE se descomprime el respaldo anterior  no se realiza el nuevo por falta de espacio
" >> /respaldos/logs/respaldo_completo_semanal_disco.log

    /usr/sbin/gunzip /respaldos/actual/*.dump.gz

    elif [ $fs_espace -gt $OS_plus_dump ]

   then
    echo "se  mueve el respaldo a anterior y se realizara el dump " >>
/respaldos/logs/respaldo_completo_semanal_disco.log
    /usr/bin/rm /respaldos/anterior/*.dump.gz
    /usr/bin/mv /respaldos/actual/*.dump.gz /respaldos/anterior/
```

```
    ufsdump 0uf /respaldos/actual/root.dump /
    ufsdump 0uf /respaldos/actual/usr.dump /usr
    ufsdump 0uf /respaldos/actual/var.dump /var
    ufsdump 0uf /respaldos/actual/opt.dump /opt


fi
```

La línea de entrada que le corresponde en el cron del usuario root es:

00 22 * * 6 /opt/shells/resp_completo_semanal_disco.sh

De esta forma traduciendo toda la línea podemos decir que a las 22 hrs. con 0 minutos, del día 6 de la semana que corresponde al sábado, durante todas las semanas, durante todos los meses del año, se ejecutará el script que hace un respaldo completo a disco.

Es importante mencionar que en el caso de los respaldos a cinta, los administradores deben de cerciorarse de que las cintas no estén llenas.

Estos scripts son exactamente los mismos para el servidor newton, lo único que cambia es que también se incluye el sistema de archivos /home, y la dirección de la máquina Sun Enterprise-3500 es la 132.248.18.21 en lugar de la 132.248.18.9.

Disco Alterno de Sistema Operativo

La tercer y última mejora que hicimos al esquema de respaldos es la creación de un disco alterno de sistema operativo. El disco donde se instaló originalmente el sistema operativo es el c1t0d0 y es de 18 GB de capacidad, este disco está divido en cinco sistemas de archivos: /, swap, /opt, /usr y /var, cada uno de los cuales tiene un tamaño específico. Analizando un poco los discos que tiene el servidor web nos dimos cuenta de que habían discos que no se estaban utilizando, de entre ellos elegimos el c1t16d0 de 18 GB el cual al ser idéntico al que está usando hoy día el sistema operativo decidimos usarlo como disco alterno.

El beneficio principal que se puede obtener con esto es que si el disco original de sistema operativo falla físicamente y el servidor web deja de funcionar, podemos utilizar este disco para reanudar el funcionamiento de todo el servidor. Para lograr esto, primero definimos las particiones del disco alterno iguales a las del disco actual de sistema operativo, lo hicimos inicializable y mediante comandos mandamos un respaldo, posteriormente se hizo un script para automatizar ese proceso.

Sabemos que el disco actual de sistema operativo es el c1t0d0, y el disco que queremos que funcione como alterno es el c1t16d0. Sabemos que el disco actual tiene definidas las particiones /, swap, /opt, /usr y /var de un tamaño de terminado, entonces el disco alterno debe de tener las mismas particiones y del mismo tamaño. La partición de disco correspondiente a cada sistema de archivos del disco actual se muestra a continuación.

```
/       c1t0d0s0
swap    c1t0d0s1
/usr    c1t0d0s3
/var    c1t0d0s4
/opt    c1t0d0s5
```

Creación de disco alterno de sistema operativo

El siguiente procedimiento nos describe la creación de un disco alterno.

1.- Copiar la tabla de particiones del disco actual al disco alterno.

```
# prtvtoc  /dev/rdsk/c1t0d0s0 | fmthard -s - /dev/rdsk/c1t16d0s0
```

Con esto en el disco actual de sistema operativo y en el disco alterno tenemos definidas las mismas particiones del mismo tamaño.

2.- Crear un nuevo sistema de archivos sobre la partición 0 del disco alterno idéntica a la del disco actual y que corresponde al sistema de archivos /var.

```
# newfs -v /dev/rdsk/c1t16d0s0
```

3.- Verificar la integridad del sistema de archivos creado.

```
# fsck -o f /dev/rdsk/c1t16d0s0
```

4.- Crear el punto de montura de este sistema de archivos, le asignaremos un nombre distinto al que originalmente tiene el del disco actual de sistema operativo para evitar confusiones.

```
# mkdir /roottmp
```

5.- Crear un nuevo sistema de archivos sobre la partición 1 del disco alterno idéntico al sistema de archivos creado en la partición 1 del disco actual y que corresponde al sistema de archivos swap.

```
# newfs -v /dev/rdsk/c1t16d0s1
```

6.- Verificar la integridad del nuevo sistema de archivos creado.

```
# fsck -o f /dev/rdsk/c1t16d0s0
```

7.- Crear el punto de montura de este sistema de archivos, le asignaremos un nombre distinto al que originalmente tiene el del disco actual de sistema operativo para evitar confusiones.

```
# mkdir /tmptmp
```

8.- Crear un nuevo sistema de archivos sobre la partición 3 del disco alterno idéntico al sistema de archivos creado en la partición 3 del disco actual y que corresponde al sistema de archivos /usr.

```
# newfs -v /dev/rdsk/c1t16d0s3
```

9.- Verificar la integridad del nuevo sistema de archivos creado.

```
# fsck -o f /dev/rdsk/c1t16d0s3
```

10.- Crear el punto de montura de este sistema de archivos, le asignaremos un nombre distinto al que originalmente tiene el del disco actual de sistema operativo para evitar confusiones.

```
# mkdir /usrtmp
```

11.- Crear un nuevo sistema de archivos sobre la partición 4 del disco alterno idéntico al sistema de archivos creado en la partición 4 del disco actual y que corresponde al sistema de archivos /var

```
# newfs -v /dev/rdsk/c1t16d0s4
```

12.- Verificar la integridad del nuevo sistema de archivos creado.

```
# fsck -o f /dev/rdsk/c1t16d0s4
```

13.- Crear el punto de montura de este sistema de archivos, le asignaremos un nombre distinto al que originalmente tiene el del disco actual de sistema operativo para evitar confusiones.

```
# mkdir /vartmp
```

14.- Crear un nuevo sistema de archivos sobre la partición 5 del disco alterno idéntico al sistema de archivos creado sobre la partición 5 del disco actual y que corresponde al sistema de archivos /opt.

```
# newfs -v /dev/rdsk/c1t16d0s5
```

15.- Verificar la integridad del nuevo sistema de archivos creado.

```
# fsck -o f /dev/rdsk/c1t16d0s5
```

16.- Crear el punto de montura de este sistema de archivos, le asignaremos un nombre distinto al que originalmente tiene el del disco actual de sistema operativo para evitar confusiones.

```
# mkdir /opttmp
```

Una vez que tenemos creados los puntos de montura, continuamos con la copia del disco original al disco alterno. Esto lo hacemos usando el comando ufsdump y copiando un sistema de archivos a la vez.

17.- En la copia del disco actual de sistema operativo al disco alterno las operaciones que se hacen son básicamente: checar la consistencia del sistema de archivos, montarlo en su punto de montura temporal correspondiente, hacer una copia de la información del sistema de archivos del disco actual al sistema de archivos del disco alterno correspondiente, y por último se desmontar el sistema de archivos del disco alterno. Estos pasos son los mismos para /, /usr, /var y /opt, con la diferencia que para el caso de / se hace una copia del archivo /etc/vfstab del disco original a /roottmp/etc del disco alterno, este archivo es leído cuando el servidor inicia y los sistemas de archivos que allí tiene definidos son los que monta de manera automática, de tal forma que si hay un daño físico en el disco actual podemos sustituir el disco dañado por el disco alterno sin ningún problema, además también para /, se ejecuta el comando bootblk que sirve para que el disco alterno pueda usarse al iniciarse el servidor o lo que es lo mismo para que sea reinicializable.

```
### Para /

fsck -Y /dev/rdsk/c1t16d0s0
mount /dev/dsk/c1t16d0s0 /roottmp
cd /
ufsdump 0f - /dev/rdsk/c1t0d0s0 | (cd /roottmp; ufsrestore rf -)
cp -p /etc/vfstab /roottmp/etc
cd /
umount /roottmp
fsck -Y /dev/rdsk/c1t16d0s0

installboot /usr/platform/SUN4u/lib/fs/ufs/bootblk /dev/rdsk/c1t16d0s0



### Para /usr

fsck -Y /dev/rdsk/c1t16d0s3
mount /dev/dsk/c1t16d0s3 /usrtmp
cd /
ufsdump 0f - /dev/rdsk/c1t0d0s3 | (cd /usrtmp; ufsrestore rf -)
cd /
umount /usrtmp
fsck -Y /dev/rdsk/c1t16d0s3

### Para /var

fsck -Y /dev/rdsk/c1t16d0s4
mount /dev/dsk/c1t16d0s4 /vartmp
cd /
ufsdump 0f - /dev/rdsk/c1t0d0s4 | (cd /vartmp; ufsrestore rf -)
cd /
umount /usrtmp
fsck -Y /dev/rdsk/c1t16d0s4
```

```
### Para /opt

fsck -Y /dev/rdsk/c1t16d0s5
mount /dev/dsk/c2t0d0s4 /opttmp
cd /
ufsdump 0f - /dev/rdsk/c1t0d0s5 | (cd /opttmp; ufsrestore rf -)
cd /
umount /opttmp
fsck -Y /dev/rdsk/c1t16d0s5
```

Automatización de creación de disco alterno de sistema operativo

Los pasos anteriores indican como crear un disco alterno de manera manual, esto se hizo solo la primera vez, pero para automatizar este proceso desarrollamos el siguiente script, debemos hacer notar que el comando para hacer el disco alterno reinicializable solo se ejecutó una vez que fue cuando hicimos el proceso de forma manual, posteriormente ya no es necesario volver a ejecutar el comando bootblk.

```
#!/bin/ksh
##############################################################################
# Este script hace una copia del Sistema Operativo del disco primario a un #disco alterno. En
este caso el disco primario es el c1t0d0 y el disco #alterno es el c1t7d0
#
##############################################################################

echo "`date '+%a %b %e %T'` `hostname` Verificando que existan los  puntos de Montura ..."


for i in roottmp usrtmp vartmp opttmp
do
cd /
test ! -d $i && mkdir $i
done


fsck -Y /dev/rdsk/c1t0d0s0
mount /dev/dsk/c1t16d0s0 /roottmp
cp -p /roottmp/etc/vfstab /respaldos/CLONSO
cd /roottmp
rm -rf *
cd /
ufsdump 0f - /dev/rdsk/c1t0d0s0 | (cd /roottmp; ufsrestore rf -)
cp -p /respaldos/CLONSO/vfstab /roottmp/etc
umount /roottmp
fsck -Y /dev/rdsk/c1t16d0s0
echo "`date '+%a %b %e %T'` `hostname` El file system / ha sido respaldado"

fsck -Y /dev/rdsk/c1t16d0s3
mount /dev/dsk/c1t16d0s3 /usrtmp
cd /usrtmp
rm -rf *
cd /
ufsdump 0f - /dev/rdsk/c1t0d0s3 | (cd /usrtmp; ufsrestore rf -)
umount /usrtmp
```

```
fsck -Y /dev/rdsk/c1t16d0s3
echo "`date '+%a %b %e %T'` `hostname` El file system /usr ha sido respaldado"

fsck -Y /dev/rdsk/c1t16d0s4
mount /dev/dsk/c1t16d0s4 /vartmp
cd /vartmp
rm -rf *
cd /
ufsdump 0f - /dev/rdsk/c1t0d0s4 | (cd /vartmp; ufsrestore rf -)
umount /vartmp
fsck -Y /dev/rdsk/c1t16d0s4
echo "`date '+%a %b %e %T'` `hostname` El file system /var ha sido respaldado"

fsck -Y /dev/rdsk/c1t16d0s5
mount /dev/dsk/c1t16d0s5 /opttmp
cd /opttmp
rm -rf *
cd /
ufsdump 0f - /dev/rdsk/c1t0d0s5 | (cd /opttmp; ufsrestore rf -)
umount /opttmp
fsck -Y /dev/rdsk/c0t5d0s5
echo "`date '+%a %b %e %T'` `hostname` El file system /opt ha sido respaldado"
```

Otra vez recalcamos que la creación del disco alterno se hizo tanto para el servidor web (dragon), como para el servidor newton, solo que en el caso de este último el disco actual de sistema operativo después de la reestructuración que se menciona en el capítulo 4 es el c0t0d0 y el disco alterno el c1t2d0 de .

También debemos recordar que la implementación tanto del nuevo esquema de respaldos a cinta, a disco y la creación del disco alterno se llevó a cabo después de que se hizo la reestructuración de discos del capítulo tres.

Capítulo 3. Alta Disponibilidad en el Manejo de Datos

Los discos y los datos que se encuentran en ellos, son la parte más crítica de cualquier sistema de cómputo. En este capitulo discutiremos las tecnologías existentes para el manejo de discos y algunas de las opciones que tenemos para proteger los datos que se encuentran en esos discos contra posibles fallas. Hablaremos de manera general de la forma en que trabajan algunos paquetes de software manejadores de discos como lo son VERITAS Volume Manager y Solstice DiskSuite, haremos un análisis de la configuración actual de discos del servidor WEB de la UNAM y posteriormente una reestructuración física y lógica de esa configuración haciendo uso de los manejadores de discos mencionados.

Tecnologías de Conectividad en Discos

Dispositivo SCSI

Un dispositivo SCSI es un circuito eléctrico que funciona como canal de transferencia de datos entre dos dispositivos, valga la redundancia. Existen diferentes variedades de SCSI, entre ellas encontramos narrow, wide y ultra, cada una de estas tecnologías posee diferentes capacidades al transportar los datos a través del bus o canal de transferencia; narrow soporta 8 bits, wide 16 bits y ultra soporta 32 bits; teóricamente la velocidad máxima de transferencia de narrow es de 10 megabytes por segundo, para wide es de 20 megabytes y para ultra es de 40 megabytes por segundo; en realidad las velocidades tope trabajan a un 10 o 15 por ciento menos de la velocidad teórica, debido a la carga que normalmente tiene el canal provocada por otros dispositivos. Así mismo de éstas tecnologías se desprenden diversas variedades para cada uno de ellos: SCSI-1, SCSI-2, Single-ended SCSI-2, Diferential SCSI-2, Fast/Narrow SCS!-2, Fast/Wide SCSI-2, SCSI-3, Wide Ultra-SCSI, cada una de las cuales tiene características diferentes, sin embargo no es objetivo de este trabajo analizar cada una de ellas, solo basta decir que nos sirven para conectar dispositivos a nuestros servidores, entre ellos los más comunes son los discos duros, aunque también nos permiten conectar y hacer uso de unidades de cinta, robots de manejo de cintas, y unidades de CD entre otros.

Fibrechannel

Fibrechannel o canal de fibra es la nueva tecnología para conectar discos a los servidores, soporta velocidades de hasta 100 MB/seg., normalmente estos arreglos de discos están disponibles con dos canales lo cual redunda en una velocidad de transferencia de 200 MB/seg. Esta tecnología soporta hasta 126 dispositivos de fibra.

Esta tecnología soporta mayores distancias en la conectividad de los discos que la tecnología SCSI, la distancia normalmente soportada es de 2 kilómetros, la cual puede ser incrementada si se utilizan repetidores de fibra. Además soporta todo el hardware que normalmente se asocia a las redes de área local como pueden ser ruteadores, concentradores. Fibrechannel ofrece completa independencia eléctrica de otros dispositivos.

Multihosting

Tenemos una conexión multiservidor cuando un arreglo de discos se conecta a dos o más servidores al mismo tiempo. Los servidores pueden acceder los mismos discos si fuera necesario, pero no al mismo tiempo, este tipo de arreglos son comúnmente utilizados en ambientes de cluster.

Multipathing

Es una conexión de un servidor con un arreglo de discos pero con más de un canal de transferencia de datos, esta tecnología requiere de múltiples tarjetas en el servidor y en el arreglo de discos, la transferencia puede ser incrementada linealmente si ambos canales están trabajando al mismo tiempo, si una conexión falla la otra conexión toma el control total de los canales de datos. Las fallas a nivel de fibra o tarjetas de datos son transparentes para las aplicaciones, esto quiere decir que no se verá interrumpido el funcionamiento de nuestro sistema.

Arreglos de discos

Un arreglo de discos es un gabinete o caja que contiene ranuras o huecos para albergar varios discos. Este tipo de dispositivos puede manejar tecnologías SCSI o canales de fibra internamente para hacer el direccionamiento de los datos o para conectarse al servidor. Así mismo dentro de los arreglos de discos podemos encontrar distintas tecnologías para los discos, las más comunes son: discos de conexión en caliente, discos de conexión en frío y discos de reserva.

Hot-Pluggable

Este tipo de discos puede añadirse o removerse del arreglo de discos sin tener que dar de baja el servidor; el hacer este tipo de movimientos no tiene impacto para otros discos conectados al servidor. Cuando se quieren implementar sistemas que sean altamente disponibles estos son los discos por los cuales debemos optar.

Warm-Pluggable

Esta es una versión simple de los discos de conexión en caliente, ya que para añadir o remover un disco se tiene que dar de baja el servidor en la mayoría de los casos, a su vez este tipo de movimientos afecta a otros discos dentro de la máquina.

Hot Spares

Los discos de reserva son aquellos que están configurados de tal forma que están en espera de que un disco falle para entrar al relevo. Cuando esto ocurre toda la información es movida del disco dañado al disco de reserva.

Existe una gran variedad de arreglos de discos en la industria, arreglos D1000, A5200, T3, NetAps, EMC2, Hitachi por citar solo algunos, a continuación explicaremos de forma breve como están constituidos los arreglos A5000, A5200 y D1000, ya que son los discos que estaremos utilizando en este trabajo.

Arreglos Sun A5000

Un arreglo de discos A5000 es una caja que tiene capacidad para albergar hasta 14 discos duros. Tiene entradas para discos tanto en la parte frontal como en la parte trasera, y en cada una de ellas se pueden tener siete discos. Esta caja puede conectarse a un servidor mediante uno o más canales de fibra óptica y cuando se configura debe de asignársele un nombre mismo con el cual será identificada por el sistema operativo del servidor al que esté conectada.

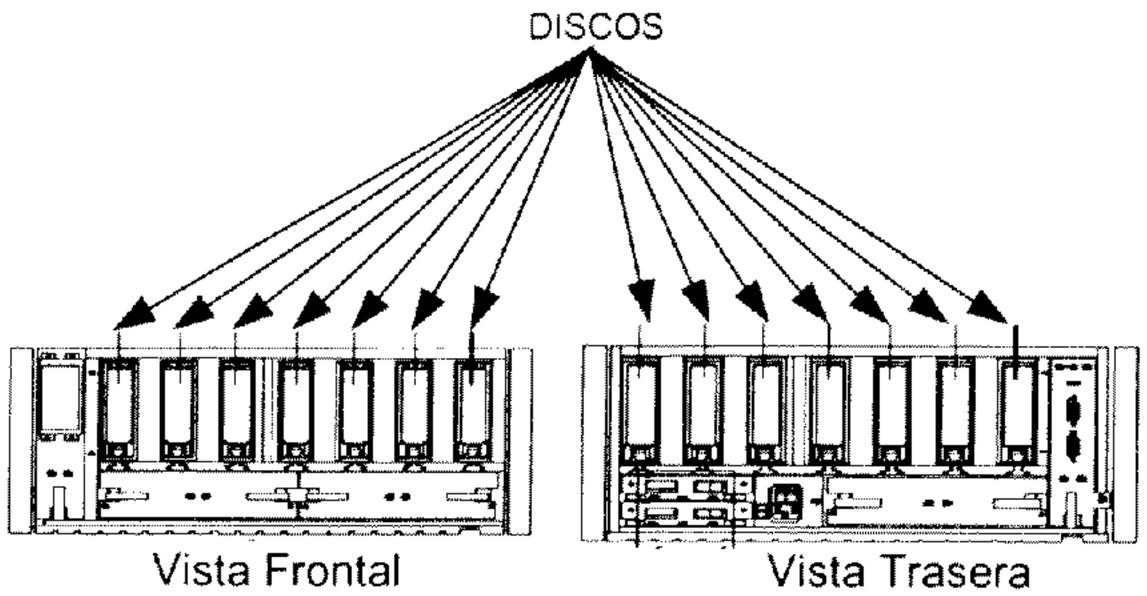


Figura 3.1. Arreglo de discos Sun A5000

Arreglos Sun A5200

Este arreglo tiene las mismas características que el A5000, solo que este tiene capacidad para albergar hasta 22 discos.

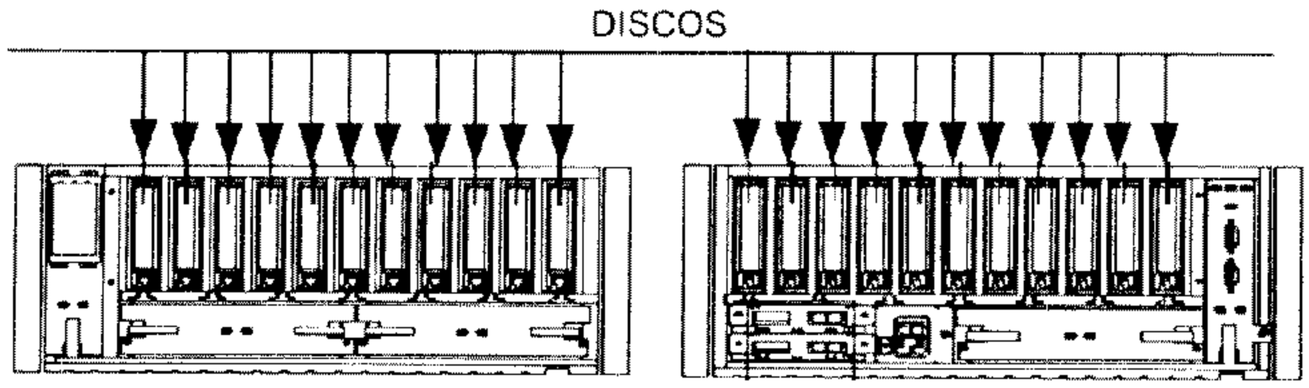


Figura 3.2. Arreglo de discos Sun A5200

Arreglos Sun D1000

Al igual que los arreglos anteriores, un arreglo de discos D1000 es una caja que puede tener varios discos en su interior. Un arreglo de este tipo tiene las entradas para los discos en la parte frontal de la caja, la cual puede ser conectada hacia un servidor por medio de canales SCSI. Existen cajas de 8 y 12 discos, la figura siguiente muestra la parte frontal un arreglo de discos D1000 de 12 discos.

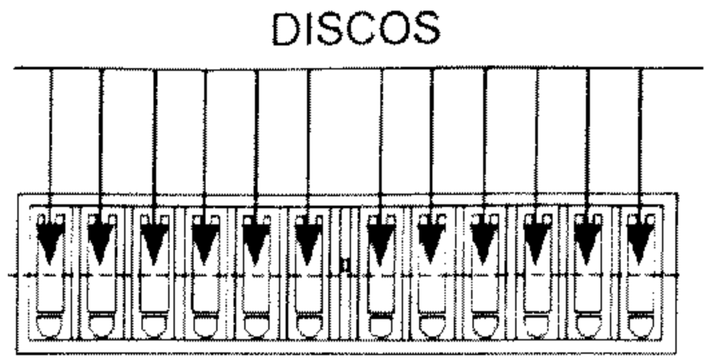


Figura 3.3. Arreglo de discos Sun D1000

Estructura Física de un Disco

El dispositivo físico de almacenamiento de datos más básico es el disco duro. Antes de utilizar un disco es necesario formatearlo, formatear un disco duro significa prepararlo y organizarlo para que la información pueda ser escrita o leída.

En el sistema operativo Solaris un disco está compuesto de las siguientes partes:

VTOC. Un disco de Solaris tiene un área llamada "Volume Table of Contents" (VTOC), o Contenido de Tabla de Volumen, la cual almacena información acerca de la estructura y organización del disco. El VTOC es también conocido como etiqueta de disco.

Particiones. Enseguida del VTOC, el disco es dividido en unidades llamadas particiones o slices. Una partición es simplemente un grupo de cilindros que tienen un uso en particular. La información referente al tamaño, localización, y uso de las particiones es almacenada en el VTOC. Un disco puede tener como máximo 8 particiones, la partición 2 que aparece en la gráfica no se toma en cuenta para tal efecto ya que hace referencia a todo el disco incluyendo el VTOC, por lo tanto en realidad sólo podemos crear hasta siete particiones.

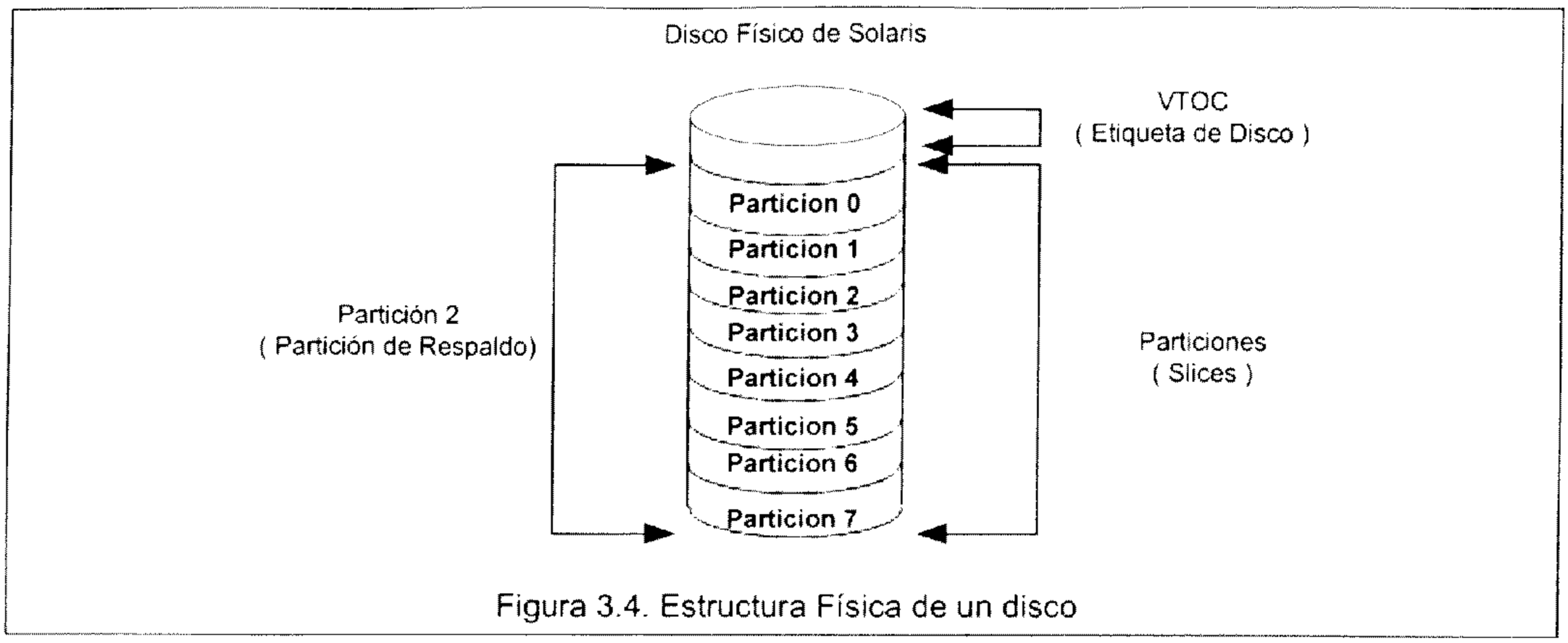


Figura 3.4. Estructura Física de un disco

Nombre Físico de Disco

Cuando guardamos y accedemos datos en un disco físico lo hacemos utilizando un nombre de dispositivo que el sistema operativo reconoce, el cual, especifica la controladora, el target ID, y el número de disco. Un nombre de dispositivo típico utiliza el formato: c#t#d#

c# Es el número de controladora de disco, la cual como su nombre lo dice se encarga de controlar el disco.
t# Es el target ID.
d# Es el número de disco.

Si el disco está dividido en particiones, entonces el nombre de dispositivo también debe incluir el número de partición.

s# Es el número de partición o slice.

Por ejemplo, el nombre de dispositivo c0t0d0s1 está conectado a la controladora número 0 del sistema, con un target ID de 0, un número de disco físico de 0 y la partición del disco a la que hace referencia es la 1.

Tecnología RAID

Los niveles de RAID (Redundant Array of Independent Disk) por sus siglas en inglés, se han convertido en estándares de la industria, nos ayudan a describir diferentes caminos para combinar y manejar un conjunto de discos independientes, los cuales, de acuerdo al nivel de RAID elegido nos proporcionará un buen funcionamiento de lectura-escritura, disponibilidad o ambas. Existen dos vertientes para implementar soluciones tipo RAID en nuestros servidores y estos son RAID por software y RAID por hardware, siendo este último el que proporciona un mejor funcionamiento en nuestros equipos, sin embargo este tipo de solución es mas costosa, en el RAID por software encontramos herramientas como VERITAS Volume Manager y Solstice DiskSuite, las cuales nos ayudan a realizar casi todos los niveles de RAID disponibles, esto es, hasta cierto punto una solución más económica pero a su vez representa carga para nuestro servidor.

En este trabajo nos enfocaremos a los niveles RAID por software, antes de ver los niveles RAID veremos de forma breve las distribuciones de discos que éstos utilizan, estos mismos tipos de distribuciones se verán mas adelante de forma más detallada cuando veamos el funcionamiento de VERITAS Volume Manager y Solstice DiskSuite.

Por lo pronto, mencionaremos de una forma más general los tipos de distribución de discos y los niveles RAID que existen.

Tipos de RAID

Concat

A partir de uno o más discos físicos se genera un disco virtual el cual va ser leído y escrito de forma secuencial.

Striping

Los discos físicos son acomodados en columnas virtuales en los discos y la escritura y lectura se hace a lo largo de esas columnas de forma paralela, incrementando lo que se conoce como performance en el sistema de archivos.

Mirror

En un espejo se generan uno o más elementos virtuales, a su vez cada uno de estos elementos virtuales está formado por uno o más discos físicos. Cada uno de los elementos

virtuales tiene una copia de los mismos datos, de allí que se llamen espejos. Los espejos a su vez pueden tener distribución de concatenado o de striping.

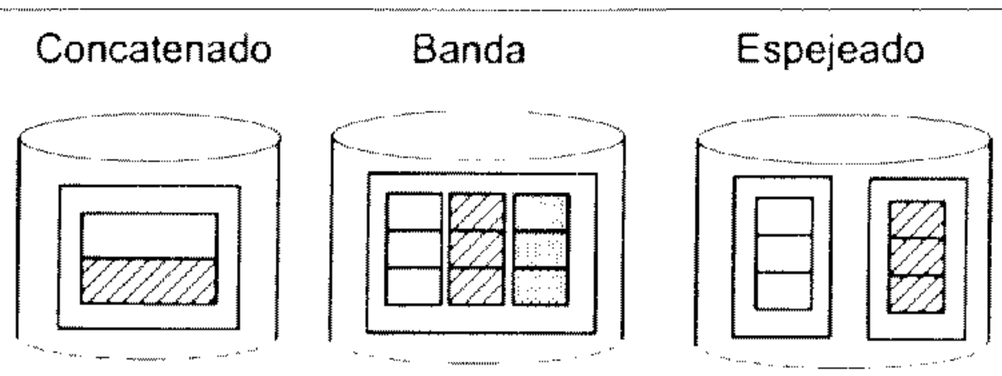


Figura 3.5. Tipos de distribución de discos

Niveles RAID

En los estándares declarados en la tecnología RAID, se describen diferentes formas para combinar y manejar un grupo de discos independientes, los cuales pueden proporcionar cierto nivel de redundancia a nuestro sistema a nivel de disco. Actualmente se utilizan en el mercado los niveles de RAID 0, 1, 3 y 5 de los cuales, el único que no proporciona protección a nuestros datos es el nivel 0. La funcionalidad de RAID puede implementarse a nivel de arreglo de disco o a nivel de controladora, a éste último se le conoce como RAID por hardware; al RAID implementado utilizando únicamente los discos de un arreglo se le conoce como RAID por software, ya que es necesario una aplicación como DiskSuite o Volume Manager para manejar los datos, este requiere trabajo del host y por ende ciclos de CPU de nuestro equipo. Así mismo, se utilizan en el campo laboral una combinación de dos niveles de RAID, a estos niveles se les conoce como RAID 0+1 y RAID 1+0.

RAID 0: striping o concat

En el RAID 0 se definen los stripes y los concats o concatenados. Para crear un RAID 0 el disco debe ser segmentado en espacios llamados chunks y es necesario tener un mínimo de 2 chunks para poder crear una estructura tipo stripe y solo un chunk para crear una estructura tipo concat, dependiendo de cómo agrupemos los chunks el sistema realiza las lecturas y escrituras al disco.

En el RAID 0 tipo concat es necesario contar como mínimo con dos discos y las lecturas y escrituras en el disco se realizaran en forma secuencial, esto es, hasta que se termine el espacio en el primer chunk se escribe en el segundo, por ende el RAID 0 tipo concat no aprovecha el que contemos con dos cabezas de lectura y por lo tanto no realiza balanceo de

cargas en los discos. Este tipo de RAID normalmente no es utilizado en configuraciones que requieran constantes lecturas y escrituras a disco.

En el RAID 0 tipo stripe al igual que concat, es necesario contar con dos discos y las lecturas y escrituras se realizan en forma paralela, esto quiere decir, que ambas cabezas de los discos trabajan al mismo tiempo leyendo y escribiendo a disco, lo cual incrementa el performance en nuestro equipo ya que se realiza un balanceo de carga.

Aunque los stripes nos ayudan a incrementar el performance en nuestros equipos a lo largo de este proyecto no lo recomendaremos para ser implementado por sí solo, ya que no proporciona redundancia en los datos. Su principal desventaja al igual que los concats radica en que si llega a fallar un disco se pierde prácticamente todo el filesystem y no es posible recuperarlo a menos que se cuente con un respaldo confiable de los datos, a medida que se utilicen más discos se incrementa la posibilidad de que uno falle.

RAID 1

El RAID 1 también es conocido con el nombre de mirror, en un mirror existen dos grupos de discos y ambos albergan la misma información, en este modelo si un disco falla nuestros datos siguen operando sin ninguna interrupción, las escrituras y las lecturas no se ven afectadas.

De hecho los mirrors no están limitados a dos copias, muchos administradores de sistemas utilizan mirrors con 3 o 4 caras para sus ambientes de producción, son tan versátiles los mirrors que en un momento determinado se puede despegar una cara del espejo con el fin de realizar un respaldo confiable de los datos.

Una concepción equivocada del mirror, es que al contar con un mirror no necesitamos hacer respaldos, esta es un idea errónea ya que los mirrors fueron diseñados para protegernos de fallas en hardware pero no nos protegen de nuestros usuarios (borrado accidental de archivos), o de la corrupción de datos, si un archivo es borrado se borra de ambas caras del mirror.

La principal ventaja de los mirrors radica en protegernos de la inevitable pérdida de los datos cuando falla un disco y la segunda radica en el aumento de performance en las lecturas. Pero también los mirrors tienen desventajas y la más notable es su alto costo ya que de un total de espacio solo podemos utilizar el 50% para almacenar datos.

Combinando RAID 0 y RAID 1.

Existen combinaciones de dos niveles de RAID, las cuales, son utilizadas con frecuencia en la industria, estos son el RAID 0+1 y RAID 1+0, ambos son métodos superiores a RAID 1, ya que tiene las ventajas del RAID 0 y RAID 1, así mismo, el implementar un nivel de RAID combinado incrementa la complejidad en la administración de discos.

Para crear un nivel de RAID 0+1, primero debemos crear un stripe y posteriormente se espejean. Para crear un RAID 1+0 primero espejamos y posteriormente se generan los stripes.

Aparentemente ambos niveles nos brindan la misma seguridad pero a medida que vamos perdiendo discos debido a fallas observamos las bondades del raid 1+0.

Otra ventaja del RAID 1+0 es su bajo costo de sincronización ya que si falla un disco, únicamente tendremos que sincronizar el tamaño del disco y no todo el stripe a diferencia de RAID 0+1.

Raid de paridad: RAID 3 y RAID 5.

Los niveles de RAID 3 y RAID 5 son conocidos como RAID de paridad porque su objetivo no es mantener una copia completa de los datos para contar con redundancia, ya que sólo requieren de aproximadamente el espacio de un disco para contar con redundancia, en este disco extra se maneja la paridad de los datos y esta se calcula a través de operaciones or exclusivas para reconstruir el dato que existía en el disco dañado. En RAID 3 y RAID 5 solo tenemos como máximo un disco de redundancia, esto quiere decir que sí perdemos dos discos nuestros datos estarán inaccesibles. Cuando el RAID se maneja por software y perdemos un disco se dice que estamos trabajando en modo degradado y el performance de nuestro equipo disminuye considerablemente, ya que se ocupan ciclos de CPU para reconstruir la información que existía en el disco dañado.

Paridad dedicada.

Para crear un RAID 3 es necesario contar como mínimo con 3 discos de los cuales dos discos se ocuparán para almacenar datos y uno más para manejar únicamente la paridad. Debido a que la paridad se almacena en un sólo disco, este nivel de RAID no es utilizado con frecuencia en la industria y sus implementaciones solo son vistas en arreglos de discos que realizan RAID por hardware.

Debido a que existe un sólo disco para la paridad se crea un cuello de botella, ya que en cada escritura a disco se debe calcular la paridad y por ende existe carga de I/O en el disco de paridad, este nivel de RAID goza de un performance bajo en términos generales.

Paridad distribuida.

Es muy similar a RAID 3 de paridad dedicada a diferencia que el cuello de botella generado por un disco único de paridad es eliminado. Debido a que es un implementación más eficiente que RAID 3, también encontramos herramientas que realizan este RAID por software como Solstice DiskSuite y Veritas Volume Manager.

Todos los niveles de RAID anteriores es posible crearlos con las herramientas VERITAS Volume Manager y Solstice DiskSuite. Para entender mejor como funciona la tecnología RAID por software, explicaremos como trabajan ambos manejadores de discos.

Funcionamiento y Conceptos de VERITAS Volume Manager

VERITAS Volume Manager es un software manejador de discos que crea un nivel virtual de manejo de discos físicos utilizando para ello "objetos virtuales" de almacenamiento de información. A lo largo de este punto explicaremos algunos conceptos de este software que nos permitirán tener un mejor entendimiento de este trabajo. Comencemos por definir que es un volumen.

Concepto de Volumen

Un **volumen** es un objeto virtual, creado por Volume Manager, que sirve para almacenar datos. Un volumen se crea con espacio de uno o más discos físicos en los que los datos son almacenados físicamente. Los volúmenes creados son reconocidos por el sistema operativo como si fueran discos físicos, al igual que las aplicaciones que hacen uso de esos volúmenes. Más adelante complementaremos esta definición.

Con Volume Manager, se habilita almacenamiento de datos de una forma virtual poniendo los discos físicos bajo su control, lo cual significa que se crean objetos virtuales y se establecen conexiones lógicas entre esos objetos y los objetos físicos o discos.

Definición de Disco de Volume Manager

Cuando un disco físico es puesto bajo el control de Volume Manager ocurre lo siguiente:

1.- Volume Manager remueve todas las entradas de la tabla de particiones del VTOC, excepto la que corresponde a la partición 2, ya que esta última contiene información del disco entero, incluyendo el VTOC, el cual es utilizado para determinar el tamaño del disco.

2.- Después, Volume Manager reescribe el VTOC y crea dos particiones en el disco físico. Una partición es llamada región privada y la otra región pública.

Región privada. La región privada almacena información, como lo son cabezas de disco, copias de configuración, y archivos de registro que Volume Manager usa para manejar objetos virtuales.

Región pública. La región pública está compuesta del espacio restante del disco y representa el espacio disponible que Volume Manager puede utilizar para asignarlo a un volumen y es donde las aplicaciones guardan sus datos.

3.- Volume Manager actualiza el VTOC con información referente a particiones existentes o nuevas particiones.

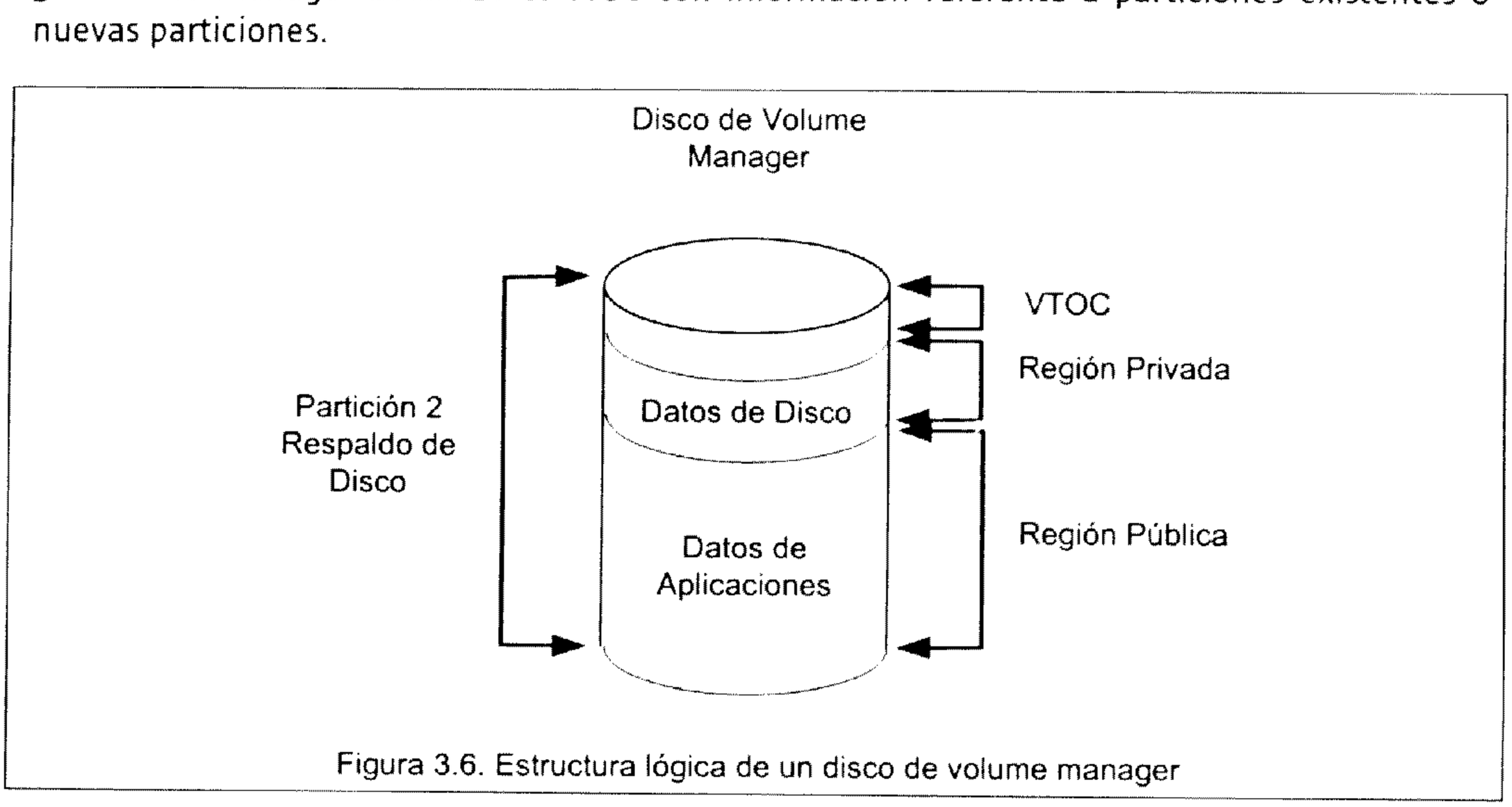


Figura 3.6. Estructura lógica de un disco de volume manager

Objetos Virtuales de Almacenamiento

Los volúmenes son parte de una variedad de tipos de objetos virtuales utilizados por Volume Manager para manejar el almacenamiento de datos.

Objetos de Veritas Volume Manager

Los objetos virtuales de Volume Manager incluyen los siguientes:

Discos de Volume Manager

Como ya vimos un disco de Volume Manager es creado a partir de la región pública de un disco físico que está bajo el control de Volume Manager. Cada disco de Volume Manager corresponde a un disco físico. Cada uno de estos discos tiene un nombre de disco único el cual es un nombre lógico usado para propósitos administrativos, y es asignado cuado el disco es agregado al control de Volume Manager. Cuando esto sucede, dentro de Volume Manager ya no haremos referencia a un disco utilizando la nomeclatura c#t#d#, sino que ahora utilizaremos el nombre lógico asignado por el administrador.
Es importante señalar que el administrador puede asignarle un nombre lógico a cualquier disco, excepto a los discos que están en el grupo llamado rootdg, ya que a los discos de este grupo Volume Manager les pone un nombre. Más adelante hablaremos de este grupo de discos.

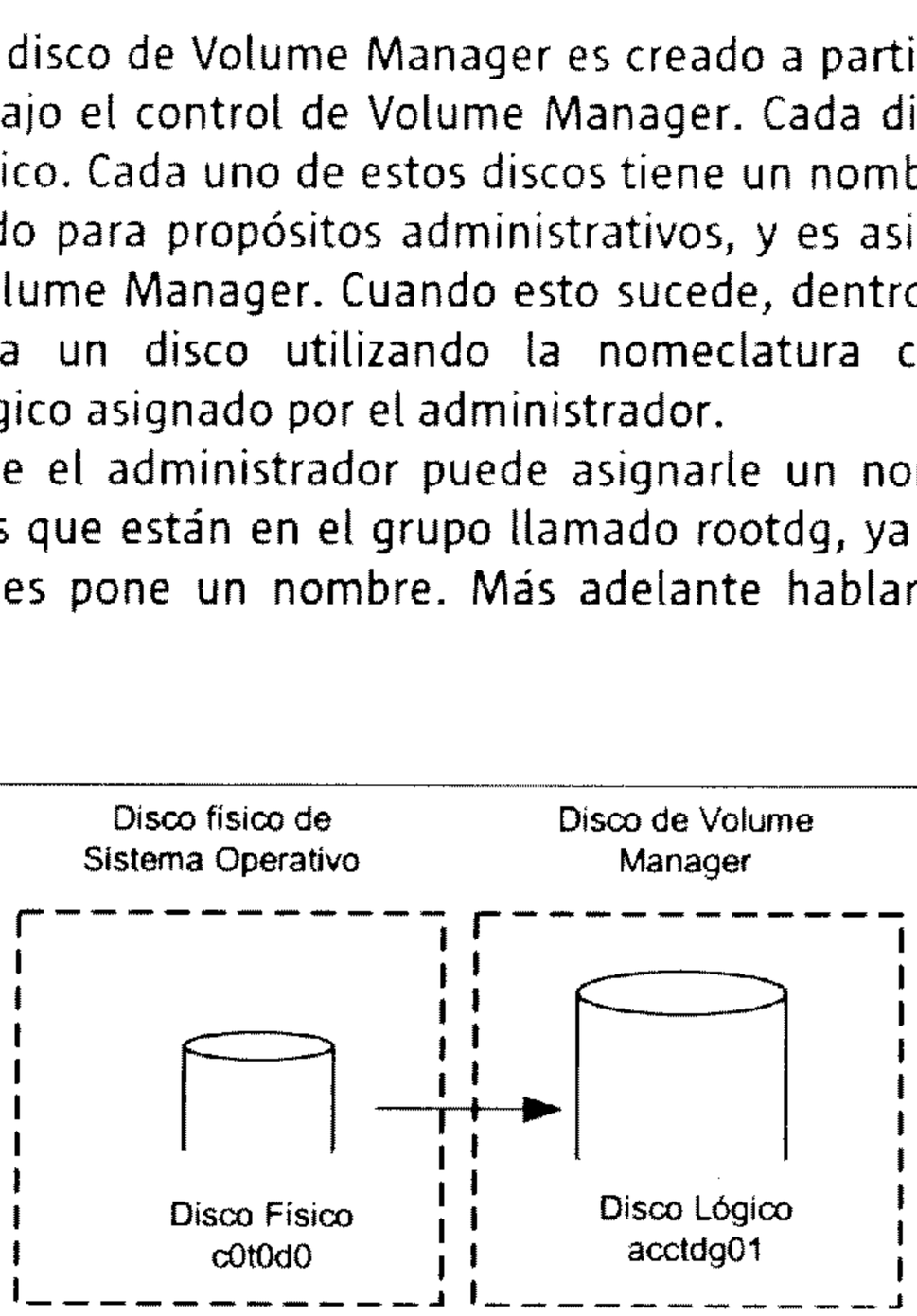


Figura 3.7. Nombre de un disco de volume manager

Grupos de Discos

Un grupo de discos es un conjunto de discos que están bajo el control de Volume Manager. Los discos se agrupan para propósitos de un mejor manejo de la información, como puede ser mantener los datos de una aplicación específica. Por ejemplo los datos de aplicaciones de contabilidad pueden ser organizados en un grupo de discos llamados acctdg.

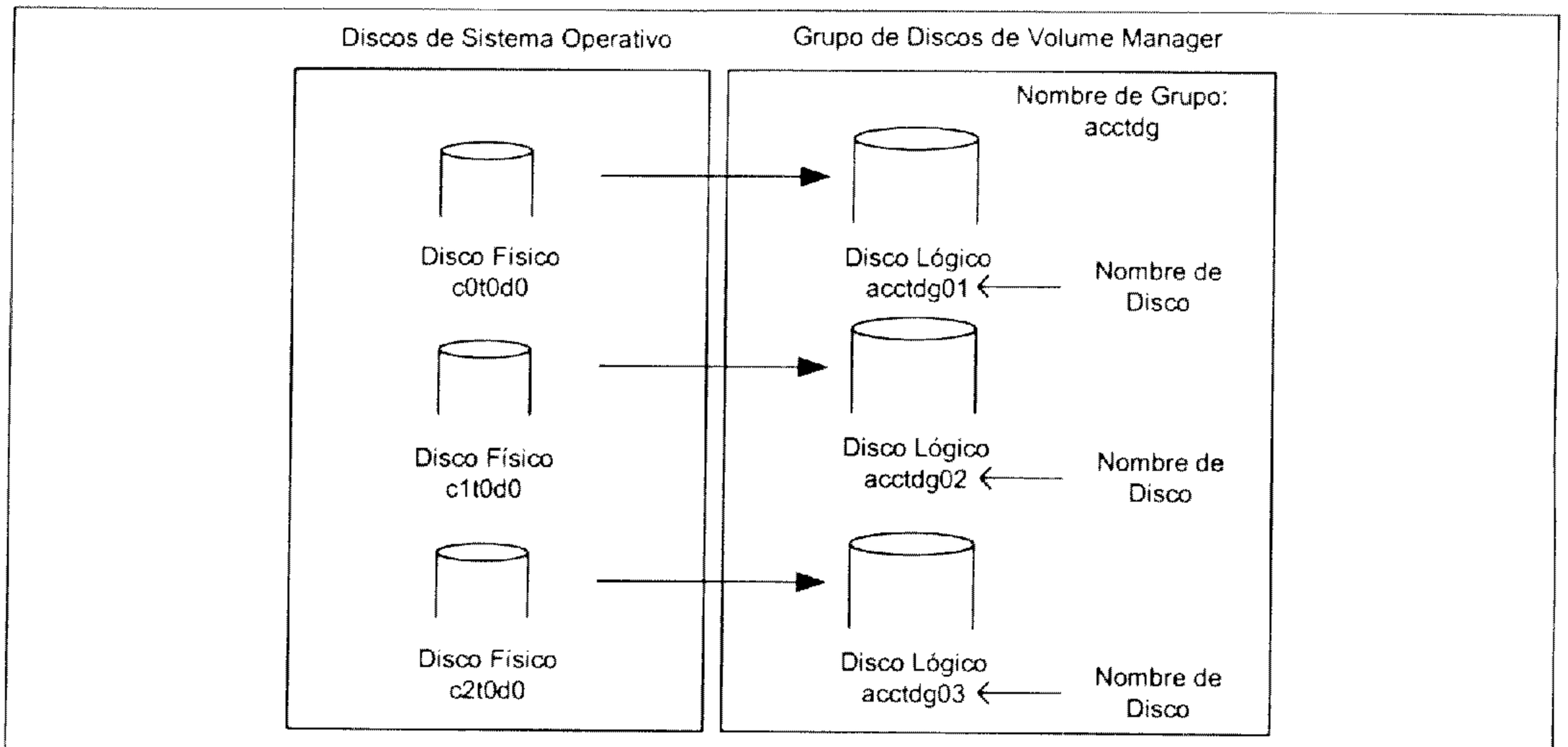


Figura 3.8. Grupo de discos de volume manager

Subdiscos

Un disco que se encuentra bajo el control de Volume Manager puede ser dividido en una o más partes llamadas subdiscos. Un subdisco es una partición de la región pública de un disco que está bajo el control de Volume Manager, bajo este contexto es la unidad mas pequeña de almacenamiento. Un disco de Volume Manager puede contener varios subdiscos, pero un subdisco no puede compartir con otro subdisco la misma porción de disco. Por convención el nombre de un subdisco toma la forma nombre_ grupo-##. Por ejemplo si tenemos un disco llamado acctdg01 dividido en tres, el primer subdisco se llamará acctdg01-01, el segundo acctdg-02, el tercero acctdg-03 y así sucesivamente.

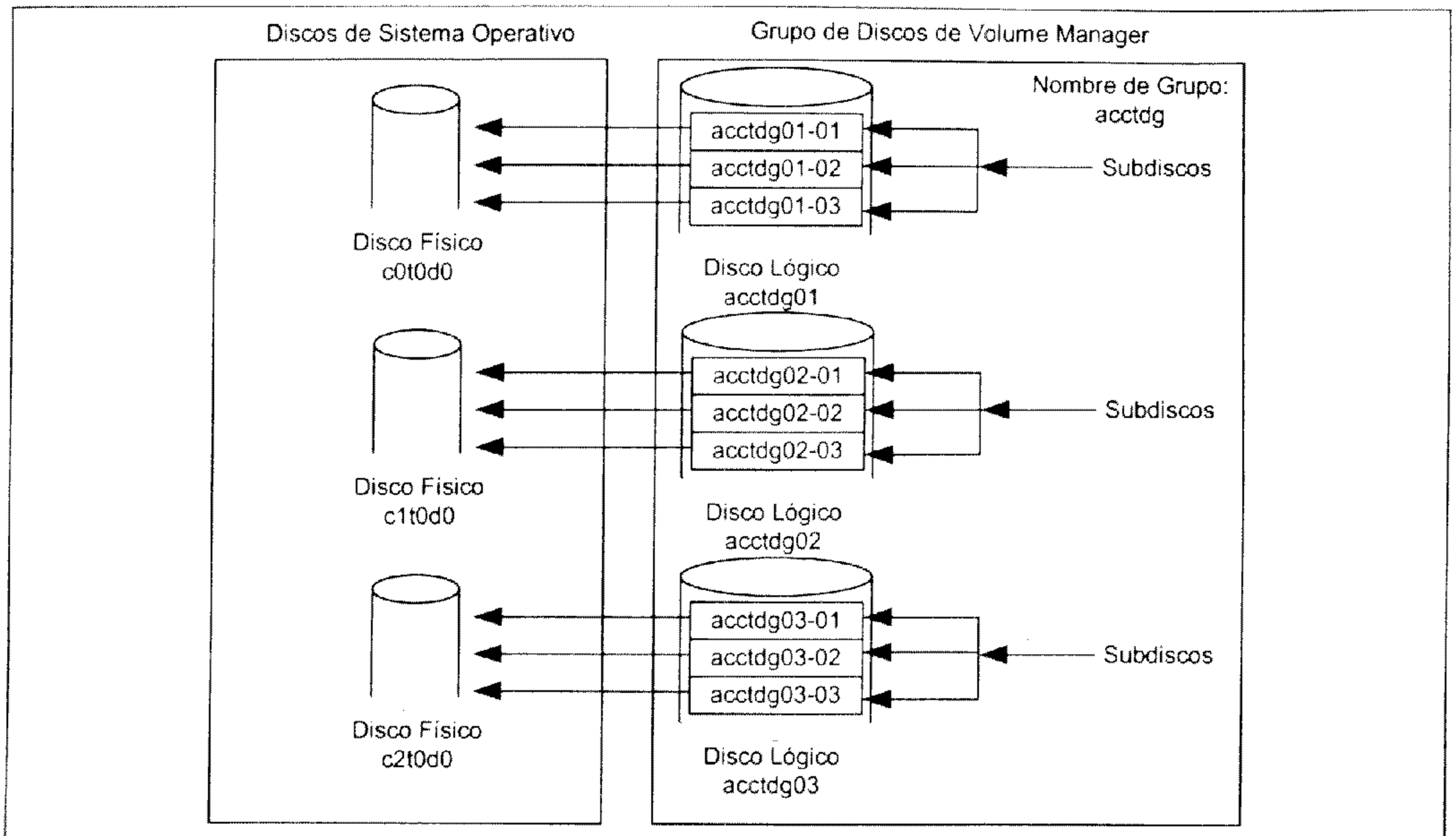


Figura 3.9. Subdiscos de volume manager

Plexes

Volume Manager utiliza subdiscos para construir objetos virtuales llamados plexes. Un plex es un conjunto ordenado de subdiscos que representan una copia de datos en un volumen. Un plex está compuesto de uno o más subdiscos localizados en uno o más discos físicos. Por convención el nombre de un plex en un volumen es nombre-volumen-##. Por ejemplo si tenemos un volumen llamado payvol el cual tiene dos plexes, el primer plex se llamará payvol-01 y el segundo plex payvol-02, y así sucesivamente.

Volúmenes

Un volumen es un dispositivo virtual de almacenamiento de datos utilizado en una forma similar a un disco físico. Un volumen está compuesto de uno o más plexes y puede estar formado a través de múltiples discos. Un volumen debe ser configurado a partir de discos y subdiscos que estén bajo el control de Volume Manager y que pertenezcan al mismo grupo. Por convención el nombre de un volumen de Volume Manager es nombre-volumen##. Por ejemplo podemos tener un volumen llamado payvol o payvol01, aunque podemos prescindir del par de números que están junto al nombre.

La siguiente figura muestra cómo están constituidos tanto un plex como un volumen.

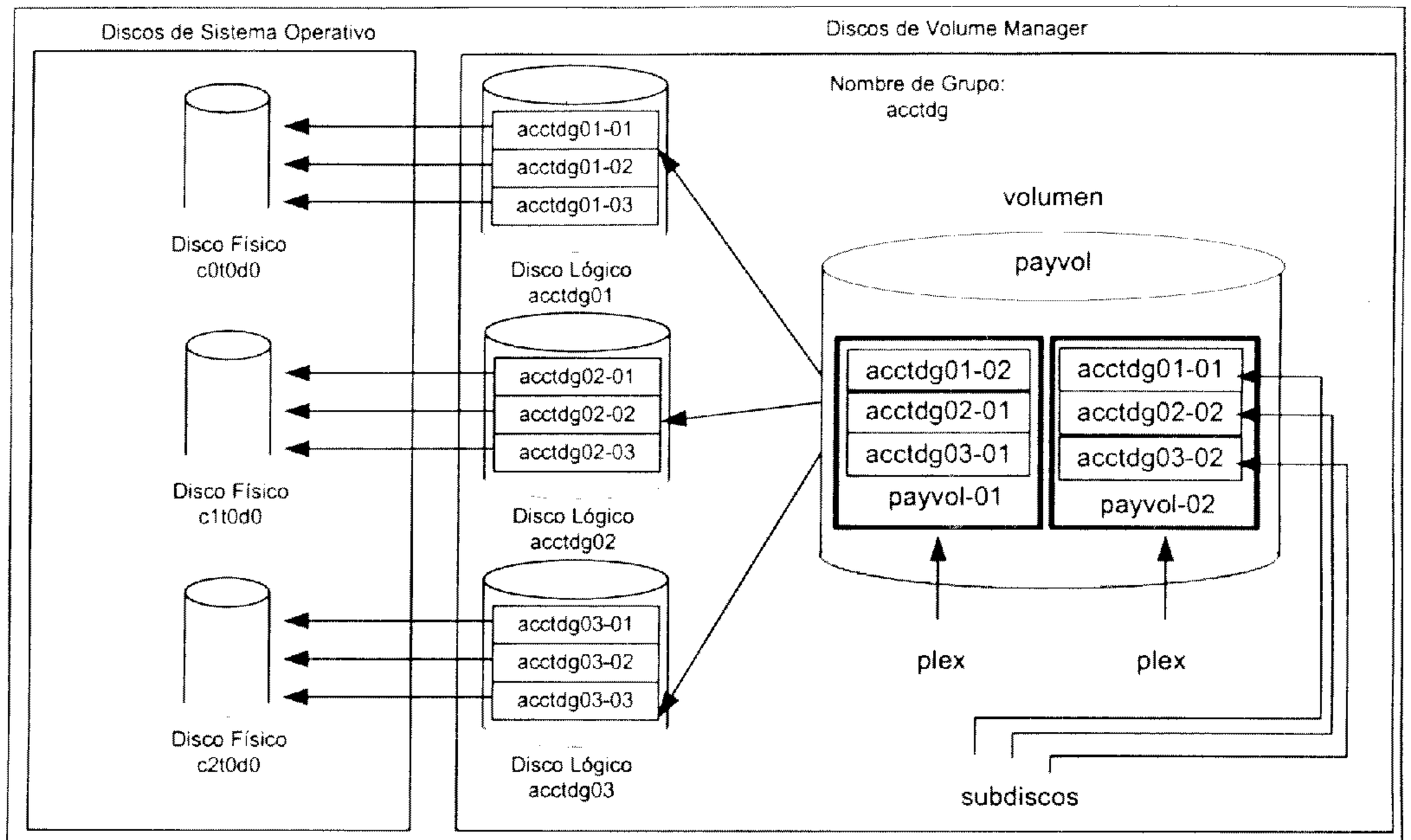


Figura 3.10. Volúmenes y plexes de volume manager

Tipos de Distribución de Volúmenes

La distribución de volúmenes se refiere a cómo están organizados los plexes dentro de un volumen. Es la forma en que los plexes están configurados para manejar el espacio de direcciones de volumen mediante el cual las operaciones de lectura/escritura son realizadas. La distribución de volúmenes está basada en la habilidad de Volume Manager de combinar lógicamente discos físicos con el propósito de almacenar datos a través de múltiples discos.

Actualmente existe una variedad disponible de distribución de volúmenes, y cada una de ellas tiene diferentes ventajas y desventajas. El tipo de distribución que se elija dependerá del nivel de funcionamiento y confiabilidad requerida para nuestro sistema.

Con Volume Manager podemos cambiar la distribución de nuestros volúmenes sin interrumpir las aplicaciones o sistemas de archivos que están haciendo uso del volumen. Una distribución de volumen puede ser configurada, reconfigurada, o modificada en su tamaño en línea. Los principales tipos de distribución de volúmenes se muestran y explican a continuación.

Concatenado

En un volumen concatenado los subdiscos son ordenados secuencial y contiguamente. La concatenación permite que los volúmenes sean creados a partir de múltiples regiones de uno o más discos si no hay espacio suficiente para un volumen entero sobre una región única de un disco.

- Ventajas

Elimina las restricciones de tamaño. La concatenación elimina las restricciones de tamaño de dispositivos de almacenamiento impuestas por el tamaño de discos físicos.
Mejor utilización de espacio libre. Permite una mejor utilización del espacio libre de los discos.
Simplifica la administración. La concatenación permite crear sistemas de archivos largos y sobre todo reduce la complejidad de la administración del sistema.

- Desventajas

No tiene protección contra fallas de discos. La concatenación no nos protege contra fallas de discos. Una sola falla de algún disco puede ocasionar la falla por completo de un volumen.

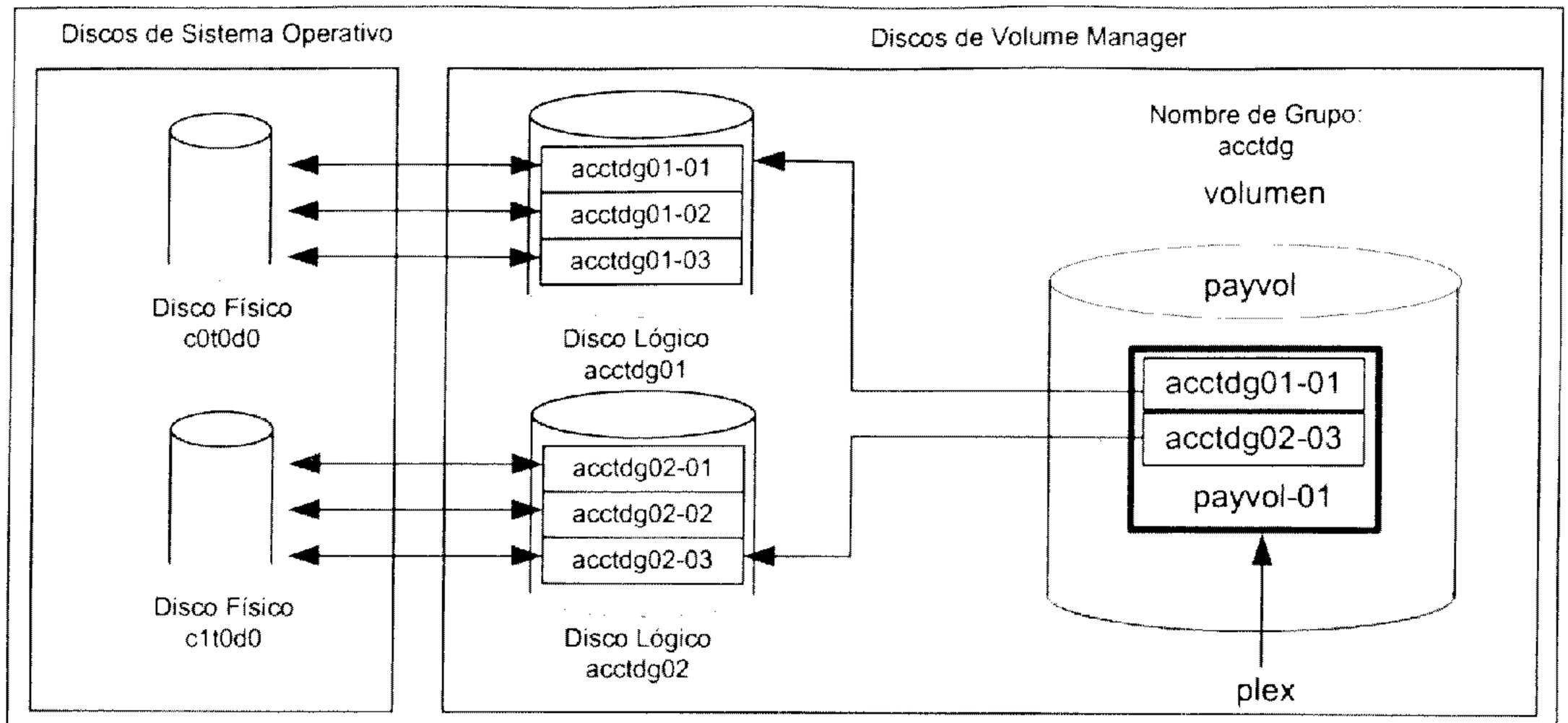


Figura 3.11. Distribución de concatenado

Distribución de striping

En un volumen de este tipo los datos son distribuidos a través de múltiples discos. Los subdiscos son agrupados en columnas, cada columna contiene uno o más subdiscos, los cuales, pueden derivarse de uno o más discos físicos. Todas las columnas deben ser del mismo tamaño. Los datos son alojados en unidades de igual tamaño llamados unidades striping las cuales están distribuidas a lo largo de las columnas. Este tipo de distribución requiere de al menos dos subdiscos para formar un plex cada uno de los cuales debe pertenecer a diferentes discos.

- Ventajas

Transferencia de datos en paralelo. La distribución de striping es de gran ayuda si necesitamos escribir o leer cantidades largas de datos de manera rápida desde un disco físico usando transferencia de datos en paralelo a múltiples discos.
Balanceo de cargas. Es también recomendable en balanceo de cargas de entrada/salida de aplicaciones multiusuario a través de múltiples discos.
Mejora el funcionamiento. Una mejora en el funcionamiento es obtenida con el incremento del ancho de striping efectivo de los canales de datos de entrada/salida.

- Desventajas

No tiene redundancia. La distribución de striping por si sola no ofrece redundancia o características de recuperación.
Fallas de discos. La distribución de striping de un volumen incrementa la posibilidad de que una falla de disco ocasione una falla en el volumen que contiene parte de ese disco. Por ejemplo si tenemos tres volúmenes de striping a través de dos discos, y uno de los discos es usado por dos volúmenes, entonces si ese disco falla, ambos volúmenes fallarán.

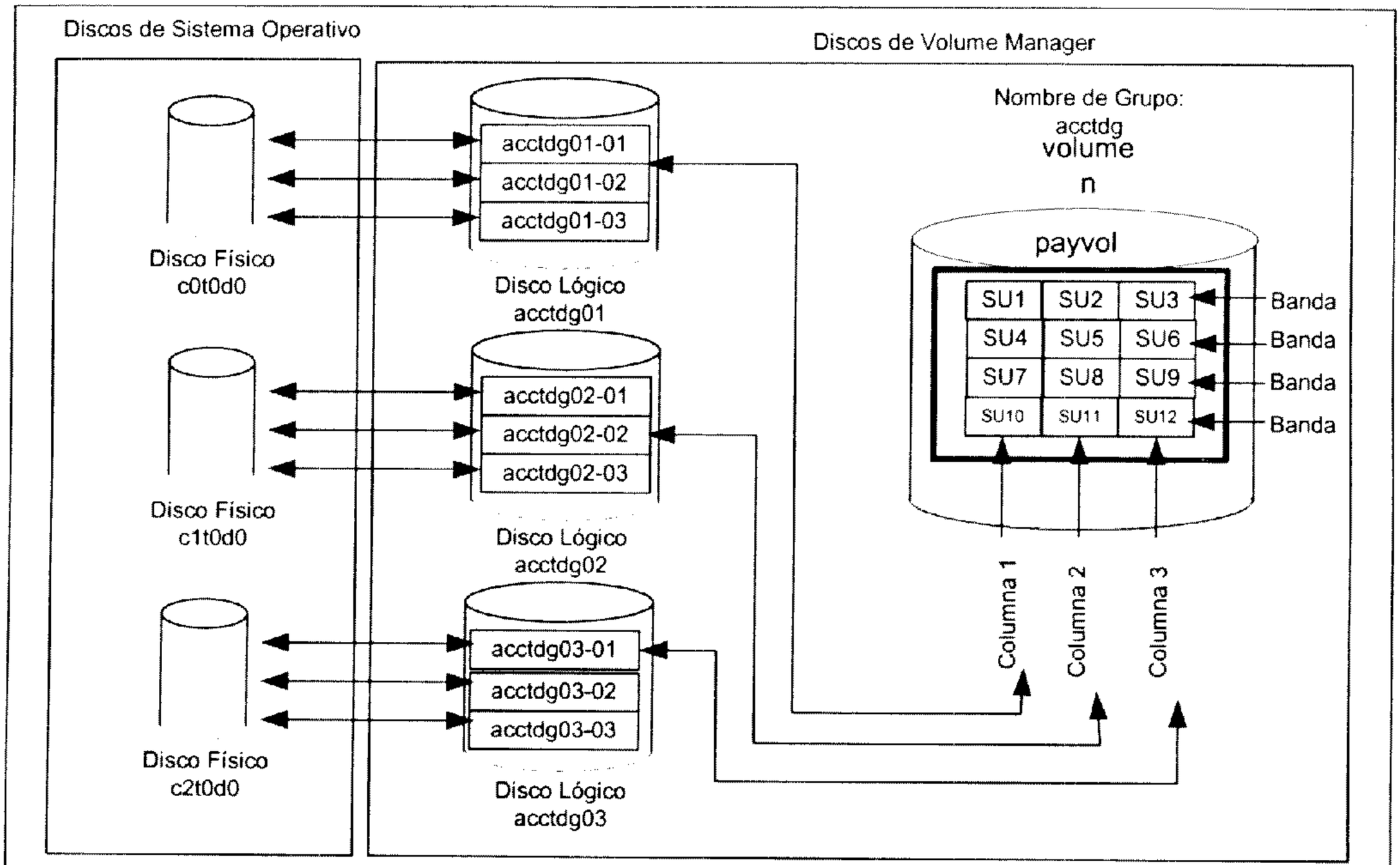


Figura 3.12. Distribución de banda

Mirror

Un volumen espejeado utiliza varios plexes para duplicar la información contenida en él. Aunque un volumen puede tener un solo plex, por lo menos son requeridos dos plexes o espejos para realmente tener redundancia de datos. Cada uno de estos plexes debe contener espacio de diferentes discos para que la redundancia sea útil. Los espejos que utiliza Volume Manager permiten que todas las copias de datos sean las mismas todo el tiempo. Cuando ocurre una escritura a un volumen, todos los plexes reciben esa escritura. Si ocurre una falla física de disco y el plex que contiene ese disco deja de funcionar, el sistema puede seguir funcionando utilizando otro plex que no ha sido afectado. Un volumen puede tener plexes o espejos que tengan distribución de concatenado o de striping o que tengan igual distribución.

- Ventajas

Mejora la confiabilidad y la disponibilidad de datos. Con concatenado o striping, la falla física de un disco puede hacer que un plex o volumen este inservible. Con el espejeo los datos son protegidos contra la falla física de algún disco. El espejeado mejora la confiabilidad y la disponibilidad de los datos de un volumen concatenado o de striping.

Mejora la lectura a disco. Al tener varios plexes de donde obtener los datos, se obtienen con ello beneficios de lectura.

- Desventajas

Requiere más espacio de disco. El espejeo requiere al menos dos veces más de espacio de disco, lo cual puede ser muy costoso para configuraciones grandes.
Escritura ligeramente lenta. La escritura de datos a los volúmenes es ligeramente lenta, porque múltiples copias de datos tienen que ser escritas en paralelo, aunque en realidad esta no es una razón de suficiente peso para no usar el espejeado.

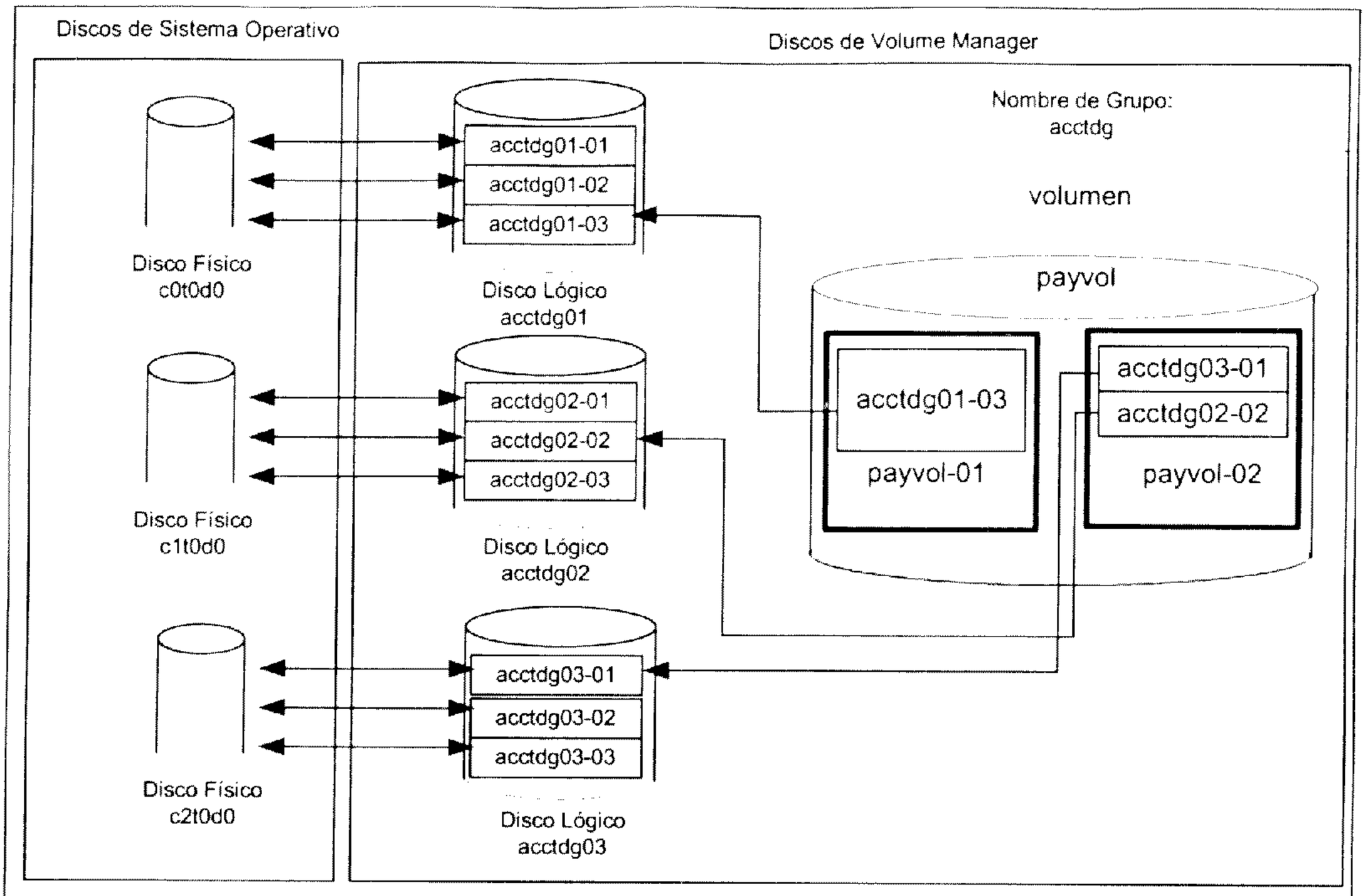


Figura 3.13. Distribución de espejo

RAID-5

Una distribución de volumen RAID-5 tiene los mismos atributos que un plex de striping, pero incluye una columna adicional de datos que es usada para la paridad. La paridad provee redundancia y es usada para reconstruir datos si un disco físico falla. En comparación con el funcionamiento de la distribución de striping, la escritura de los volúmenes de RAID-5 es más lenta, ya que la información de paridad necesita ser actualizada cada vez que el dato es accesado. Aunque en comparación con el espejeo, el uso de la paridad reduce la cantidad de espacio de disco requerido. RAID-5 requiere como mínimo tres discos para datos y paridad. RAID-5 no puede ser espejeado. Cuando se implementa con archivos de registro o logs, un disco adicional es requerido.

- **Ventajas**

Paridad a través de redundancia. Con una distribución RAID-5, los datos pueden ser recreados a partir de datos restantes y la paridad en caso de alguna falla de disco.
Requiere menos espacio que el espejeado. RAID-5 almacena información de paridad, en lugar de una copia completa de datos.
Mejora la lectura. RAID-5 proporciona mejoras similares a la distribución de striping en la lectura de datos.
Recuperación rápida a través de logs. Los logs de RAID-5 minimizan el tiempo de recuperación en caso de fallas de disco.

- **Desventajas**

Escritura Lenta. La escritura de datos a disco tiende a ser lenta, ya que implica leer el dato anterior y la paridad, calcular la nueva paridad, y escribir el nuevo dato y la nueva paridad.

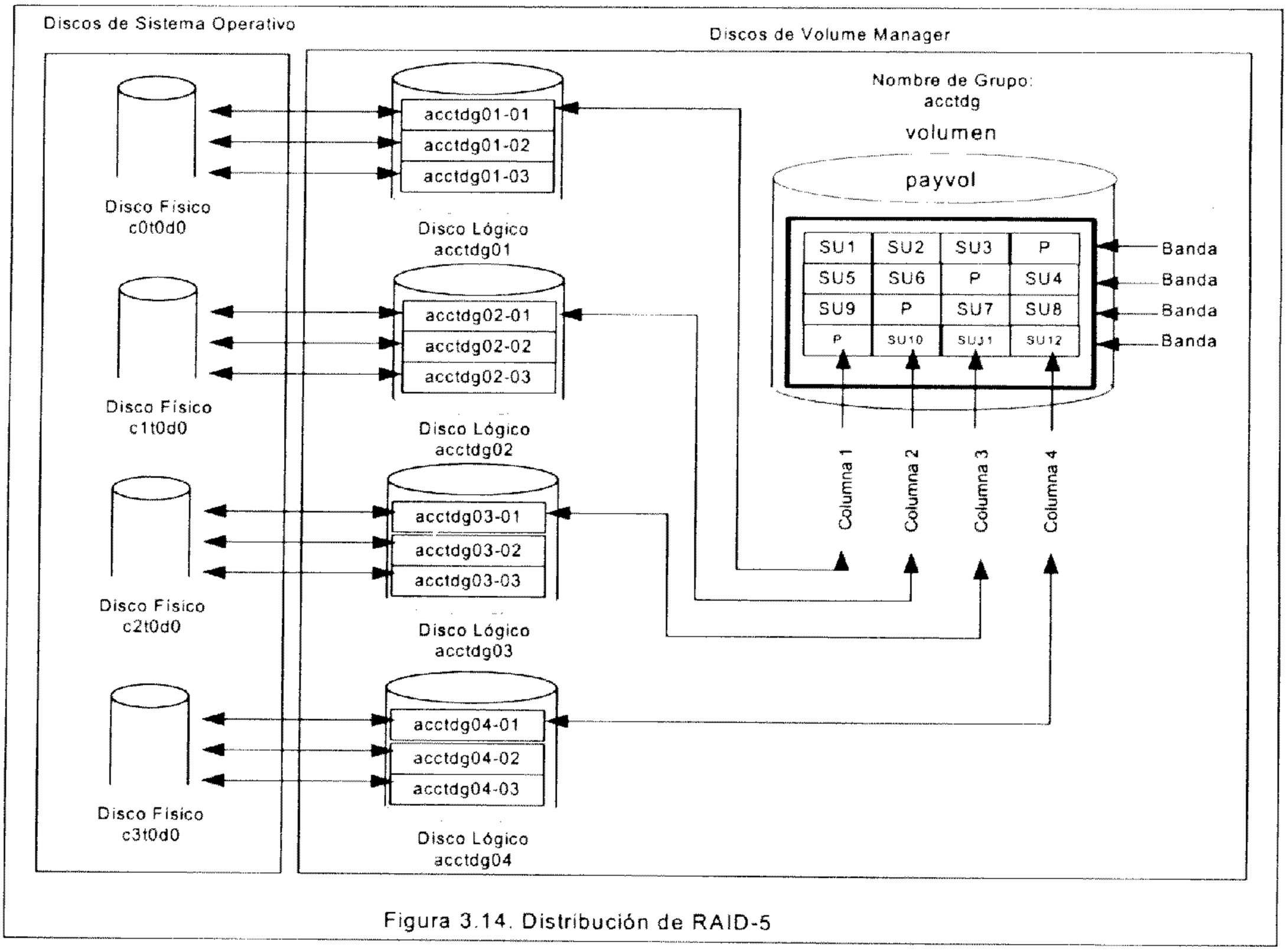


Figura 3.14. Distribución de RAID-5

Niveles de la tecnología RAID soportados

Como vimos anteriormente RAID acrónimo de Redundant Array of Independent Disks. RAID es un acceso al manejo de almacenamiento de datos en el cual un arreglo de discos es creado, y una parte de la capacidad de almacenamiento combinado de los discos es usada para almacenar información duplicada de datos en un arreglo. Para mantener arreglos de discos redundantes, se pueden generar datos en el caso de que un disco falle.

Los modelos de configuración RAID están clasificados en términos de niveles RAID, los cuales son definidos por el número de discos en el arreglo, el canal de datos distribuido a través de múltiples discos, y el método de redundancia usado. Cada nivel RAID tiene características especiales y ventajas y desventajas de funcionamiento.

VERITAS Volume Manager soporta los siguientes niveles RAID:

- RAID-0
- RAID-1
- RAID-5
- RAID-0+1
- RAID-1+0

Funcionamiento de Solstice DiskSuite

Solstice DiskSuite es un software que sirve para manejar discos proporcionando mejor funcionamiento, mayor capacidad y mejora en la disponibilidad de los datos. En general este software tiene muchas características que tiene el software de VERITAS Volume Manager, aunque también carece de otras, maneja las mismas distribuciones de capas de discos, los mismos niveles RAID, y también hace uso de discos virtuales, solo que con otros nombres.

Veamos los conceptos y las características más importantes.

Metadispositivo

Los metadispositivos son las unidades de funcionamiento básico de DiskSuite. Un metadispositivo es un dispositivo virtual que puede estar compuesto por una o más particiones de discos físicos. Podemos configurar las particiones que componen un metadispositivo para que sean usadas como un solo dispositivo, para que tengan una configuración de concatenado o una configuración de striping.

Por convención el nombre de un metadispositivo comienza con "d#", donde # es un número que va desde 0 hasta 127. Los nombres de los metadispositivos se localizan en la ruta de sistema operativo /dev/md/dsk y /dev/md/rdsk.

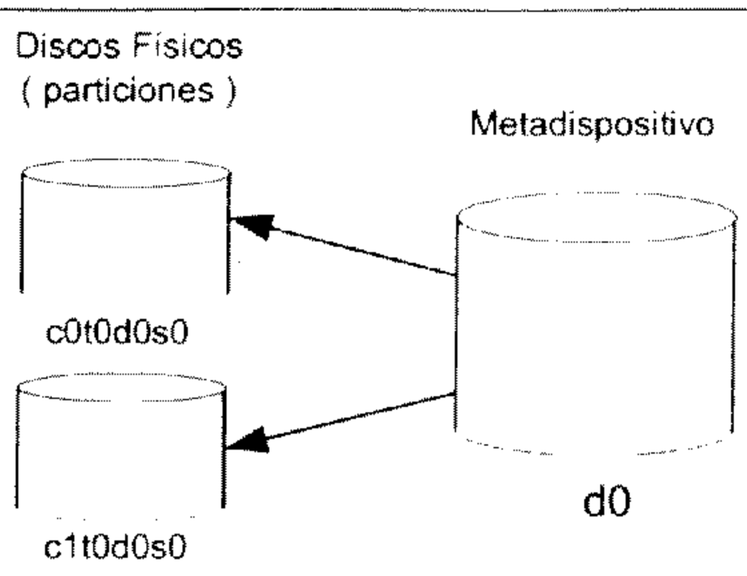


Figura 1.15. Metadispositivo de Disk Suite

Distribución de Concatenado

Las particiones de disco que componen un metadispositivo pueden utilizar las distribución de concatenado, en la cual como ya sabemos las lecturas y escrituras se hacen de manera secuencial. La siguiente figura nos muestra un metadispositivo concatenado formado por varias particiones de discos físicos diferentes.

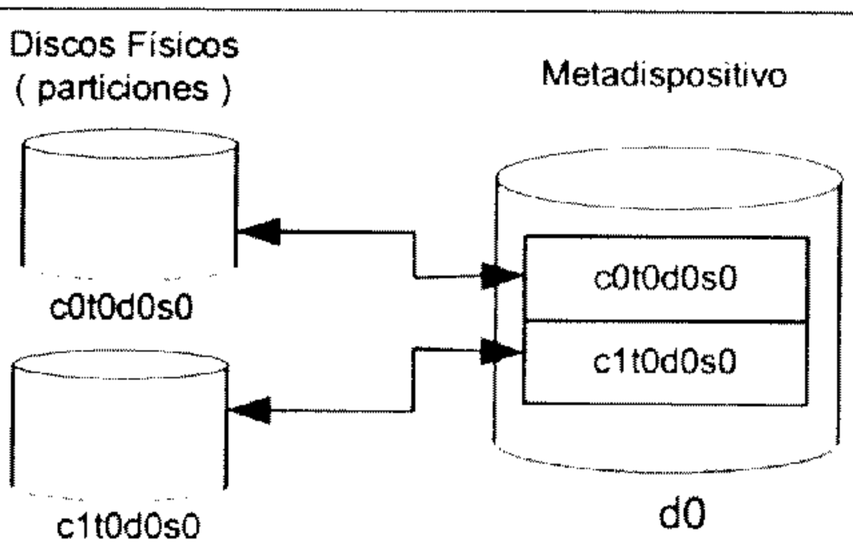


Figura 1.16. Metadispositivo con distribución de concatenado

Distribución de Striping

La distribución de striping es similar a la de concatenado, solo que como ya hemos visto, la lectura y escritura en lugar de hacerse secuencialmente se hacen a través de todas las particiones de disco que componen el metadispositivo. El tamaño de las unidades mediante las cuales se hacen las operaciones anteriores por convención es de 16 KB. Como los datos son manejados a través de la striping de discos el funcionamiento de las lecturas y escrituras tiende a mejorar.

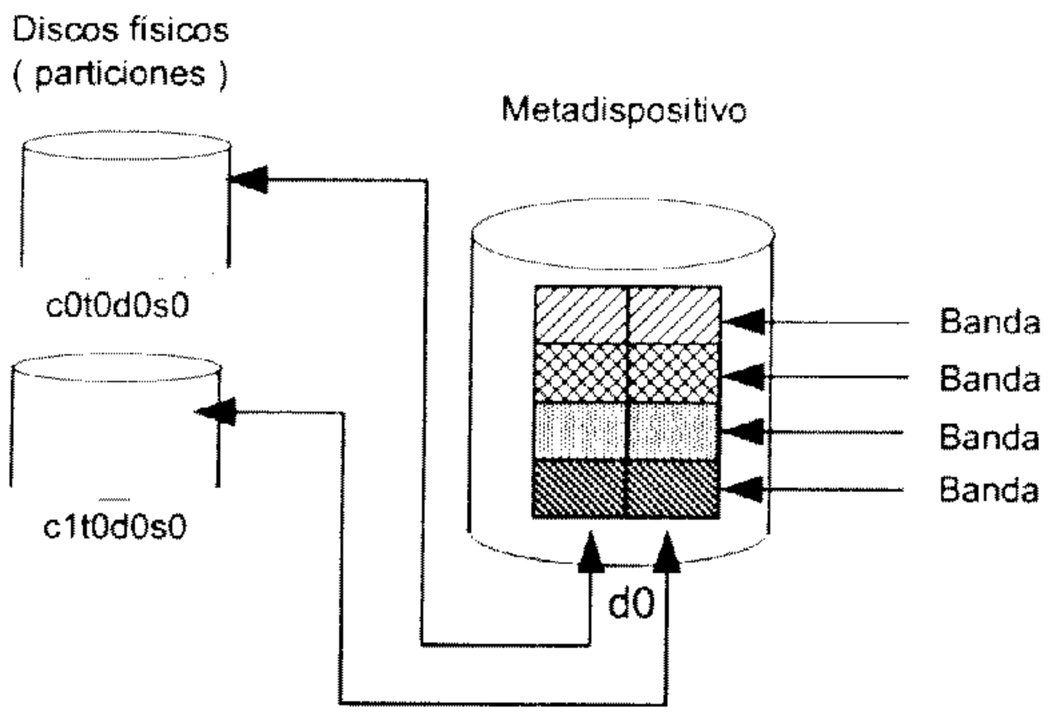


Figura 3.17. Metadispositivo con distribución de banda

Distribución de Mirror o Espejo

Para crear un espejo con DiskSuite, es necesario crear un metaespejo. Un metaespejo es un tipo especial de metadispositivo compuesto a su vez de uno o más metadispostivos. Cada metadispositivo de un metaespejo es llamado subespejo. Los metaespejos pueden tener nombres como d0, d1, d2 y así sucesivamente, la misma convención usada para metaespejos es usada para los subespejos. Por ejemplo un metaespejo puede llamarse d0 y uno de sus subespejos que lo componen puede llamarse d1, el otro subespejo puede llamarse d2.

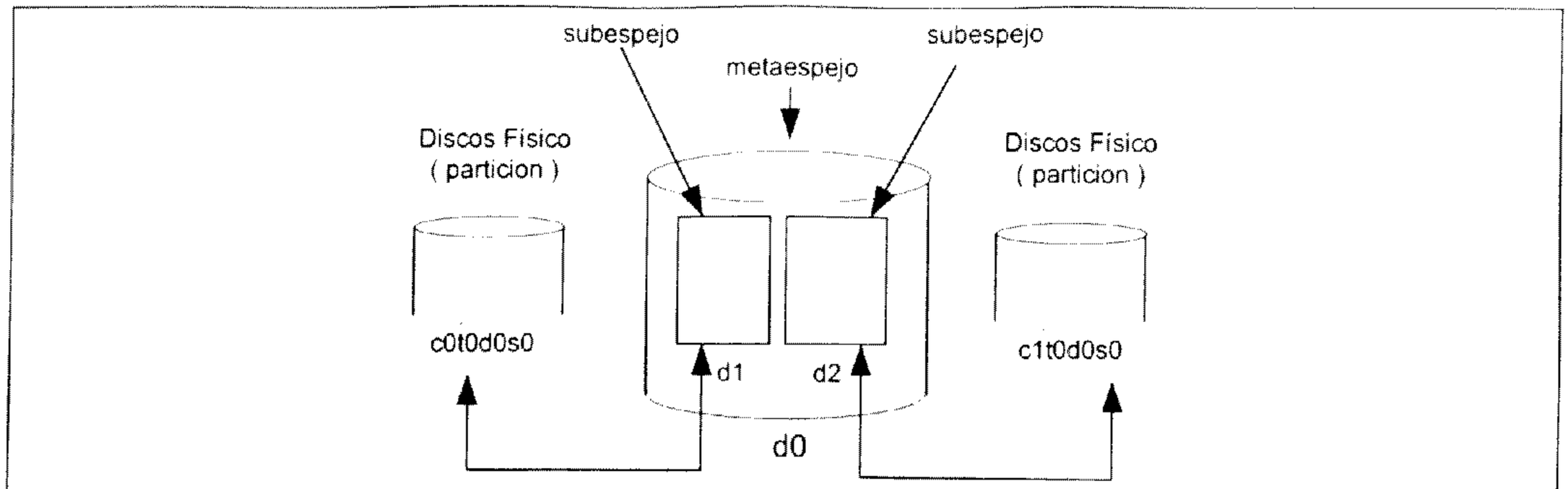


Figura 3.18. Metaespejo de Disk Suite

Distribución RAID-5

La características de RAID con que cuenta DiskSuite provee soporte para dispositivos RAID. El nivel RAID que soporta DiskSuite es RAID-5.

En el nivel RAID-5 los metadispositivos se componen de tres o más particiones físicas. En este nivel cada partición es tomada como una columna, y como ya se mencionó utiliza la paridad distribuida a lo largo de varias columnas para recuperar los datos que pudieran perderse en un momento dado. Un metadispositivo de este tipo puede ser incrementado concatenándole particiones adicionales.

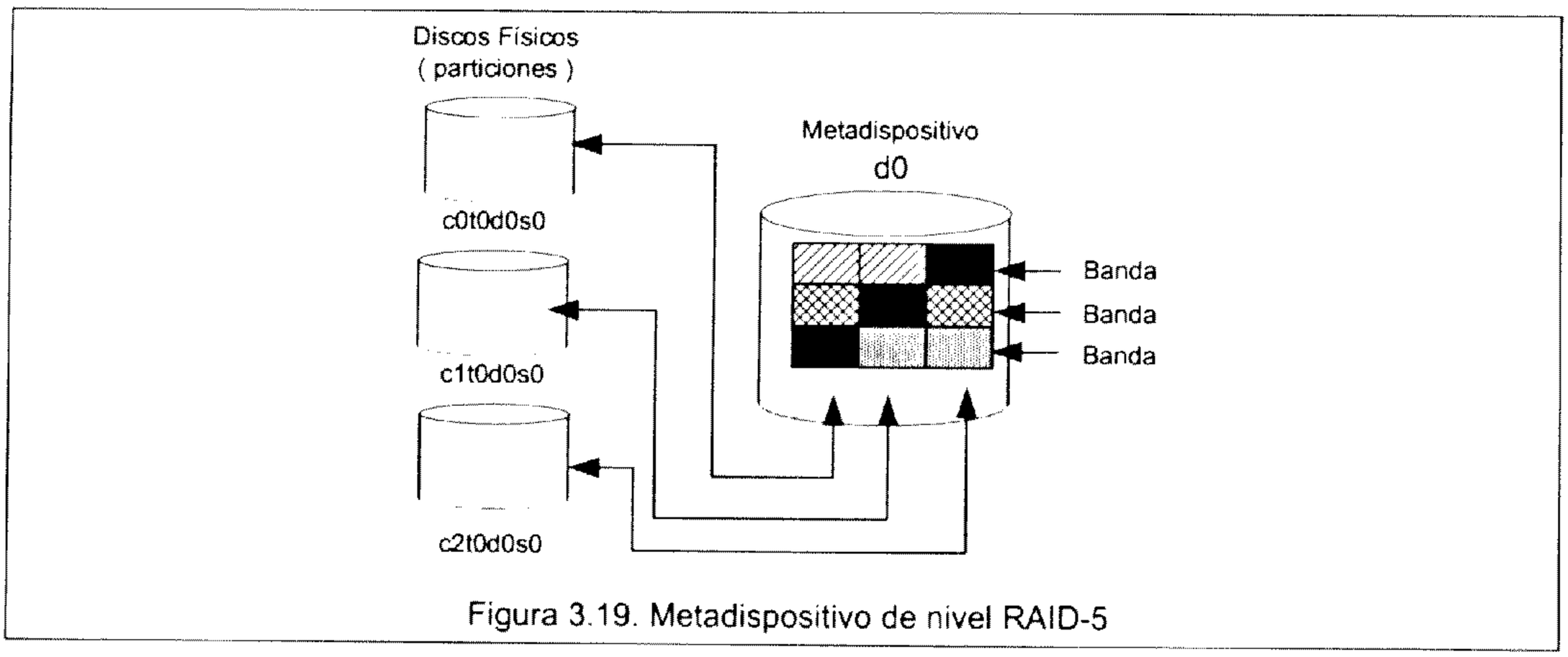


Figura 3.19. Metadispositivo de nivel RAID-5

Configuración Actual de discos de dragón

En este punto veremos cómo se encuentran configurados los discos del servidor web de la UNAM conocido como dragón. Mencionaremos los arreglos de discos que está utilizando, como están particionados, y como están formados los sistemas de archivos, y los metadispositivos. No hablaremos del servidor llamado newton ya que su configuración se verá en el capítulo 4 de replicación de datos.

Cómo vimos en el capítulo 1 de este trabajo, el servidor web de la UNAM tiene dos cajas de discos A5000 y una caja de discos D1000. Uno de los arreglos A5000 se llama "yodo" y tiene 12 discos y el otro se llama "yodo2" y tiene 14 discos es decir esta lleno.

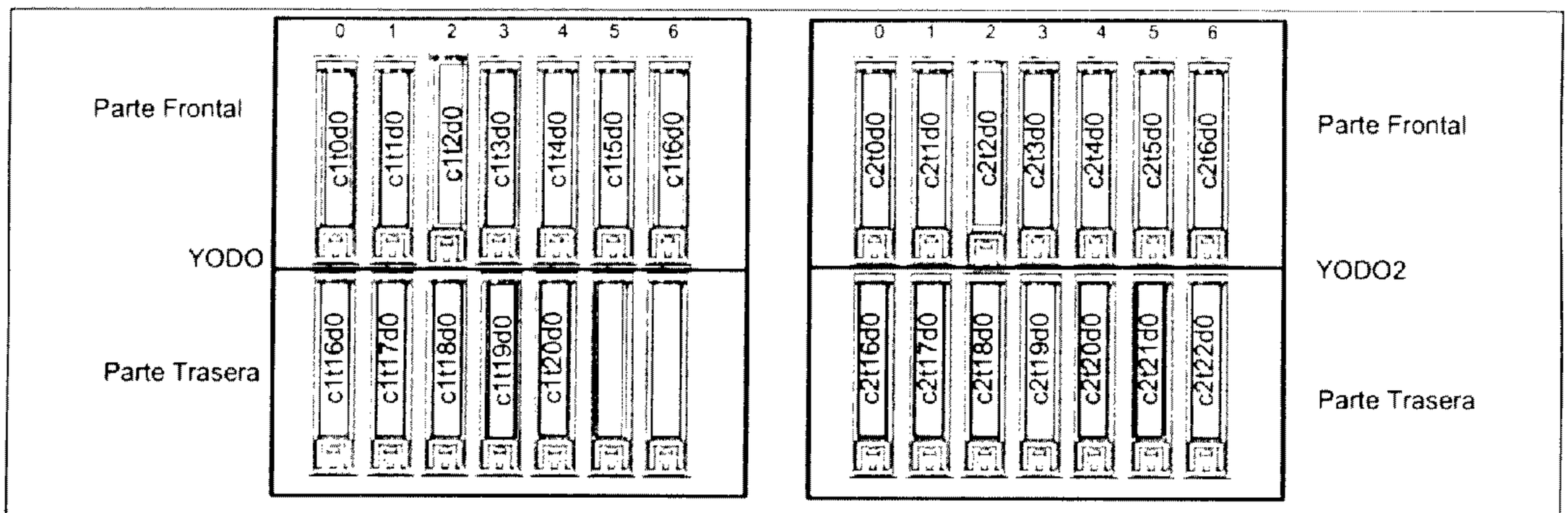


Figura 3.20. Posición de discos de arreglos A5000 de dragón

De estos 26 discos solo se están ocupando los siete discos de la parte frontal del arreglo yodo y son el disco c1t1d0 en el cual está instalado el sistema operativo de dragón, y los discos c1t1d0, c1t2d0, c1t3d0, c1t4d0, c1t5d0, c1t6d0 los cuales contienen datos de aplicaciones y usuarios, los demás discos no están siendo utilizados a pesar de que están conectados a través de fibra al servidor. Inicialmente la UNAM adquirió el arreglo llamado "yodo2" para tener redundancia de datos con el arreglo "yodo", pero por diferentes circunstancias no se había podido llevar esto a cabo, así que en este trabajo utilizaremos estos discos para lo que realmente fueron adquiridos. Todos los discos de estos dos arreglos tienen una capacidad de 18 GB.

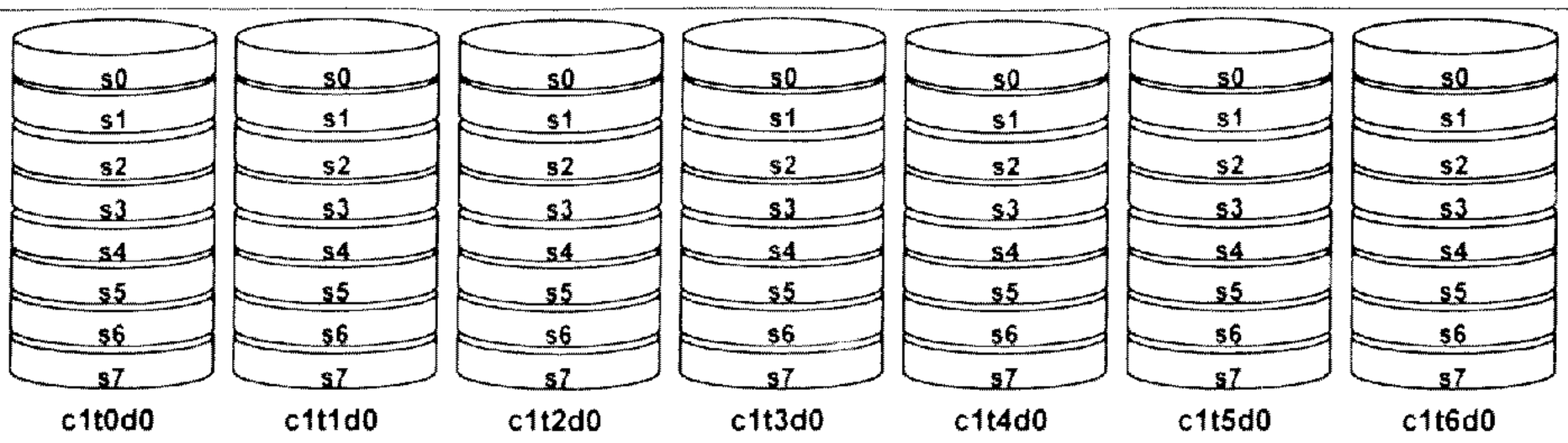


Figura 3.21. Particiones de discos utilizados del arreglo yodo

La caja de discos D1000, es un arreglo con capacidad para 12 discos, aunque solo tiene 7 discos: c0t0d0, c0t1d0, c0t2d0, c0t3d0, c0t8d0, c0t9d0, c0t10d0, los cuales contienen también información de aplicaciones y usuarios.

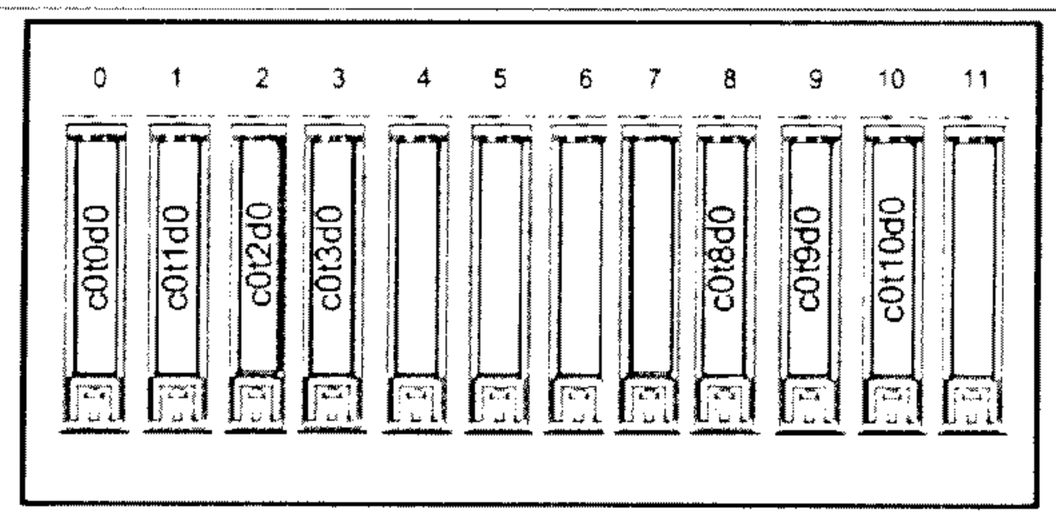


Figura 3.22. Posición de discos del arreglo D1000

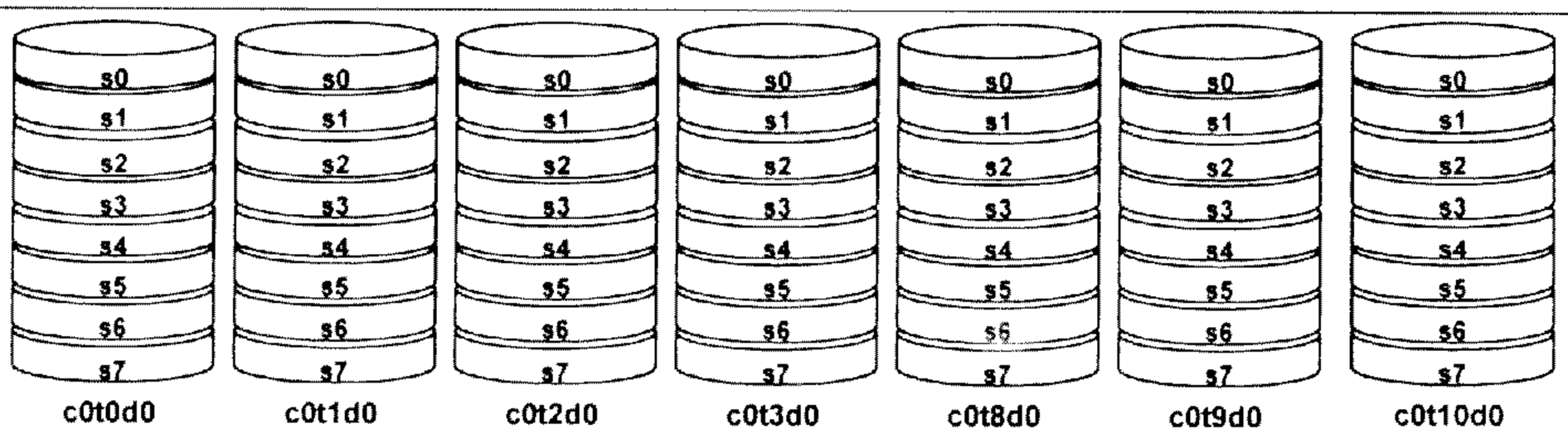


Figura 3.23. Particiones de discos utilizados del arreglo D1000

Ahora bien, los sistemas de archivos de sistema operativo están instalados sobre particiones de disco normales y no tiene ninguna redundancia, mientras que los sistemas de archivos de los datos de las aplicaciones y de los usuarios están creados con metadispositivos de Solstice DiskSuite. En seguida listamos todos los sistemas de archivos del servidor web de la UNAM, se muestran su nombre, su tamaño en kilobytes, el espacio utilizado también en kilobytes, porcentaje de utilización a que se encuentran y punto de montura.

```
Sistema de archivos  Tamaño  Espacio   Espacio %usado  Directorio
utilizado  libre           de montado

/dev/dsk/c1t0d0s0     617275  327903  233818    59%    /
/dev/dsk/c1t0d0s3    4131866 2646600 1443948    65%    /usr
/dev/dsk/c1t0d0s4    3099287 2162276  875026    72%    /var
/dev/dsk/c1t0d0s5    2056211  466484 1528041    24%    /opt
swap                 3034584     152 3034432     1%    /tmp
/dev/md/dsk/d1      30945914 27291825 3344630  90%    /mirror/home
/dev/md/dsk/d0      30945914 12978909 17657546 43%    /home/log
/dev/md/dsk/d3      6186810 4193040 1931902    69%    /mirror/users00
/dev/md/dsk/d4      6186810 3778354 2346588    62%    /home/users01
/dev/md/dsk/d5      6186810 3492786 2632156    58%    /home/users02
/dev/md/dsk/d6      6186810 5857917  267025    96%    /home/users03
/dev/md/dsk/d7      3871954 2069039 1764196    54%    /home/users04
/dev/md/dsk/d2      6186810 2881346 3243596    48%    /sybase
/dev/md/dsk/d8      3871954 1961986 1871249    52%    /raid
/dev/md/dsk/d9      3871954 1100983 2732252    29%    /raid2
/dev/md/dsk/d10     1527116 1140770  325262    78%    /usr/local/pgsql
/dev/md/dsk/d13     4743974 3615240 1081295    77%    /usr/sybase
/dev/md/dsk/d11     30957590 28151616 2496399  92%    /home
/dev/md/dsk/d12     6199998 3354000 2783999    55%    /home/users00
```

Ahora ¿por cuáles discos se encuentran formados los sistemas de archivos anteriores?, ¿qué metadispositivos tenemos?, ¿cuáles son y en que arreglo están localizados?, también veremos qué tipo de distribución tienen. Esto nos va a servir mas adelante cuando realicemos la nueva estructura de discos.

Los sistemas de archivos /, /usr, /var, /opt y /tmp se encuentran definidos sobre particiones de disco normales. El disco sobre el que se encuentran es el c1t0d0 que está en la posición 0 del arreglos A5000 llamados “yodo”. La figura siguiente ilustra como está dividido este disco.

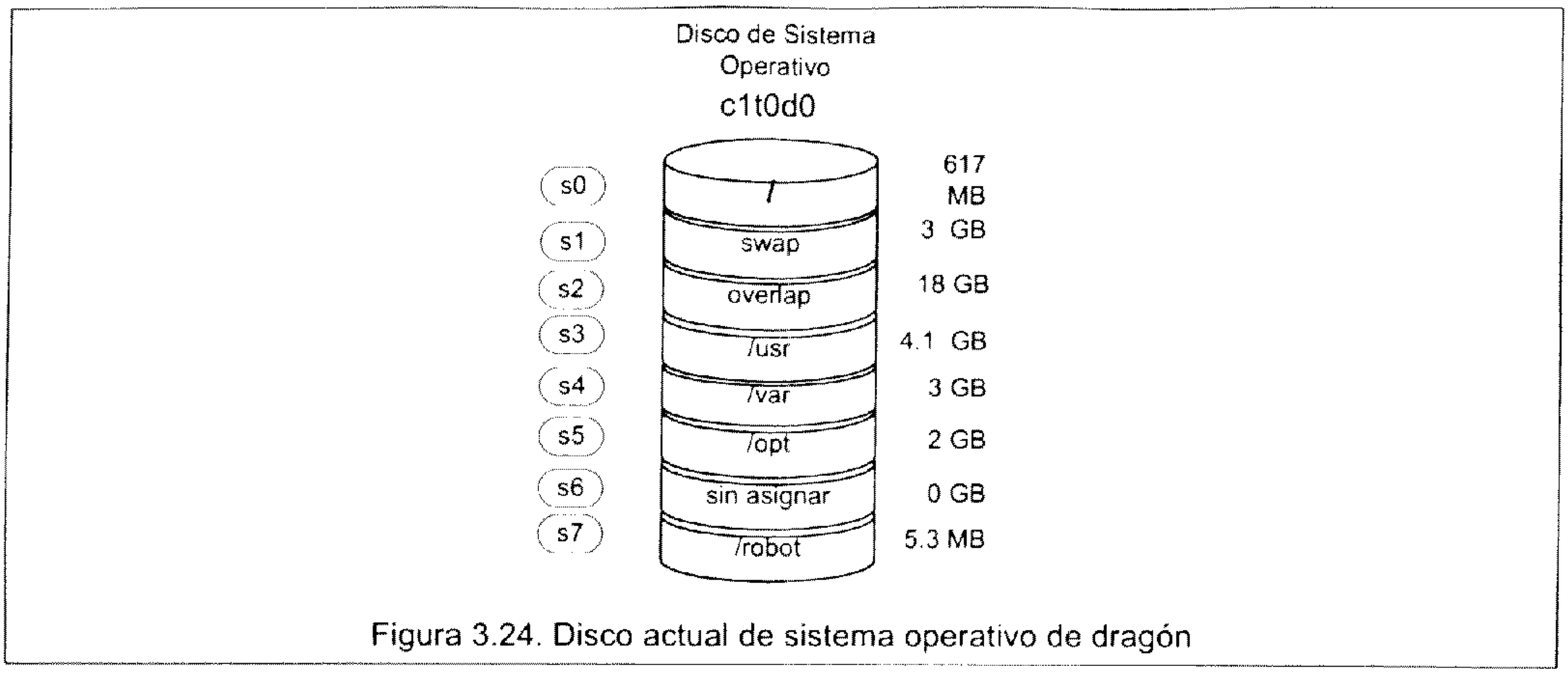


Figura 3.24. Disco actual de sistema operativo de dragón

Para los sistemas de archivos que están con DiskSuite, haremos el análisis de un metadispositivo a la vez y siguiendo un orden ascendente, es decir, comenzaremos con el metadispositivo d0, luego con el d1, d2, hasta llegar al d13.

Metadispositivo d0. Está formado por las particiones c1t1d0s1 c1t1d0s3 c1t2d0s1 c1t2d0s3 c1t3d0s1 c1t3d0s3 c1t4d0s1 c1t4d0s3 c1t5d0s1 c1t6d0s1 de los discos de la posición 1 a la 6 de la parte frontal del arreglo yodo. La distribución de discos que está utilizando es la de striping.

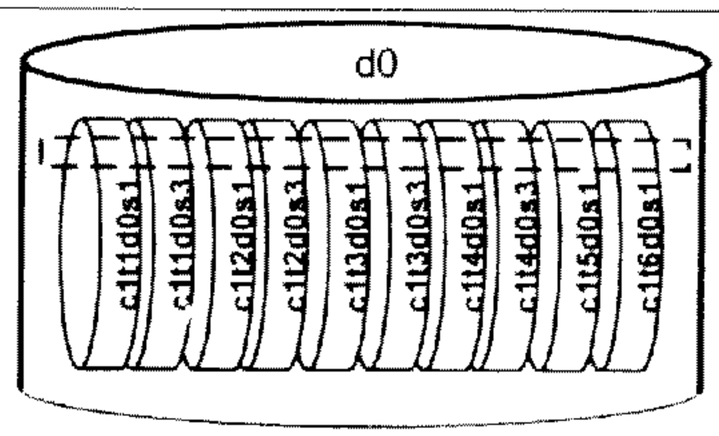


Figura 3.25-1 Metadispositivo d0

Metadispositivo d1. Está formado por las particiones c1t1d0s4 c1t1d0s5 c1t2d0s4 c1t2d0s5 c1t3d0s4 c1t4d0s4 c1t5d0s3 c1t5d0s4 c1t6d0s3 c1t6d0s4 de los discos de la posición 1 a la 6 de la parte frontal del arreglo yodo. La distribución de discos que está utilizando es la de striping.

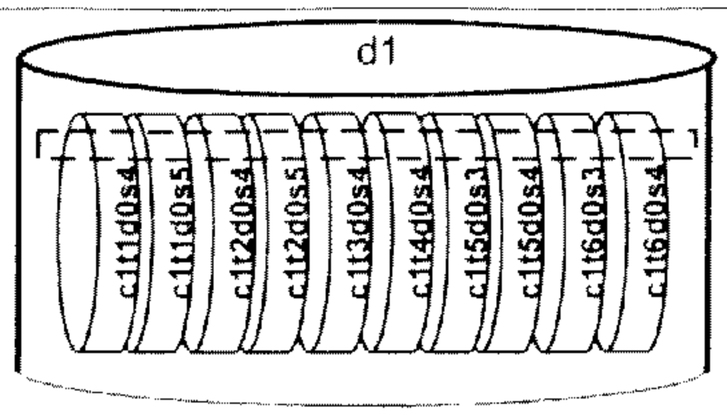


Figura 3.25-2 Metadispositivo d1

Metadispositivo d2. Está formado por las particiones c1t3d0s5 c1t4d0s5 de los discos de la posición 3 y 4 de la parte frontal del arreglo yodo. La distribución de discos que está utilizando es la de striping.

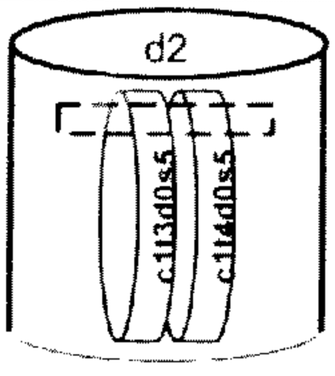


Figura 3.25-3 Metadispositivo d2

Metadispositivo d3. Está formado por las particiones c1t5d0s5 c1t6d0s5 de los discos de la posición 5 y 6 de la parte frontal del arreglo yodo. La distribución de discos que está utilizando es la de striping.

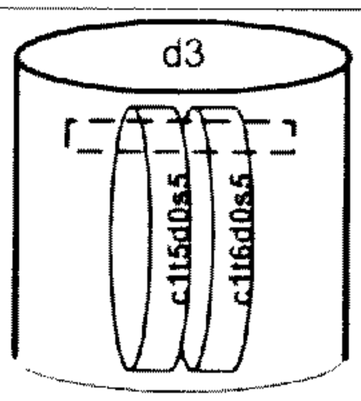


Figura 3.25-4 Metadispositivo d3

Metadispositivo d4. Está formado por las particiones c1t1d0s6 c1t2d0s6 de los discos de la posición 1 y 2 de la parte frontal del arreglo yodo. La distribución de discos que está utilizando es la de striping.

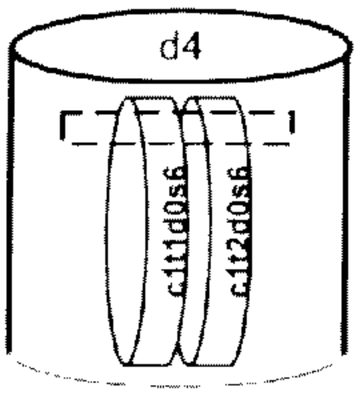


Figura 3.25-5 Metadispositivo d4

Metadispositivo d5. Está formado por las particiones c1t3d0s6 c1t4d0s6 de los discos de la posición 3 y 4 de la parte frontal del arreglo yodo. La distribución de discos que está utilizando es la de striping.

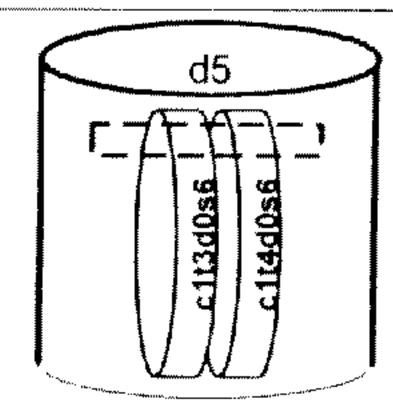


Figura 3.25-6 Metadispositivo d5

Metadispositivo d6. Está formado por las particiones c1t5d0s6 c1t6d0s6 de los discos de la posición 5 y 6 de la parte frontal del arreglo yodo. La distribución de discos que está utilizando es la de striping.

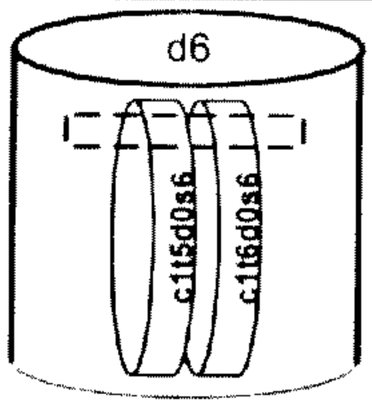


Figura 3.25-7 Metadispositivo d6

Metadispositivo d7. Está formado por las particiones c1t1d0s7 c1t2d0s7 de los discos de la posición 1 y 2 de la parte frontal del arreglo yodo. La distribución de discos que está utilizando es la de striping.

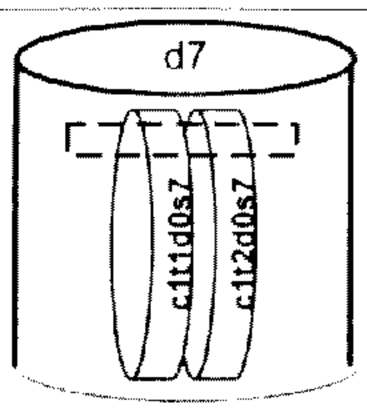


Figura 3.25-8 Metadispositivo d7

Metadispositivo d8. Está formado por las particiones c1t3d0s7 c1t4d0s7 de los discos de la posición 3 y 4 de la parte frontal del arreglo yodo. La distribución de discos que está utilizando es la de striping.

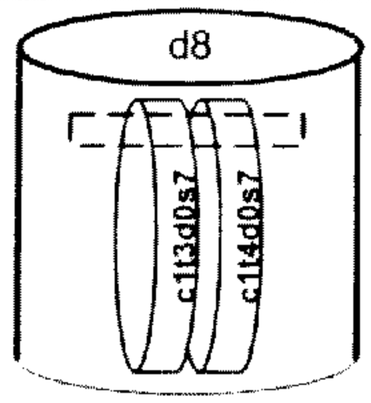


Figura 3.25-9 Metadispositivo d8

Metadispositivo d9. Está formado por las particiones c1t5d0s7 c1t6d0s7 de los discos de la posición 5 y 6 de la parte frontal del arreglo yodo. La distribución de discos que está utilizando es la de striping.

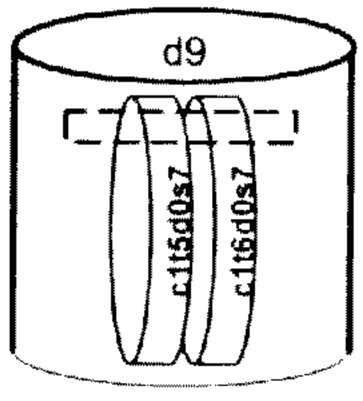


Figura 3.25-10 Metadispositivo d9

Metadispositivo d10. Está formado por la partición c0t10d0s0 del disco c0t10d0 que ocupa la posición 10 del arreglo D1000. La distribución de discos que está utilizando es la de striping.

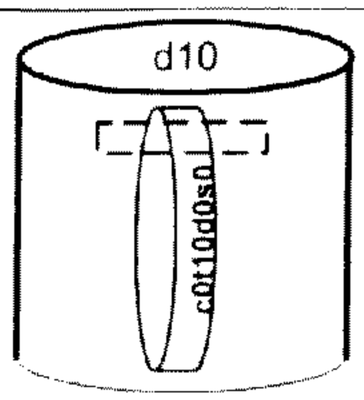


Figura 3.25-11 Metadispositivo d10

Metadispositivo d11. Está formado por las particiones c0t0d0s0 c0t0d0s1 c0t1d0s0 c0t1d0s1 c0t2d0s0 c0t2d0s1 c0t3d0s0 c0t3d0s1 c0t3d0s3 c0t3d0s4 c0t8d0s0 c0t8d0s1 c0t8d0s3 c0t8d0s4 c0t9d0s0 c0t9d0s1 c0t9d0s3 c0t9d0s4 c0t10d0s1 c0t10d0s3 los discos que ocupan de la posición 0, 1,2,3, 8, 9 y 10 del arreglo D1000. La distribución de discos que está utilizando es la de striping.

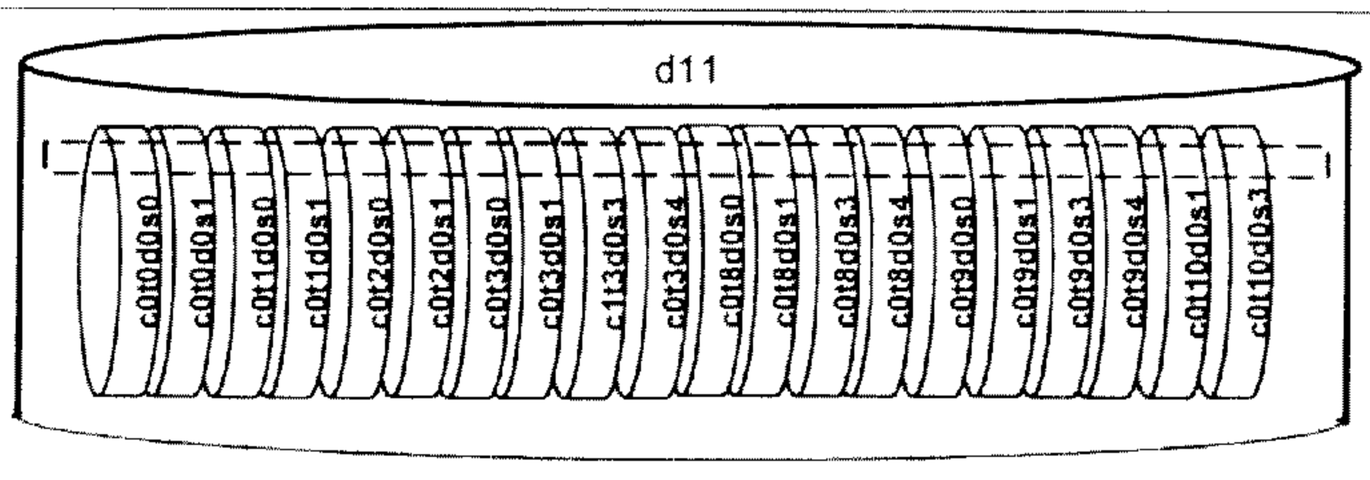


Figura 3.25-12 Metadispositivo d11

Metadispositivo d12. Está formado por las particiones c0t3d0s5 c0t8d0s5 c0t9d0s5 c0t10d0s5 de los discos que ocupan de la posición 3, 8, 9 y 10 del arreglo D1000. La distribución de discos que está utilizando es la de striping.

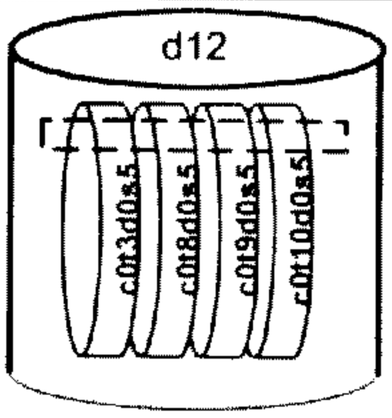


Figura 3.25-13 Metadispositivo d12

Metadispositivo d13. Está formado por las particiones c0t2d0s3 c0t3d0s6 c0t8d0s6 c0t9d0s6 c0t10d0s6 de los discos que ocupan de la posición 2, 3,8, 9 y 10 del arreglo D1000. La distribución de discos que está utilizando es la de striping.

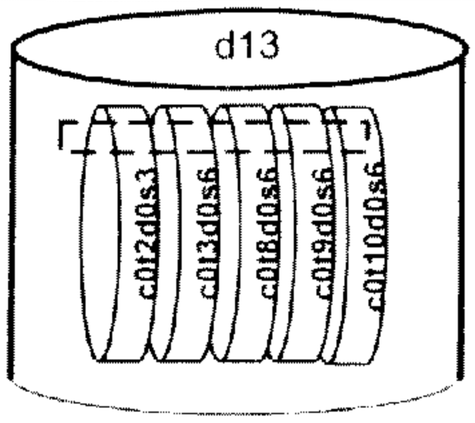


Figura 3.25-14 Metadispositivo d13

Desventajas de la configuración actual

Existen varias desventajas de la configuración de discos actual, la primera de ellas es que el disco que tiene el sistema operativo no tiene redundancia en línea, aunque ya tenemos un disco alterno de inicio que se implementó en el capítulo anterior no se cuenta con ningún tipo de redundancia que permita mantener el sistema funcionando en caso de que ocurra una falla física del disco de sistema operativo. Otra desventaja es que a pesar de que los sistemas de archivos creados con DiskSuite están distribuidos a través de varias particiones de discos diferentes y por lo mismo tienen un buen funcionamiento de lectura y escritura, no están espejeados, es decir si llega a fallar alguno de estos discos no podremos recuperar la información que se pierda manteniendo el sistema en línea y directamente de algún disco que este como espejo, pues no tenemos implementado ningún nivel RAID para protección de los datos.

Nueva configuración de discos

Para eliminar las desventajas de la configuración de discos actual del servidor web, se hicieron una serie de cambios en la estructura física y lógica de dicha configuración. Para ello fue necesario mover de posición algunos discos de los arreglos A5000, la actualización de sistema operativo, la actualización de Solstice DiskSuite y la instalación de VERITAS Volume Manager ya que este último no se encontraba instalado.

Lo primero que se hizo fue reubicar físicamente un disco del arreglo "yodo2" en el arreglo "yodo", el disco que se movió fue el de la posición 5 de la parte trasera de "yodo2" a la posiciones 6 de la parte trasera de "yodo", de manera que en cada caja quedaron 13 discos, la siguiente figura muestra como quedaron distribuidos los discos de los arreglos.

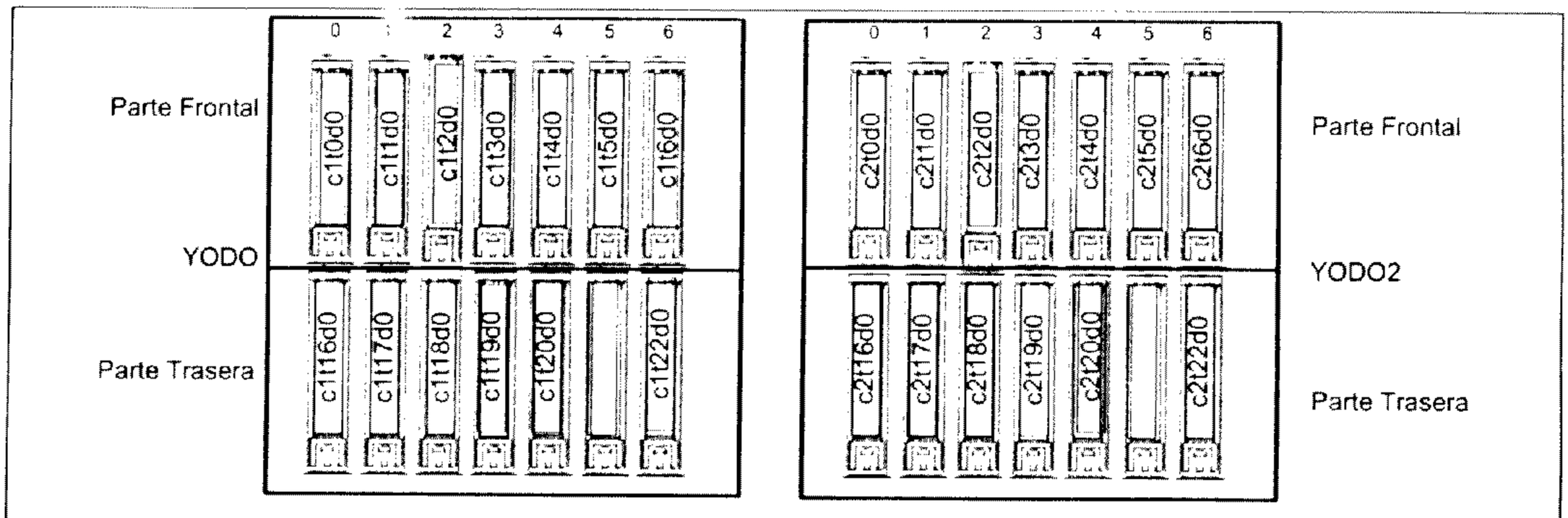


Figura 3.26. Reubicación de discos de arreglos A5000 de dragón

Los discos de yodo se usarán de la manera siguiente: el disco 0 de la parte frontal se va a ocupar para instalar sistema operativo; los discos de las posiciones 1 a la 6 de la parte frontal y de la 1 a la 3 se ocuparán para los datos de usuarios y aplicaciones; el disco de la posición 0 de la parte trasera del arreglo es el disco que se utilizó en el capítulo 2 para configurarlo como disco alterno de sistema operativo; el disco de la posición 4 de la parte trasera se utilizará para la configuración de Volume Manager y en él se creará el sistema de archivos "/respaldos" del que se hace referencia en el capítulo 2; y finalmente el disco de la posición 6 de la parte trasera se configurará como disco de reserva.

Respecto a los discos de yodo2: el disco 0 de la parte frontal se va a ocupar para espejear el disco de sistema operativo; los discos de las posiciones 1 a la 6 de la parte frontal y de la 1 a la 3 se ocuparán para espejear los volúmenes de los datos de usuarios y aplicaciones; el disco de la posición 0 de la parte trasera del arreglo por el momento no lo usaremos hasta que se requiera espacio adicional, por lo tanto lo dejaremos sin asignar; el disco de la posición 4 de la parte trasera se utilizará para espejear el disco que se utilizará para la configuración de Volume Manager y la creación del sistema de archivos "/respaldos"; y por último el disco de la posición 6 de la parte trasera se configurará como disco de reserva.

Lo que se pretende es que el sistema operativo quede espejeado con el manejador de discos Solstice DiskSuite utilizando para ello los discos ya mencionados. También se quiere lograr que los datos de las aplicaciones queden espejeados dentro de volúmenes de VERITAS Volume Manager. Con esto queremos tener una copia tanto de sistema operativo como de datos de aplicaciones y usuarios en los discos del arreglo yodo, y otra copia en los discos del arreglo yodo2, aprovechando que cada arreglo tiene una controladora de discos diferentes.

Manteniendo integra la información

Una de las tareas más críticas antes de hacer la reconfiguración de los discos era mantener integra la información de dragón, sobre todo los datos correspondientes a las aplicaciones y a los usuarios. Para lograr esto se utilizó la máquina SUN Enterprise 3500 donde se manda respaldar a cinta información de dragón y se aprovechó que el arreglo de discos yodo2 no tenía información de ningún tipo; primero se conectó el arreglo yodo2 a la 3500 para que al igual que dragón tuviera acceso a sus discos; una vez que ambas máquinas tuvieron acceso a los discos del arreglo yodo2 necesitábamos que también ambas máquinas tuvieran instalado Volume Manager, afortunadamente la 3500 si lo tenía, pero dragón no por lo cual se instaló. Con los discos de yodo2 se crearon en dragón un grupo de discos llamado tempodg y los volúmenes necesarios con un solo plex para almacenar la información contenida en los metadispositivos que aproximadamente utilizaban 150 GB. Una vez que se crearon los volúmenes se copió toda la información de los metadispositivos hacia ellos. Después se deportó el grupo de discos tempodg de dragón y se importó en la Enterprise 3500. Una vez que se comprobó que los datos estaban íntegros se comenzó con la reestructuración de los discos de dragón.

Es muy importante señalar que en este punto no estamos mostrando los procedimientos que se hicieron, ya que fue algo temporal para llegar a nuestro objetivo final, del cual, sí se describirán procedimientos en los puntos posteriores, por tal motivo solo se menciona lo que hicimos de manera muy general.

Actualización de sistema operativo

La versión del sistema operativo Solaris que se tenía en dragón era la 2.6, la versión a la que se actualizó fue la 2.8, esto se hizo con el propósito de tener una versión mas reciente y que por lo mismo tuviera menos "bugs" o errores de software. El sistema operativo se instaló en el disco de la posición 0 de la parte frontal del arreglo yodo, es decir en el disco c1t0d0. No fue realmente una actualización como tal, sino más bien una instalación desde cero de una versión más reciente de sistema operativo. Es importante mencionar que la instalación del sistema operativo solo se hizo en el disco c1t0d0 del arreglo yodo, no fue necesario hacer una instalación en el disco c2t0d0 del arreglo yodo2, ya que esto se hizo a través del software de DiskSuite.

Los sistemas de archivos de sistema operativo quedaron de la siguiente forma después de la instalación:

```
/dev/dsk/c1t0d0s0    2056211  676656 1317869    34%    /
/dev/dsk/c1t0d0s3    4131866  908055 3182493    23%    /usr
/dev/dsk/c1t0d0s4    4131866 1337807 2752741    33%    /var
/dev/dsk/c1t0d0s5    4131866  413741 3676807    11%    /opt
swap                 3034584     152 3034432     1%    /tmp
```

En este punto se hizo un respaldo de sistema operativo a cinta, por si en determinado momento era requerido.

Nuevamente aclaramos la descripción del proceso de instalación de sistema operativo no está dentro del objetivo de este trabajo, así que solo lo mencionamos de manera general

Instalación y configuración de Solstice DiskSuite

Para la instalación y configuración de Solstice DiskSuite versión 4.2.1 se ocuparon los discos c1t0d0 del arreglo yodo en el que se instaló el sistema operativo, y el c2t0d0 del arreglo yodo2. Se utilizó el siguiente procedimiento.

El software se encuentra en el CD 2/2 de Solaris 8 en la ruta siguiente:

```
<CD2>/Solaris_8/EA/products/DiskSuite_4.2.1/sparc/Packages
```

1. Instalar todos los paquetes de DiskSuite versión 4.2.1.

```
dragon # pwd
dragon # <CD2>/Solaris_8/EA/Products/DiskSuite_4.2.1/sparc/Packages
dragon # pkgadd -d .
```

2. Terminada la instalación se instaló el parche recomendado número 108693-07.

```
dragon # patchadd 108693-07
```

3. Crear las particiones en el disco espejo de sistema operativo para los metadispositivos de DiskSuite.

3a. Crear una réplica de la tabla de particiones del sistema operativo del disco c1t0d0 que contiene el sistema operativo, al disco que será el espejo c2t0d0.

```
dragon # prtvtoc /dev/rdsk/c1t0d0s0 | fmthard -s - /dev/rdsk/c2t0d0s0 →sistema de archivos  /
dragon # prtvtoc /dev/rdsk/c1t0d0s1 | fmthard -s - /dev/rdsk/c2t0d0s1 →sistema de archivos  swap
dragon # prtvtoc /dev/rdsk/c1t0d0s3 | fmthard -s - /dev/rdsk/c2t0d0s3 →sistema de archivos  /usr
dragon # prtvtoc /dev/rdsk/c1t0d0s4 | fmthard -s - /dev/rdsk/c2t0d0s4 →sistema de archivos  /var
dragon # prtvtoc /dev/rdsk/c1t0d0s5 | fmthard -s - /dev/rdsk/c2t0d0s5 →sistema de archivos  /opt
dragon # prtvtoc /dev/rdsk/c1t0d0s7 | fmthard -s - /dev/rdsk/c2t0d0s7 → réplicas de DiskSuite

fmthard:  New volume table of contents now in place.
```

Con esto el disco c2t0d0 está dividido en las mismas particiones y del mismo tamaño que el disco c1t0d0 donde se instaló sistema operativo.

3b. Ejecutar el script hecho en shell de unix para crear las réplicas de las bases de datos de configuración propias de DiskSuite y los metadispositivos. Se explicará línea por línea este script.

```
dragon # cd /usr/sbin

### De las siguientes dos líneas la primera crea tres réplicas en la partición  c1t0d0s7, y la
### segunda  crea otras tres réplicas en la partición c2t0d0s7.

metadb -a -f -c 3 c1t0d0s7
metadb -a -f -c 3 c2t0d0s7

### De las siguientes tres líneas la primera indica la creación de un subespejo llamado d10 con
la partición c1t0d0s0,  ###  la segunda línea indica la creación de un segundo subespejo llamado
d20 con la partición c2t0d0s0, y la tercer ### línea inicializa el metaespejo d0 utilizando el
subespejo d10 que corresponde al sistema de archivos / de sistema
### operativo.
```

```
metainit -f d10 1 1 c1t0d0s0
metainit -f d20 1 1 c2t0d0s0
metainit d0 -m d10

### De las siguientes tres lineas la primera indica la creación de un subespejo llamado d11 con
la partición c1t0d0s1,  ### la segunda línea indica la creación de un segundo subespejo llamado
d21 con la partición c2t0d0s1, y la tercer ### línea inicializa el metaespejo d1 utilizando el
subespejo d11 que corresponde al swap del sistema
### operativo.

metainit -f d11 1 1 c1t0d0s1
metainit -f d21 1 1 c2t0d0s1
metainit d1 -m d11

### De las siguientes tres líneas la primera indica la creación de un subespejo llamado d12 con
la partición c1t0d0s3,  ### la segunda línea indica la creación de un segundo subespejo llamado
d22 con la partición c2t0d0s3, y la tercer ### línea inicializa el metaespejo d2 utilizando el
subespejo d12 que corresponde al sistema de archivos /usr de
### sistema operativo.

metainit -f d12 1 1 c1t0d0s3
metainit -f d22 1 1 c2t0d0s3
metainit d2 -m d12
```

De las siguientes tres líneas la primera indica la creación de un subespejo llamado d13 con la partición c1t0d0s4, ### la segunda línea indica la creación de un segundo subespejo llamado d23 con la partición c2t0d0s4, y la tercer ### línea inicializa el metaespejo d3 utilizando el subespejo d13 que corresponde al sistema de archivos /var de
sistema operativo.

```
metainit -f d13 1 1 c1t0d0s4
metainit -f d23 1 1 c2t0d0s4
metainit d3 -m d13

### De las siguientes tres líneas la primera indica la creación de un subespejo llamado d14 con
la partición c1t0d0s5,  ### la segunda línea indica la creación de un segundo subespejo llamado
d24 con la partición c2t0d0s5, y la tercer  ### línea inicializa el metaespejo d4 utilizando el
subespejo d14 que corresponde al sistema de archivos /opt de
### sistema operativo.

metainit -f d14 1 1 c1t0d0s5
metainit -f d24 1 1 c2t0d0s5
metainit d4 -m d14

### La partición donde están guardadas las copias de las réplicas de DiskSuite no se espejea.
### Con el siguiente comando se modifica el archivo de configuración /etc/vfstab para que la
siguiente vez el
### servidor inicalize de metadispositivos de DiskSuite y no de particiones de disco simples.

metaroot d0
```

3c. Verificar que los metadispositivos y las réplicas de DiskSuite fueron creados correctamente

```
### Debemos tener 3 réplicas en la partición c1t0d0s7 y 3 mas en la partición c2t0d0s7, ya que
fueron las que
### debieron de haberse creado con la segunda línea del script anterior, usemos el siguiente
comando.
```

```
dragon # metadb -i
        flags           first blk       block count
     a m  p  luo        16              1034            /dev/dsk/c1t0d0s7
     a    p  luo        1050            1034            /dev/dsk/c1t0d0s7
     a    p  luo        2084            1034            /dev/dsk/c1t0d0s7
     a    p  luo        16              1034              /dev/dsk/c2t0d0s7
     a    p  luo        1050            1034            /dev/dsk/c2t0d0s7
     a    p  luo        2084            1034            /dev/dsk/c2t0d0s7
 o - replica active prior to last mddb configuration change
 u - replica is up to date
 l - locator for this replica was read successfully
 c - replica's location was in /etc/lvm/mddb.cf
 p - replica's location was patched in kernel
 m - replica is master, this is replica selected as input
 W - replica has device write errors
 a - replica is active, commits are occurring to this replica
 M - replica had problem with master blocks
 D - replica had problem with data blocks
 F - replica had format problems
 S - replica is too small to hold current data base
 R - replica had device read errors


# Para comprobar que los metaespejos se hayan hecho correctamente usemos el comando siguiente:

dragon # metastat -p
d0 -m d10 d20 1       ### Indica que el metaespejo d0 está formado por los subespejos d10 y d20
d10 1 1 c1t0d0s0      ### Indica que el suespejo d10 está formado por  c1t0d0s0 y pertenece a d0
d1 -m d11 d21 1       ### Indica que el metaespejo d1 está formado por los subespejos d11 y d21
d11 1 1 c1t0d0s1      ### Indica que el suespejo d11 está formado por  c1t0d0s1 y pertenece a d1
d2 -m d12 d22 1       ### Indica que el metaespejo d2 está formado por los subespejos d12 y d22
d12 1 1 c1t0d0s3      ### Indica que el suespejo d12 está formado por  c1t0d0s3 y pertenece a d2
d3 -m d13 d23 1       ### Indica que el metaespejo d3 está formado por los subespejos d13 y d23
d13 1 1 c1t0d0s4      ### Indica que el suespejo d13 está formado por  c1t0d0s4 y pertenece a d3
d4 -m d14 d24 1       ### Indica que el metaespejo d4 está formado por los subespejos d14 y d24
d14 1 1 c1t0d0s5      ### Indica que el suespejo d14 está formado por  c1t0d0s5 y pertenece a d4

d20 1 1 c2t0d0s0     ### Las 4 líneas de la izquierda muestran los subespejos que no están
sincronizados
d22 1 1 c2t0d0s3     ### con su respectivo metaespejo.
d23 1 1 c2t0d0s4
d24 1 1 c2t0d0s5
```

4. Editar el archivo /etc/vfstab para incluir las rutas de los metadispositivos y comentar las rutas de los dispositivos fisicos.

```
### Se recomienda respaldar este archivo y editar el archivo original, dando de alta los
metadispositivos. Al finalizar ### el archivo deberá de quedar de la forma siguiente:

* Las letras en negritas son las líneas que se agregan al editar el vfstab

#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
#/dev/md/dsk/d0 /dev/md/rdsk/d0 /         ufs     1       yes     - ### Esta linea ya se habia
agregado
fd      -       /dev/fd fd      -       no      -
```

```
/proc    -          /proc    proc     -        no       -
/dev/md/dsk/d0   /dev/md/rdsk/d0 /        ufs      1        no       -
swap     -          /tmp     tmpfs    -        yes      -
/dev/md/dsk/d1   -          -        swap     -        no       -
/dev/md/dsk/d2   /dev/md/rdsk/d2 /usr     ufs      1        no       -
/dev/md/dsk/d3   /dev/md/rdsk/d3 /var     ufs      1        no       -
/dev/md/dsk/d4   /dev/md/rdsk/d4 /opt     ufs      2        yes      -

### Este procedimiento requiere que se reinicialize el sistema.

dragon # sync         ### Sincroniza los datos de memoria a disco
dragon # sync         ### Sincroniza los datos de memoria a disco
dragon # init 6        ### Reinicializa el sistema
```

5. Por último, una vez que el sistema está funcionando nuevamente, es necesario asociar los subespejos que están desasociados a sus metaespejos correspondientes para que la información se comience a sincronizar. Para ello se emplea el script siguiente.

```
### Con las siguientes líneas de comandos lo que estamos haciendo es asociar cada subespejo que
estaba
### desasociado  con su subespejo y metaespejo correspondiente para  iniciar la sincronización de
sus datos.

metattach d0 d20
metattach d1 d21
metattach d2 d22
metattach d3 d23
metattach d4 d24

### Verificar el avance de la sincronicación

# metastat |grep -i sync

### Cuando la sincronozación haya terminado el proceso de instalación y configuración de Solstice
DiskSuite estará
### concluido.
```

De esta manera ya tenemos dos discos con el mismo sistema operativo, uno en la posición 0 del arreglo yodo y el otro en la posición 0 del arreglo yodo2. La gráfica siguiente muestra como quedaron los metaespejos del sistema operativo del servidor web de la UNAM.

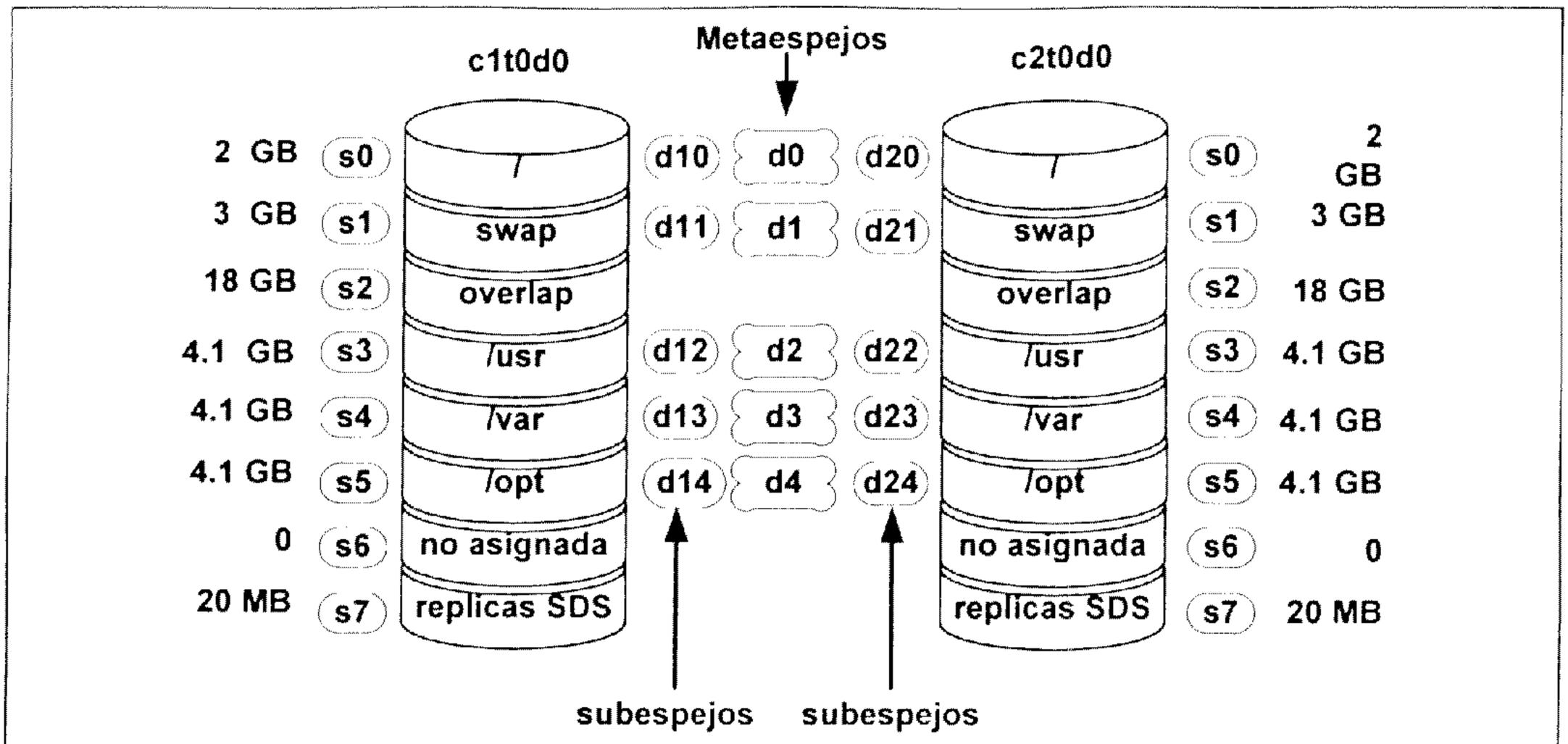


Figura 3.27. Metaespejos de disk suite de sistema operativo de dragon

Instalación y configuración de VERITAS Volume Manager y VERITAS File System

El siguiente paso es la instalación y configuración de VERITAS Volume Manager. El objetivo de esto es que para los discos de datos de las aplicaciones y usuarios tengamos una configuración RAID 0+1. Es decir tendremos volúmenes que tendrán un primer plex con distribución de striping, y un segundo plex con la misma distribución y que al mismo tiempo será espejo del primero.

Antes que nada haremos otro respaldo a cinta del sistema operativo del disco del arreglo yodo, ya teníamos un respaldo pero este no incluye todo lo que se hizo a partir de la instalación de DiskSuite, por lo tanto haremos otro. También desasociaremos los subespejos d20, d21, d22,d23 y d24 de sus respectivos metaespejos para que los nuevos cambios de configuración solo se hagan en los subespejos d10,d11,d12,d13 y d14, y de esta manera tener un segundo respaldo de sistema operativo en disco como medida de contingencia.

Los discos que utilizaremos para Volume Manager son los de la posición 1, 2, 3, 4, 5 y 6 de la parte frontal tanto de yodo como de yodo2, y los de las posiciones 1, 2, 3 y 4 de la parte trasera también de ambos arreglos de discos. Primero usaremos yodo para crear los volumenes con los plexes principales y luego yodo2 para crear los espejos.

Sigamos el siguiente procedimiento para instalar y configurar VERITAS Volume Manager versión 3.0.4., este procedimiento incluye también la instalación de VERITAS File Systems

aunque respecto a este último no hay que haecer grandes cambios, solo hay que instalar su paquete correspondiente y comenzar a usarlo para crear sistemas de archivos.

1.- Instalación de los paquetes de VERITAS. Una vez descomprimido el directorio que contiene los paquetes de VERITAS, pasarse a él e instalarlos, estos paquetes contienes los archivos binarios, las páginas de manual, documentación y ambiente gráfico.

```
dragon # pkgadd -d .

### Después de ejecutar el comando anterior nos mostrará los paquetes de software disponibles para instalarse, le
### indicaremos que se instalen los paquetes 6 2 1 7 5 4 5 3 en ese mismo orden, es importante señalar que el paquete ### 7 corresponde a
VERITAS File System. Antes de instalar cada paquete nos aparecerán algunas preguntas como ### son: si queremos  seguir instalando estos
paquetes, o si se requiere la instalación de algún paquete adicional o un
### parche. A continuación se muestra la mayoría del proceso de instalación.

The following packages are available:

  1  VRTSfsdoc      VERITAS File System Documentation Package
                    (SPARC) 3.3.3 GA Release

  2  VRTSvmdev      VERITAS Volume Manager, Header and Library Files
                    (sparc) 3.0.4,REV=04.18.2000.10.00

  3  VRTSvmdoc      VERITAS Volume Manager (user documentation)
                    (sparc) 3.0.4,REV=04.18.2000.10.00

  4  VRTSvmman      VERITAS Volume Manager, Manual Pages
                    (sparc) 3.0.4,REV=04.18.2000.10.00

  5  VRTSvmsa       VERITAS Volume Manager Storage Administrator
                    (sparc) 3.0.4,REV=04.03.2000.14.30

  6  VRTSvxfs       VERITAS File System
                    (sparc) 3.3.3,REV=GA03

  7  VRTSvxvm       VERITAS Volume Manager, Binaries
                    (sparc) 3.0.4,REV=04.18.2000.10.00

Select package(s) you wish to process (or 'all' to process

all packages). (default: all) [?,??,q]: 7 2 1 8 6 4 5 3


Processing package instance <VRTSvxfs> from </var/tmp/SOFTWARE>

VERITAS File System
(sparc) 3.3.3,REV=GA03
Copyright (c) 1991 - 1999 VERITAS SOFTWARE CORP.  ALL RIGHTS RESERVED.
THIS SOFTWARE IS THE PROPERTY OF AND IS LICENSED BY VERITAS SOFTWARE,
AND/OR ITS SUPPLIERS.

Using </> as the package base directory.
## Processing package information.
## Processing system information.
   19 package pathnames are already properly installed.
## Verifying disk space requirements.
```

```
## Checking for conflicts with packages already installed.
The following files are already installed on the system and are being
used by another package:

  /usr/sbin/vxlicense

Do you want to install these conflicting files [y,n,?,q] y
```

Checking for setuid/setgid programs.

```
The following files are being installed with setuid and/or setgid
permissions:
  /usr/lib/fs/vxfs/vxdump <setuid root setgid tty>
  /usr/lib/fs/vxfs/vxquota <setuid root>
  /usr/lib/fs/vxfs/vxrestore <setuid root setgid bin>

Do you want to install these as setuid/setgid files [y,n,?,q] y

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <VRTSvxfs> [y,n,?] y

Installing VERITAS File System as <VRTSvxfs>

## Installing part 1 of 1.
/dev/vxportal <symbolic link>
/usr/sbin/vxdump <symbolic link>
/usr/sbin/vxedquota <symbolic link>
/usr/sbin/vxquota <symbolic link>
/usr/sbin/vxquotaoff <symbolic link>
/usr/sbin/vxquotaon <symbolic link>
/usr/sbin/vxrepquota <symbolic link>
/usr/sbin/vxrestore <symbolic link>
/usr/sbin/vxtunefs <symbolic link>
/usr/sbin/vxupgrade <symbolic link>
[ verifying class <all> ]
/usr/share/man/man1/cp_vxfs.1
/usr/share/man/man1/cpio_vxfs.1
/usr/share/man/man1/getext.1
/usr/share/man/man1/ls_vxfs.1
/usr/share/man/man1/mv_vxfs.1
.
.
.
/usr/lib/libvxckpt.a
/usr/lib/libxdsm.a
/usr/sbin/vxlicense
[ verifying class <s58> ]
/usr/lib/fs/vxfs/bin/cp <linked pathname>
/usr/lib/fs/vxfs/bin/ln <linked pathname>
/usr/lib/fs/vxfs/vxquotaoff <linked pathname>
/kernel/drv/sparcv9/vxportal
/kernel/fs/sparcv9/vxfs
[ verifying class <s58b64> ]
## Executing postinstall script.

You must install a license key before using VxFS.

Send the machine type and the hostid of your machine to
VERITAS Customer Support to get a license key.

You can obtain the machine type and hostid of your system
by running the commands 'uname -i' and 'hostid', respectively.
```

```
To install the license after obtaining a license key,
run the following command:

  vxlicense -c

VRTSvxfs: If you have finished installing all VRTS packages
VRTSvxfs: reboot the system now.


Installation of <VRTSvxfs> was successful.


1 Processing package instance <VRTSfsdoc> from </var/tmp/SOFTWARE>

VERITAS File System Documentation Package
(SPARC) 3.3.2 GA Release
VERITAS Software

This appears to be an attempt to install the same architecture and
version of a package which is already installed.  This installation
will attempt to overwrite this package.

  1  PostScript
  2  PDF

Select the document formats to be installed (default: all) [?,??,q]: 2
[PDF] will be installed.
## Processing package information.
## Processing system information.
   8 package pathnames are already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing VERITAS File System Documentation Package as <VRTSfsdoc>

## Installing part 1 of 1.
[ verifying class <none> ]
[ verifying class <PDF> ]

Installation of <VRTSfsdoc> was successful.

Processing package instance <VRTSvxvm> from </var/tmp/SOFTWARE>

VERITAS Volume Manager, Binaries
(sparc) 3.0.4,REV=04.18.2000.10.00
Copyright (c) 1990-2000 VERITAS Software Corporation.
ALL RIGHTS RESERVED.
THIS SOFTWARE IS THE PROPERTY OF AND IS LICENSED BY VERITAS SOFTWARE,
AND/OR ITS SUPPLIERS.

        This package, VxVM 3.0.4, is supported on Solaris 2.5.1, 2.6,
        7, and 8.  You appear to be running Solaris 8.  Press
        ENTER to install VxVM 3.0.4 for Solaris 8, or enter
        another Solaris version number if you are certain that you
        want to install the drivers for a different release of
        Solaris.

        Install for which version of Solaris?
        [8, 7, 2.6, 2.5.1] (default: 8):
Installing VxVM for Solaris 8

  The following Sun patch(s) are required for Solaris 8.
  Sun patch(s):
```

```
Continue installation? [y,n,q,?] (default: n): y

Using </> as the package base directory.
## Processing package information.
## Processing system information.
   12 package pathnames are already properly installed.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

The following files are being installed with setuid and/or setgid
permissions:
  /usr/sbin/vxprint <setuid root>

Do you want to install these as setuid/setgid files [y,n,?,q] y

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <VRTSvxvm> [y,n,?] y

Installing VERITAS Volume Manager, Binaries as <VRTSvxvm>

## Executing preinstall script.
## Installing part 1 of 1.
/etc/init.d/vxvm-reconfig
/etc/init.d/vxvm-recover
/etc/init.d/vxvm-startup1
/etc/init.d/vxvm-startup2
/etc/init.d/vxvm-sysboot
/etc/vx/bin <symbolic link>
.
.
.
/usr/sbin/vxrecover.wrap
/usr/sbin/vxrelayout
/usr/sbin/vxsd
/usr/sbin/vxstat
/usr/sbin/vxtask
/usr/sbin/vxtrace
/usr/sbin/vxvol
[ verifying class <none> ]
/etc/rc2.d/S95vxvm-recover <linked pathname>
/etc/rcS.d/S25vxvm-sysboot <linked pathname>
/etc/rcS.d/S35vxvm-startup1 <linked pathname>
/etc/rcS.d/S85vxvm-startup2 <linked pathname>
/etc/rcS.d/S86vxvm-reconfig <linked pathname>
/usr/sbin/vxserial <linked pathname>
## Executing postinstall script.

Copy //kernel/drv/vxio.SunOS_5.8 to //kernel/drv/vxio...
Copy //kernel/drv/vxspec.SunOS_5.8 to //kernel/drv/vxspec...
Copy //kernel/drv/sparcv9/vxio.SunOS_5.8 to //kernel/drv/sparcv9/vxio...
Copy //kernel/drv/sparcv9/vxspec.SunOS_5.8 to //kernel/drv/sparcv9/vxspec...
Copy //sbin/vxconfigd.SunOS_5.8 to //sbin/vxconfigd...
Copy //kernel/drv/vxdmp.SunOS_5.8 to //kernel/drv/vxdmp...
Copy //kernel/drv/sparcv9/vxdmp.SunOS_5.8 to //kernel/drv/sparcv9/vxdmp...

Adding vxdmp driver for SunOS version 5.8...
Adding vxio driver for SunOS version 5.8...
Adding vxspec driver for SunOS version 5.8...
Adding vxspec lines to //etc/devlink.tab...
Adding vxdmp lines to //etc/devlink.tab...
Running /usr/sbin/devlinks -r / -t //etc/devlink.tab ...
```

```
Adding vxio vxspec vxdmp lines to //etc/system...
Copy libthread.so.1 to //etc/vx...
Copy libc.so.1 to //etc/vx...

Installation of <VRTSvxvm> was successful.

Processing package instance <VRTSvmsa> from </var/tmp/SOFTWARE>

VERITAS Volume Manager Storage Administrator
(sparc) 3.0.6,REV=04.03.2000.14.30
Copyright (c) 2000 VERITAS Software Corporation.
ALL RIGHTS RESERVED.
THIS SOFTWARE IS THE PROPERTY OF AND IS LICENSED BY VERITAS SOFTWARE,
AND/OR ITS SUPPLIERS.

Where should this package be installed? (default: /opt]) [?,q]

Should the Apache HTTPD (Web Server) included in this package be installed?
(default: n) [y,n,?,q] n

Should the Volume Manager Storage Administrator Server be installed on this
system?
(The Volume Manager Storage Administrator Client will be installed regardless)
(default: y) [y,n,?,q] y

Using </opt> as the package base directory.
## Processing package information.
## Processing system information.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <VRTSvmsa> [y,n,?] y

Installing VERITAS Volume Manager Storage Administrator as <VRTSvmsa>

## Installing part 1 of 1.
/opt/VRTSvmsa/jre/CHANGES
/opt/VRTSvmsa/jre/COPYRIGHT
/opt/VRTSvmsa/jre/LICENSE
/opt/VRTSvmsa/jre/README
/opt/VRTSvmsa/jre/bin/.java_wrapper
/opt/VRTSvmsa/jre/bin/java
/opt/VRTSvmsa/jre/bin/javakey
/opt/VRTSvmsa/jre/bin/jre
.
.
.
/opt/VRTSvmsa/vmsa/vmsa2.html
/opt/VRTSvmsa/vmsa/vmsa3.html
/opt/VRTSvmsa/vmsa/vmsa3_blank.html
/opt/VRTSvmsa/vmsa/vmsa3_intro.html
/opt/VRTSvmsa/vmsa/vmsa4.html
/opt/VRTSvmsa/vmsa/vmsaz.html
[ verifying class <client> ]
## Executing postinstall script.

Installation of <VRTSvmsa> was successful.

Processing package instance <VRTSvmdoc> from </var/tmp/SOFTWARE>
```

```
VERITAS Volume Manager (user documentation)
(sparc) 3.0.4,REV=04.18.2000.10.00
Copyright (c) 2000 VERITAS Software Corporation.
ALL RIGHTS RESERVED.
THIS SOFTWARE IS THE PROPERTY OF AND IS LICENSED BY VERITAS SOFTWARE,
AND/OR ITS SUPPLIERS.

  1  PostScript
  2  PDF

Select the document formats to be installed (default: all) [?,??,q]: 2
[PDF] will be installed.
Using </opt> as the package base directory.
## Processing package information.
## Processing system information.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing VERITAS Volume Manager (user documentation) as <VRTSvmdoc>

## Installing part 1 of 1.
/opt/VRTSvxvm/docs/cli.pdf
/opt/VRTSvxvm/docs/gsg.pdf
/opt/VRTSvxvm/docs/install.pdf
/opt/VRTSvxvm/docs/ref.pdf
/opt/VRTSvxvm/docs/vmsaguide.pdf
[ verifying class <PDF> ]

Installation of <VRTSvmdoc> was successful.

Processing package instance <VRTSvmman> from </var/tmp/SOFTWARE>

VERITAS Volume Manager, Manual Pages
(sparc) 3.0.4,REV=04.18.2000.10.00
Copyright (c) 1990-2000 VERITAS Software Corporation.
ALL RIGHTS RESERVED.
THIS SOFTWARE IS THE PROPERTY OF AND IS LICENSED BY VERITAS SOFTWARE,
AND/OR ITS SUPPLIERS.
Using </opt> as the package base directory.
## Processing package information.
## Processing system information.
   1 package pathname is already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing VERITAS Volume Manager, Manual Pages as <VRTSvmman>

## Installing part 1 of 1.
/opt/VRTSvxvm/man/man1m/vxapslice.1m
/opt/VRTSvxvm/man/man1m/vxassist.1m
/opt/VRTSvxvm/man/man1m/vxbootsetup.1m
/opt/VRTSvxvm/man/man1m/vxconfigd.1m
/opt/VRTSvxvm/man/man1m/vxdctl.1m
/opt/VRTSvxvm/man/man1m/vxdg.1m
.
.
.
/opt/VRTSvxvm/man/man7/vxconfig.7
/opt/VRTSvxvm/man/man7/vxdmp.7
/opt/VRTSvxvm/man/man7/vxinfo.7
/opt/VRTSvxvm/man/man7/vxio.7
/opt/VRTSvxvm/man/man7/vxiod.7
/opt/VRTSvxvm/man/man7/vxtrace.7
```

```
[ verifying class <none> ]

Installation of <VRTSvmman> was successful.

Processing package instance <VRTSvmdev> from </var/tmp/SOFTWARE>

VERITAS Volume Manager, Header and Library Files
(sparc) 3.0.4,REV=04.18.2000.10.00
Copyright (c) 1990-2000 VERITAS Software Corporation.
ALL RIGHTS RESERVED.
THIS SOFTWARE IS THE PROPERTY OF AND IS LICENSED BY VERITAS SOFTWARE,
AND/OR ITS SUPPLIERS.
Using </opt> as the package base directory.
## Processing package information.
## Processing system information.

   1 package pathname is already properly installed.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

Installing VERITAS Volume Manager, Header and Library Files as <VRTSvmdev>

## Installing part 1 of 1.
/opt/VRTSvxvm/include/common.h
/opt/VRTSvxvm/include/dmpapi.h
/opt/VRTSvxvm/include/gencommon.h
/opt/VRTSvxvm/include/libcmd_sys.h
/opt/VRTSvxvm/include/libvxvmsc.h
/opt/VRTSvxvm/include/sliced.h
.
.
.
/opt/VRTSvxvm/include/vxvm/volstats.h
/opt/VRTSvxvm/include/vxvm/volstats_cvm.h
/opt/VRTSvxvm/include/vxvm/voltrace.h
/opt/VRTSvxvm/include/vxvm/vxio.h
/opt/VRTSvxvm/include/vxvm/vxtypes.h
/opt/VRTSvxvm/lib/libvxvm.a
/opt/VRTSvxvm/lib/libvxvmsc.so
[ verifying class <none> ]

Installation of <VRTSvmdev> was successful.

### Una vez que ha terminado la instalación aparece nuevamente la lista de paquetes disponibles
para instalar,
### debemos dar " q " para salir de este menú.

The following packages are available:

  1  VRTSfsdoc      VERITAS File System Documentation Package
                    (SPARC) 3.3.2 GA Release


  2  VRTSvmdev      VERITAS Volume Manager, Header and Library Files
                    (sparc) 3.0.4,REV=04.18.2000.10.00

  3  VRTSvmdoc      VERITAS Volume Manager (user documentation)
                    (sparc) 3.0.4,REV=04.18.2000.10.00

  4  VRTSvmman      VERITAS Volume Manager, Manual Pages
                    (sparc) 3.0.4,REV=04.18.2000.10.00

  5  VRTSvmsa       VERITAS Volume Manager Storage Administrator
```

```
                    (sparc) 3.0.6,REV=04.03.2000.14.30

  6  VRTSvxfs       VERITAS File System
                    (sparc) 3.3.3,REV=GA03

  7  VRTSvxvm       VERITAS Volume Manager, Binaries
                    (sparc) 3.0.4,REV=04.18.2000.10.00


Select package(s) you wish to process (or 'all' to process

all packages). (default: all) [?,??,q]: q
```

2.- Instalar el parche número 109299-02 para VERITAS File System (VxFS) y el 110263-05 para VERITAS Volume Manager (VxVM), después de eso reinicializar el servidor.

```
dragon # patchadd  109299-02

Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...

Patch number 109299-02  has been successfully installed.
See /var/sadm/patch/109299-02/log for details

dragon # patchadd  110263-05

Checking installed patches...
Verifying sufficient filesystem capacity (dry run method)...
Installing patch packages...

Patch number 110263-05 has been successfully installed.
See /var/sadm/patch/110263-05/log for details

### Después de este procedimiento se requiere reinicializar el servidor para que tengan efecto
los cambios.

dragón # init 6
```

3.- Una vez que el servidor está funcionando nuevamente se procede con la configuración de Volume Manager.

```
### Inicializar el proceso o de configuración de Volume Manager  vxconfigd, el cual mantiene
configuraciones de
### discos de volume manger y de sus grupos de discos.

dragon # vxconfigd

###  Crear el archivo /etc/vx/volboot, este archivo contiene un identificador del servidor que es
utilizado por volume ### manager  para establecer a quién pertencen los discos físicos.

dragon # vxdctl init

### Agregar licencias de Volume Manager y file system. El agregar una licencia de VERITAS implica
introducir un
```

```
### número de varias cifras lo cual nos va a permitir hacer uso de este software de manera legal.
En este caso por
### razones de seguridad este número de licencia lo denotaremos con "#".

### Para VERITAS Volume Manager

dragon # vxlicense -c
...
...
#### #### #### #### #### ####
...
### Para VERITAS File System

dragon # vxlicense -c
...
...
#### #### #### #### #### ####
...

###  Crear el grupo de discos rootdg. Antes de crear cualquier grupo de discos primero debe
inicializarse el grupo ### rootdg  ya que es indispensable para la configuración de Volume
Manager.

# vxdg init rootdg

### Inicializar el disco rootdg01 y agregarlo al grupo rootdg, en este caso el disco que usaremos
para el grupo rootdg ### es el c1t20d0 del arreglo de discos yodo.

dragon # vxdisksetup -i c1t20d0
dragon # vxdisk -f init c1t20d0s2
dragon # vxdg -g rootdg adddisk rootdg01=c1t20d0s2

### Habilitar el manejador de discos Volume Manager

dragon # vxdctl enable

### Remover archivo install-db. Este archivo se crea automáticamente cuando se instala Volume
Manager, y al estar ### creado lo que hace es evitar que Volume Manager inicialize cuando el
servidor inicia.

dragon # cd /etc/vx/reconfig.d/state.d
dragon # rm install-db
```

Inicializar y agregar el disco espejo del disco rootdg01 que se configuró dos pasos atrás. El disco que utilizaremos como espejo es el c2t20d0 del arreglo yodo2.

```
dragon # vxdisksetup -i c2t20d0
dragon # vxdisk -f init c2t20d0s2
dragon # vxdg -g rootdg adddisk rootdg02=c2t20d0s2

#### Se reinicializa el servidor dragón  para verificar que Volume Manager funcione de manera
#### automática y de forma adecuada.
```

4.- Creación de volúmenes y sistemas de archivos del grupo rootdg. El único sistema de archivos que vamos a tener sobre el grupo rootdg es el de /respaldos que nos servirá para guardar respaldos de sistema operativo, como recordarán la creación de este sistema de archivos quedó pendiente en el capitulo 2 precisamente por la reestructuración que se iba a hacer en el capítulo actual.

Primero crearemos el volumen con un plex sobre el disco rootdg01 y luego el segundo plex o espejo sobre el disco rootdg02.

```
### Crear el volumen respaldos y su sistema de archivos; crear punto de montura y montarlo.

dragon # vxassist -g rootdg make repaldos 17g rootdg01
dragon # mkfs -F vxfs -o  largefiles,bsize=8192 /dev/vx/rsdk/rootdg/respaldos
dragon # fsck -F vxfs /dev/vx/rdsk/rootdg/respaldos
dragon # mkdir /respaldos
dragon # mount -F vxfs /dev/vx/dsk/rootdg/respaldos /respaldos

### Crear el espejo del volumen  respaldos. Al crear el espejo con el comando siguiente, lo que se hizo fue empezar ### la sincronización del
plex del disco rootdg01 con el plex del disco rootdg02, de tal forma al término de ésta
### sincronización existían un volumen con dos plexes, un plex en cada disco.

dragon # vxassist -g rootdg mirror respaldos rootdg02
```

5.- Configuración del grupo de discos para datos de aplicaciones y usuarios.

Hasta este momento sólo hemos configurado el grupo de rootdg el cual sabemos que es obligatorio en la configuración de Volume Manager, y hemos aprovechado los discos que usa para crear un volumen espejeado para guardar respaldos. Pero no hemos realizado nada acerca de los volúmenes que tendrán los datos de las aplicaciones y usuarios. Básicamente la creación de los volúmenes para las aplicaciones implica los mismos pasos que se usaron para crear el volumen respaldos del grupo rootdg, lo que cambian son los discos, los nombres de los discos, los nombres del grupo y los tamaños de los volúmenes.

El primer plex de cada volumen quedará en el arreglo yodo, y el segundo plex en el arreglo yodo2, de esta manera los dos plexes de cada volumen estarán en una controladora de discos diferente, por lo tanto si una controladora completa falla y con ello todos los discos de su arreglo dejan de funcionar dejará de funcionar el plex de cada volumen correspondiente a esa controladora, pero tendremos disponible la otra controladora con todos sus plexes y por lo tanto nuestro sistema seguirá funcionando.

Necesitamos 148 GB de capacidad en cada arreglo de discos para recrear la estructura de sistemas de archivos de datos de dragón. Ese espacio lo cubrimos con los 9 discos de 18 GB que tenemos en cada arreglo A5000 destinados para eso. Como mencionamos en este capítulo anteriormente, los discos que utilizaremos para los datos de aplicaciones y usuarios son los de

la posición 1, 2, 3, 4, 5 y 6 de la parte frontal tanto de yodo como de yodo2, y los de las posiciones 1, 2 y 3 de la parte trasera también de ambos arreglos de discos. A continuación presentamos una tabla de los sistemas de archivos de los datos de aplicaciones de dragón, en la primer columna presentamos el punto de montura del sistema de archivos, y en la segunda columna el tamaño que deberá tener cada volumen que hagamos, este tamaño es el mismo que mostramos anteriormente, solo que lo estamos redondeando.

Punto de montura de sistemas de archivos	Tamaño en GB
/home/users00	7
/home/users01	7
/home/users02	7
/home/users03	7
/home/users04	4
/mirror/home	31
/mirror/users00	7
/home	31
/home/log	31
/sybase	7
/raid	4
/raid2	4
/usr/local/pgsql	2
/usr/sybase	5

Tabla 3.1. Sistemas de archivos de aplicaciones y usuarios de dragon

Sigamos los siguientes pasos para la creación de los volúmenes de las aplicaciones.

Primero creamos e inicializamos el grupo de discos de las aplicaciones. Decidimos nombrar al grupo como "datosdg" para hacer referencia a la información que tendrá. Con el propósito de que el nombre del disco de Volume Manager fuera ilustrativo, decidimos nombrar

a cada disco empezando por el nombre del arreglo, después una "f" si el disco está en la parte frontal de la caja o una "t" si el disco está en la parte trasera, después le sigue un número que indica la posición del disco en la caja. Obviamente este nombre de disco hace referencia al nombre físico del disco como se vio en la parte teórica de Volume Manager.

```
###Con la siguiente linea inicializamos el grupo datosdg con el disco "yodof1".

# vxdg init datosdg yodof1=c1t1d0
```

El siguiente paso es inicializar cada disco físico y agregarlos al grupo de discos datosdg. Para esto hicimos dos scripts y los ejecutamos. El disco c1t1d0 ya no se toma en cuenta porque ya fue puesto bajo el control de Volume Manager en el paso anterior.

```
### Script que inicializa los discos del arreglo yodo

/etc/vx/bin/vxdisksetup -i c1t2d0
/etc/vx/bin/vxdisksetup -i c1t3d0
/etc/vx/bin/vxdisksetup -i c1t4d0
/etc/vx/bin/vxdisksetup -i c1t5d0
/etc/vx/bin/vxdisksetup -i c1t6d0
/etc/vx/bin/vxdisksetup -i c1t17d0
/etc/vx/bin/vxdisksetup -i c1t18d0
/etc/vx/bin/vxdisksetup -i c1t19d0

### Script que nombra  los discos del arreglo yodo que usará el grupo datosdg

/etc/vx/bin/vxdg -g datosdg adddisk yodof2= c1t2d0
/etc/vx/bin/vxdg -g datosdg adddisk yodof3= c1t3d0
/etc/vx/bin/vxdg -g datosdg adddisk yodof4= c1t4d0
/etc/vx/bin/vxdg -g datosdg adddisk yodof5= c1t5d0
/etc/vx/bin/vxdg -g datosdg adddisk yodof6= c1t6d0
/etc/vx/bin/vxdg -g datosdg adddisk yodot1= c1t17d0
/etc/vx/bin/vxdg -g datosdg adddisk yodot2= c1t18d0
/etc/vx/bin/vxdg -g datosdg adddisk yodot3= c1t19d0
```

6.- Creación de volúmenes del grupo datosdg.

Ya que tenemos dados de alta los discos con sus nombres respectivos, creamos los volúmenes con un plex. También aquí utilizamos un pequeño script.

```
### Script que crea los volúmenes con su primer plex

 vxassist -g datosdg make users00 7g  layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4 yodof5
yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make users01 7g  layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4 yodof5
yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make users02 7g  layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4 yodof5
yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make users03 7g  layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4 yodof5
yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make users04 4g  layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4 yodof5
yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make home   31g layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4 yodof5
yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make mirrorhome   31g layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4
yodof5 yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make mirrorusers00 7g  layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4
yodof5 yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make homelog 31g layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4 yodof5
yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make sybase   7g  layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4 yodof5
yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make raid   4g layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4 yodof5
yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make raid2   4g layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4 yodof5
yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make pgsql   2g layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4 yodof5
yodof6 yodot1 yodot2 yodot3
 vxassist -g datosdg make usrsybase   5g layout=stripe ncolumn=9 yodof1 yodof2 yodof3 yodof4
yodof5 yodof6 yodot1 yodot2 yodot3
```

7.- Creación de los sistemas de archivos de los volúmenes. Una vez creados los volúmenes es necesario crear sus correspondientes sistemas de archivos, es aquí donde se hace uso del software instalado VERITAS File System. Veamos el script correspondiente.

```
### Este ciclo for de shell de unix, crea un sistema de archivos para cada volumen que creamos en
el punto anterior, ### nótese que no se está especificando el tamaño de cada sistema de archivos,
debido a que toma el tamaño de cada ### volumen. También en este ciclo le estamos indicando que
los sistemas de archivo van a ser de tipo  vxfs, es decir ### de VERITAS file system, y además
que van a soportar tamaños largos de archivos.

for i in users00 users01 users02 users03 users04 home mirrorhome mirrorusers00 homelog sybase
raid raid2 pgsql usrsybase
> do
> mkfs -F vxfs -o  largefiles,bsize=8192 /dev/vx/rsdk/datosdg/$i
```

8.- Creación de los puntos de montura y montado de sistemas de archivo.

```
### El siguiente script crea los puntos de montura para cada sistema de archivos. Es importante
que estos puntos de ### montura tengan el nombre que tenian originalmente

mkdir /home/users00
mkdir /home/users01
mkdir /home/users02
mkdir /home/users03
mkdir /home/users04
mkdir  /mirror/home
mkdir /mirror/users00
mkdir /home
mkdir /home/log
mkdir /sybase
mkdir /raid
mkdir /raid2
mkdir /usr/local/pgsql
mkdir /usr/sybase

### Para montar los sistemas de archivos se utilzó el siguiente script.

mount -F vxfs /dev/vx/dsk/datosdg/users00 /home/users00
mount -F vxfs /dev/vx/dsk/datosdg/users01 /home/users01
mount -F vxfs /dev/vx/dsk/datosdg/users02 /home/users02
mount -F vxfs /dev/vx/dsk/datosdg/users03 /home/users03
mount -F vxfs /dev/vx/dsk/datosdg/users04 /home/users04
mount -F vxfs /dev/vx/dsk/datosdg/home    /home
mount -F vxfs /dev/vx/dsk/datosdg/mirrorhome   /mirror/home
mount -F vxfs /dev/vx/dsk/datosdg/mirrorusers00   /mirror/users00
mount -F vxfs /dev/vx/dsk/datosdg/homelog /home/log
mount -F vxfs /dev/vx/dsk/datosdg/sybase /sybase
mount -F vxfs /dev/vx/dsk/datosdg/raid    /raid
mount -F vxfs /dev/vx/dsk/datosdg/raid2  /raid2
mount -F vxfs /dev/vx/dsk/datosdg/pgsql   /usr/local/pgsql
mount -F vxfs /dev/vx/dsk/datosdg/usrsybase   /usr/sybase
```

9.- Automatizar el montado de sistemas de archivos.

Agregar los sistemas de archivos creados en su correspondiente volumen al archivo /etc/vfstab. Se agregan tanto los volúmenes del grupo rootdg como los del grupo datosdg. Esto se hace con la finalidad de que cuando el servidor reinicie monte de manera automática estos sistemas de archivos ya que este archivo de configuración es leído siempre que un servidor inicia su funcionamiento. Las letras en negritas son las lineas que se agregaron. Recordemos que anteriormente también se agregaron las líneas correspondientes a DiskSuite.

```
### En seguida se muestra el contenido del archivo /etc/vfstab


#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
```

```
fd      -       /dev/fd fd      -       no      -
swap    -       /tmp    tmpfs   -       yes     -
/proc   -       /proc   proc    -       no      -
/dev/md/dsk/d0  /dev/md/rdsk/d0 /       ufs     1       no      -
/dev/md/dsk/d1  -       -       swap    -       no      -
/dev/md/dsk/d2  /dev/md/rdsk/d2 /usr    ufs     1       no      -
/dev/md/dsk/d3  /dev/md/rdsk/d3 /var    ufs     1       no      -
/dev/md/dsk/d4  /dev/md/rdsk/d4 /opt    ufs     2       yes     -
/dev/vx/dsk/rootdg/respaldos    /dev/vx/rdsk/rootdg/respaldos   /respaldos vxfs    3     yes
-
/dev/vx/dsk/datosdg/users01 /dev/vx/rdsk/datadg/users01        /users01   vxfs    3       yes
-
/dev/vx/dsk/datosdg/users02 /dev/vx/rdsk/datadg/users02        /users02   vxfs    3       yes
-
/dev/vx/dsk/datosdg/users03 /dev/vx/rdsk/datadg/users03        /users03   vxfs    3       yes
-
/dev/vx/dsk/datosdg/users04 /dev/vx/rdsk/datadg/users04        /users04   vxfs    3       yes
-
/dev/vx/dsk/datosdg/home /dev/vx/rdsk/datadg/home       /home   vxfs    3       yes     -
/dev/vx/dsk/datosdg/mirrorhome /dev/vx/rdsk/datadg/mirrorhome       /mirror/home   vxfs    3
yes     -
/dev/vx/dsk/datosdg/mirrorusers00 /dev/vx/rdsk/datadg/mirrorusers00       /mirror/users00   vxfs
3       yes     -
/dev/vx/dsk/datosdg/homelog /dev/vx/rdsk/datadg/homelog        /home/log   vxfs    3       yes
-
/dev/vx/dsk/datosdg/sybase /dev/vx/rdsk/datadg/sybase       /sybase   vxfs    3       yes     -
/dev/vx/dsk/datosdg/raid /dev/vx/rdsk/datadg/raid       /raid   vxfs    3       yes     -
/dev/vx/dsk/datosdg/raid2 /dev/vx/rdsk/datadg/raid2       /raid2   vxfs    3       yes     -
/dev/vx/dsk/datosdg/pgsql /dev/vx/rdsk/datadg/pgsql       /usr/local/pgsql   vxfs    3       yes
-
/dev/vx/dsk/datosdg/usrsybase /dev/vx/rdsk/datadg/usrsybase       /usr/sybase   vxfs    3
yes     -
```

10.- Hasta aquí, hemos creado los volúmenes con un plex en cada uno de ellos, y a su vez su correspondiente sistema de archivos y también los hemos montado, todo esto en el servidor dragón. El siguiente paso es recuperar la información de las aplicaciones y de los usuarios, como recordaremos esta información quedó en los discos del arreglo yodo2, lo que haremos ahora será deportar el grupo de discos tempodg de la máquina 3500 que tiene conectado el arreglo yodo2 e importarlo en el servidor web en éste último también crearemos puntos de montura temporales para montar allí los volúmenes del grupo tempodg, una vez montados haremos una copia hacia los volúmenes que creamos en los puntos anteriores. Posteriormente desmontaremos los volúmenes temporales y se harán pruebas de funcionamiento del servidor. Una vez que se haya probado que todos los datos están bien, destruiremos el grupo de disco tempodg del servidor 3500 y del servidor web, utilizaremos los discos que se liberen para crear los espejos de los volúmenes de dragón del grupo de discos datosdg.

Este procedimiento no lo describiremos, solo basta saber que nuevamente los datos de las aplicaciones y usuarios están disponibles en el servidor web, de esta forma tenemos los volúmenes de los datos y aplicaciones con un plex, el cual, contiene los datos pero aún no están espejeados, así que lo que haremos a continuación será crear para cada plex de cada

volumen un segundo plex o espejo el cual tendrá exactamente los mismos datos que el primer plex.

11.-Creación de espejos de volúmenes.

Una vez liberados los discos del arreglo yodo2 se utilizan para crear los espejos de los volúmenes de dragón. Primero se inicializan los discos y se dan de alta en Volume Manager y después se crean los espejos. Regularmente cuando se genera un espejo el servidor tiene mas carga de trabajo, así que no podíamos mandar crear todos los espejos al mismo tiempo, lo que se hizo fue mandar de tres en tres de manera manual y cuando terminaban de generarse esos tres espejos, mandábamos otro bloque de tres y así sucesivamente hasta que terminamos con los catorce espejos requeridos.

```
### Script que inicializa los discos del arreglo yodo2

/etc/vx/bin/vxdisksetup -i c2t2d0
/etc/vx/bin/vxdisksetup -i c2t3d0
/etc/vx/bin/vxdisksetup -i c2t4d0
/etc/vx/bin/vxdisksetup -i c2t5d0
/etc/vx/bin/vxdisksetup -i c2t6d0
/etc/vx/bin/vxdisksetup -i c2t17d0
/etc/vx/bin/vxdisksetup -i c2t18d0
/etc/vx/bin/vxdisksetup -i c2t19d0

### Script que nombra  los discos del arreglo yodo2 que usará el grupo datosdg

/etc/vx/bin/vxdg -g datosdg adddisk yodo2f2= c2t2d0
/etc/vx/bin/vxdg -g datosdg adddisk yodo2f3= c2t3d0
/etc/vx/bin/vxdg -g datosdg adddisk yodo2f4= c2t4d0
/etc/vx/bin/vxdg -g datosdg adddisk yodo2f5= c2t5d0
/etc/vx/bin/vxdg -g datosdg adddisk yodo2f6= c2t6d0
/etc/vx/bin/vxdg -g datosdg adddisk yodo2t1= c2t17d0
/etc/vx/bin/vxdg -g datosdg adddisk yodo2t2= c2t18d0
/etc/vx/bin/vxdg -g datosdg adddisk yodo2t3= c2t19d0

### En seguida se muestran los bloques de lineas que ejecutamos para crear los espejos de los
### volúmenes en el orden en que lo hicimos.

# vxassist -g datosdg mirror users00  layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3
# vxassist -g datosdg mirror users01  layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3
# vxassist -g datosdg mirror home      layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3

# vxassist -g datosdg mirror users02  layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3
# vxassist -g datosdg mirror users03  layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3
# vxassist -g datosdg mirror mirrorhome   layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3

# vxassist -g datosdg mirror users04  layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3
```

```
# vxassist -g datosdg mirror mirrorusers00  layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3
yodo2f4 yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3
# vxassist -g datosdg mirror homelog      layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3

# vxassist -g datosdg mirror sybase  layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3
# vxassist -g datosdg mirror raid  layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3
# vxassist -g datosdg mirror raid2    layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3


# vxassist -g datosdg mirror pgsql  layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3
# vxassist -g datosdg mirror usrsybase  layout=stripe nstripe=9 yodo2f1 yodo2f2 yodo2f3 yodo2f4
yodo2f5 yodo2f6 yodo2t1 yodo2t2 yodo2t3
```

Para verificar cuándo la sincronización o espejeo de los volúmenes terminaba y así ejecutar otra línea, utilizamos ### el comando "vxtask list", el cual nos muestra el progreso de cada espejo que se esté sincronizando. Cuando
terminaron de espejearse todos los volúmenes este comando no desplegó nada.

```
# vxtask list
```

12.- Configuración del disco de reserva

Los discos que utilizaremos como discos de reserva son el c1t22d0 del arreglo yodo y el c2t22d0 del arreglo yodo2. Como ya hemos visto en la teoría este tipo de discos están en espera de que algún otro disco del arreglo falle para entrar en acción, cuando esto sucede la información que estaba contenida en el disco que falló es movida al disco de reserva y de esta manera nos proteje contra posibles pérdidas de datos.

```
### Inicializar los dos discos que estarán como discos de reserva.

dragon # vxdisksetup -i c1t22d0
dragon # vxdisksetup -i c2t22d0

### Poner los discos bajo el control de Volume Manager

dragon # vxdg -g datosdg adddisk yodot6= c1t22d0
dragon # vxdg -g datosdg adddisk yodo2t6= c2t22d0

### Configuración de discos de reserva

dragon # vxedit -g datosdg set spare=on yodot6
dragon # vxedit -g datosdg set spare=on yodo2t6
```

Configuración final de discos de aplicaciones y usuarios

De esta forma terminamos la configuración de volúmenes para los datos de las aplicaciones y usuarios. Quedaron volúmenes espejeados con el manejador de discos VERITAS Volume Manager. Todos los volúmenes quedan con dos plexes que tienen exactamente la misma información; el primer plex de cada volumen está configurado sobre los discos del arreglo yodo y utiliza una distribución de striping; el segundo plex de cada volumen queda configurado sobre los discos del arreglo yodo2 y también utiliza una configuración de striping; el segundo plex a su vez es un espejo del primero. Con esto hemos llegado a la configuración RAID 0+1 que pretendíamos.

La nueva configuración de los discos de las aplicaciones del servidor web de la UNAM queda como se muestra a continuación:

```
Filesystem kbytes used avail capacity Mounted on
/dev/md/dsk/d0 617275 327903 233818 59% /
/dev/md/dsk/d2 4131866 2646600 1443948 65% /usr
/proc 0 0 0 0% /proc
fd 0 0 0 0% /dev/fd
/dev/md/dsk/d3 3099287 2162276 875026 72% /var
/dev/md/dsk/d4 2056211 466484 1528041 24% /opt
swap 3034584 152 3034432 1% /tmp
/dev/vx/datosdg/dsk/mirrorhome 30945914 27291825 3344630 90% /mirror/home
/dev/vx/datosdg/dsk/homelog 30945914 12978909 17657546 43% /home/log
/dev/vx/datosdg/dsk/users00 6186810 4193040 1931902 69% /mirror/users00
/dev/vx/datosdg/dsk/users01 6186810 3778354 2346588 62% /home/users01
/dev/vx/datosdg/dsk/users02 6186810 3492786 2632156 58% /home/users02
/dev/vx/datosdg/dsk/users03 6186810 5857917 267025 96% /home/users03
/dev/vx/datosdg/dsk/users04 3871954 2069039 1764196 54% /home/users04
/dev/vx/datosdg/dsk/sybase 6186810 2881346 3243596 48% /sybase
/dev/vx/datosdg/dsk/raid 3871954 1961986 1871249 52% /raid
/dev/vx/datosdg/dsk/raid2 3871954 1100983 2732252 29% /raid2
/dev/vx/datosdg/dsk/pgsql 1527116 1140770 325262 78% /usr/local/pgsql
/dev/vx/datosdg/dsk/sybase 4743974 3615240 1081295 77% /usr/sybase
/dev/vx/datosdg/dsk/home 30957590 28151616 2496399 92% /home
/dev/vx/datosdg/dsk/users00 6199998 3354000 2783999 55% /home/users00
```

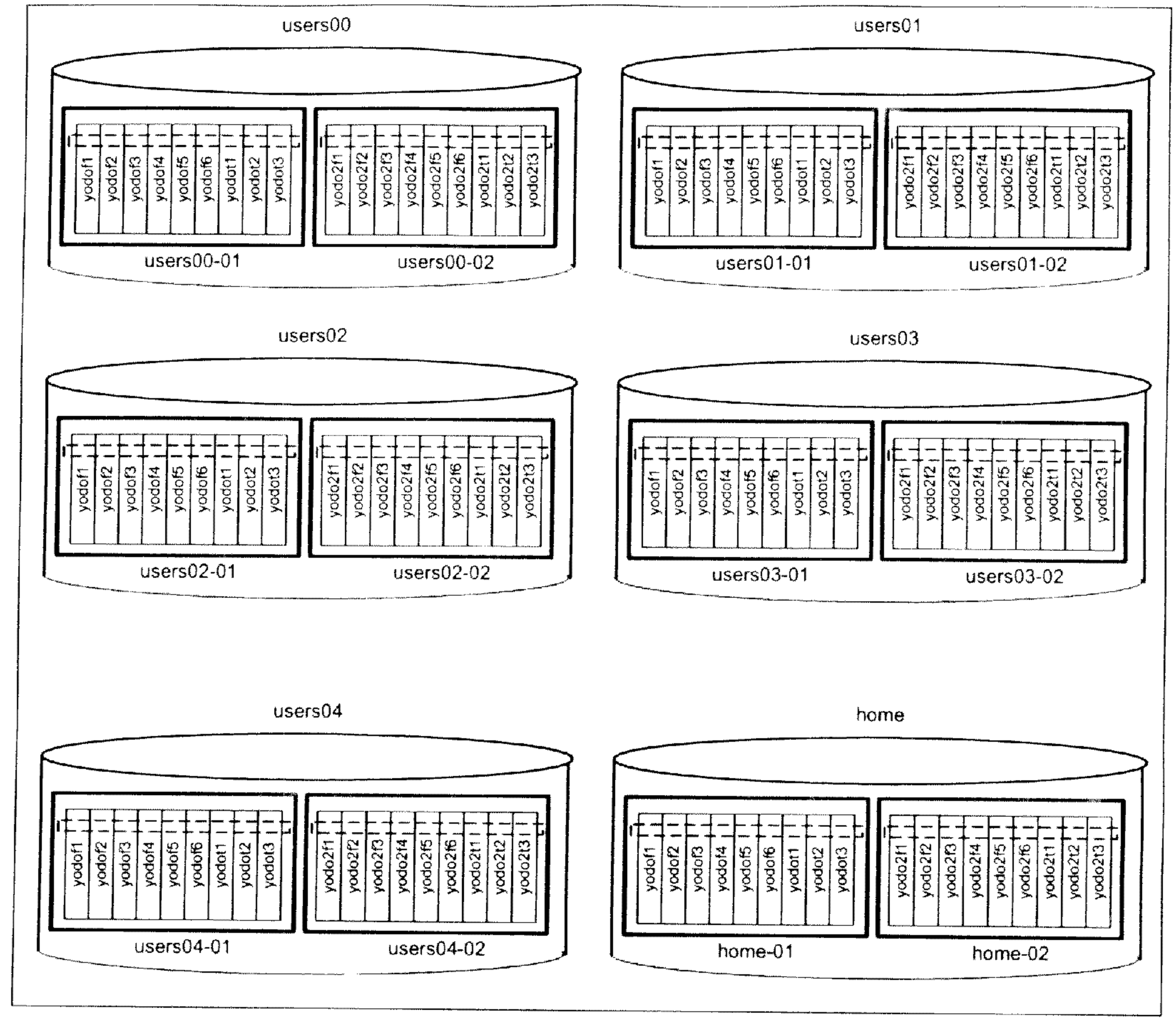


Figura 3.28a. Volúmenes finales del servidor WEB

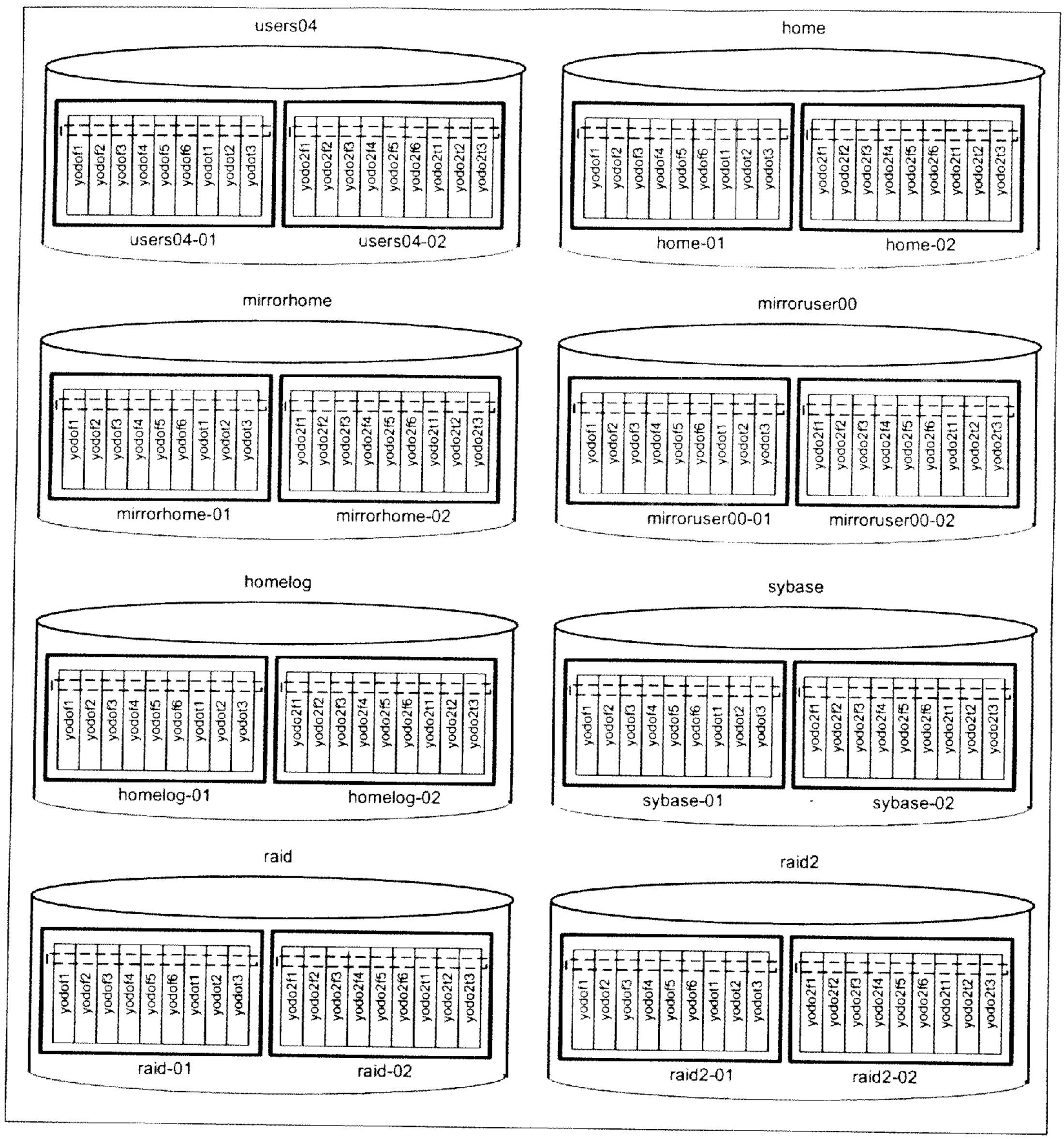


Figura 3.28b. Volúmenes finales del servidor WEB

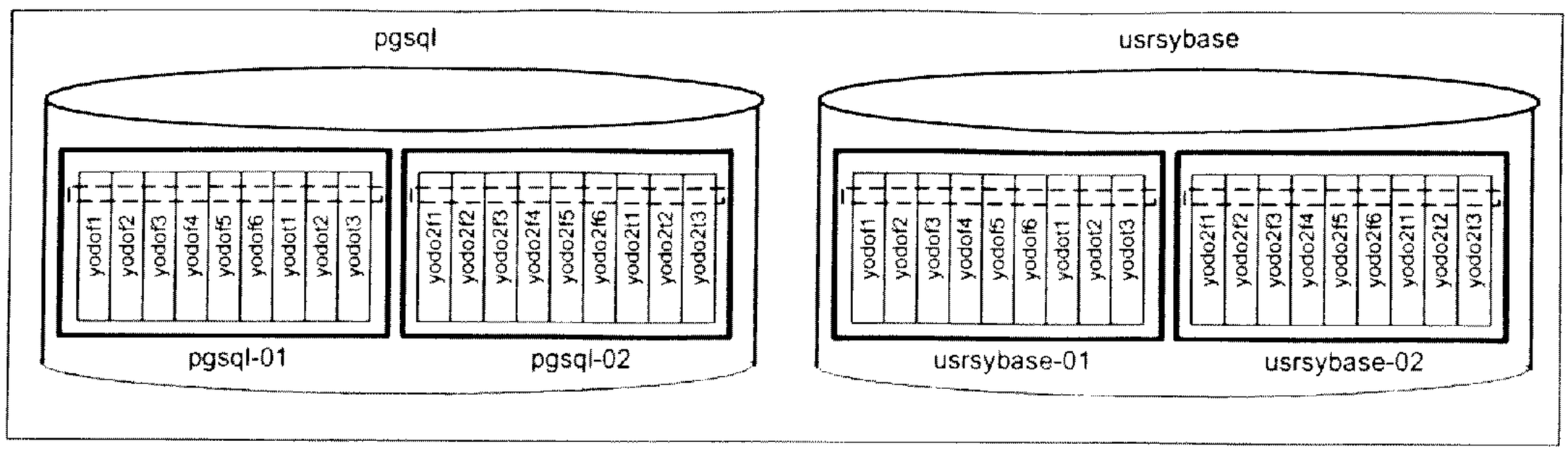


Figura 3.28c. Volúmenes finales del servidor WEB

Capítulo 4. Replicación de Datos

La replicación nos permite tener dos copias de los mismos datos, en discos totalmente diferentes y normalmente en servidores ubicados a varios kilómetros de distancia. La replicación de datos puede implementarse sobre sistemas de archivos, servidores WEB, o bases de datos. En este capítulo nos enfocaremos a ver la replicación de datos sobre sistemas de archivos por ser el tipo de replicación que propondremos implementar en la UNAM, los servidores que utilizaremos para esto son el servidor web y el servidor newton. Primeramente hablaremos de algunos conceptos y técnicas para replicar datos y posteriormente haremos un análisis de la configuración actual de newton el cual nos va a permitir darnos cuenta en qué es diferente newton a dragón y lo que debemos cambiar. Finalmente haremos los cambios necesarios en newton y aplicaremos la técnica de replicación de datos mediante sistemas de archivos.

Significado de replicación

La replicación de datos significa mover datos de un conjunto de discos a otro conjunto de discos completamente redundante. La replicación no es lo mismo que el espejeo de discos ya que este último trata a los discos como un volumen lógico mientras que la replicación trata a los discos como dos entidades completamente diferentes.

Se debe tomar en cuenta que para hacer una replicación adecuada se necesitan dos conjuntos de discos independientes conectados físicamente a servidores diferentes localizados idealmente en lugares geográficos distintos, aunque también pueden estar en el mismo centro de cómputo, además se necesita un software especializado que es el encargado directo de replicar la información entre ambas máquinas. Esto último trae como consecuencia un incremento en el costo de la implementación de la replicación de datos. Sin embargo, puede hacerse replicación de datos utilizando comandos de sistema operativo, aunque con ciertas deventajas.

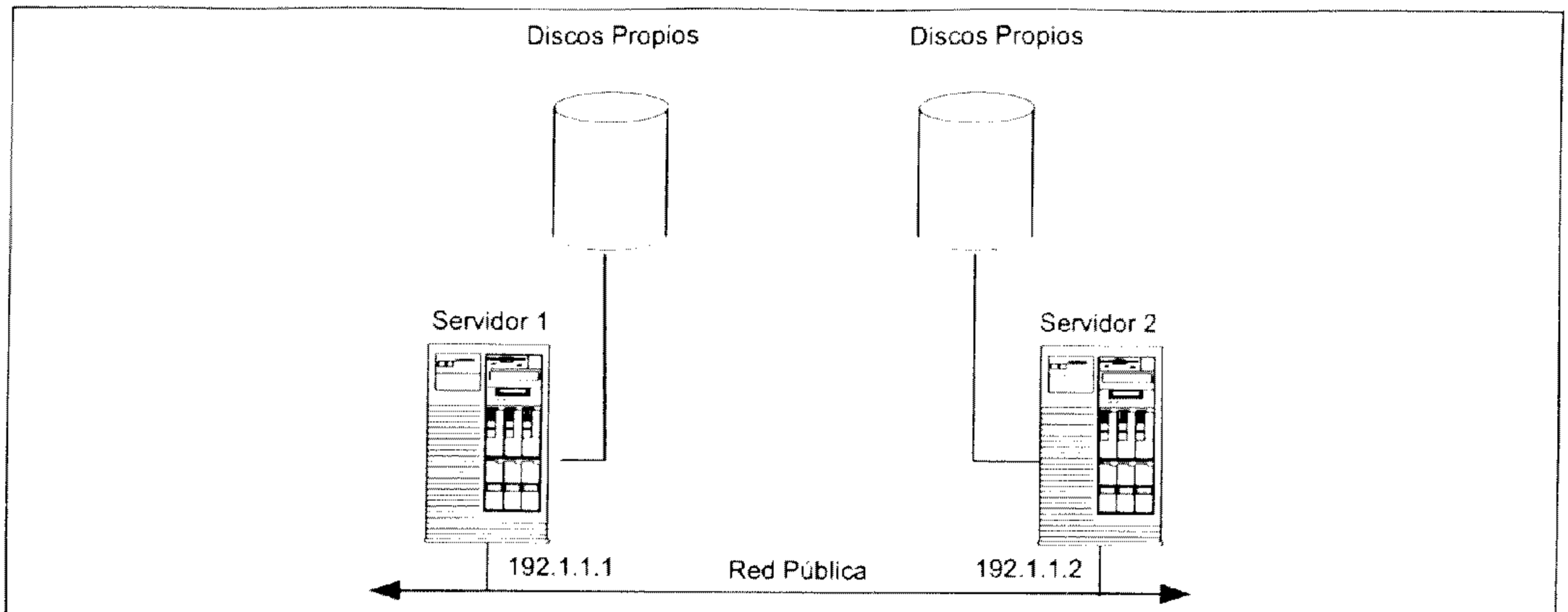


Figura 4.1. Esquema de replicación de datos

Hagámosnos la siguiente pregunta:

¿Por qué replicar los datos si podemos contar con un nivel RAID5 o un espejo de discos o con un cluster que tenga acceso a discos compartidos?

La conclusión a la que llegamos radica en que existen aplicaciones que requieren lapsos de tiempo de segundos para reanudar su funcionamiento, así mismo, algunas operaciones demandan que la corrupción o inconsistencias en las bases de datos sean solucionadas en la menor cantidad de tiempo posible, esto normalmente se logra con la replicación al contar con una copia de datos que está disponible en línea en otro arreglo de discos y no ocupar horas en recuperar un respaldo.

Puntos a considerar en la replicación de datos

A continuación se mencionan algunos puntos que deben considerarse cuando se implementan programas de replicación de datos.

1. - Servidores idénticos en hardware

El tener dos servidores que difieran en el hardware que los compone puede ser un problema muy grave cuando se implementa replicación de datos. Es recomendable que los dos servidores tengan la misma arquitectura, los mismos procesadores a la misma velocidad, la misma cantidad de memoria, las mismas tarjetas de entrada/salida de datos, deben de ser tan iguales como sea posible.

2.- Servidores idénticos en software

El software es un caso especial en la replicación de sistemas de archivos porque hay varias formas en las que el tener software diferente en ambos servidores puede afectar todo el sistema. Las versiones de todos los paquetes incluyendo librerías, herramientas, compiladores, servidores de bases de datos, etc. Deben ser las mismas, esto es crítico ya que de no cumplir con estas condiciones experimentaremos problemas tales como que los programas se ejecuten perfectamente en un servidor y en otro no. Para cumplir con esto, se requiere llevar una bitácora de control de cambios de nuestro servidor, es necesario conocer las versiones de cada uno de los paquetes que se instalan en el sistema, versiones de parches, versiones de firmware de los discos, de las controladoras, etc., esto con el fin de tener la menor cantidad de problemas en nuestra replicación.

Aplicaciones con requerimientos locales de configuración, tales como impresoras o tablas de autenticación de usuarios, deben configurarse por separado y no entrar a un esquema de replicación. Esto es debido a que tal vez se actualicen los archivos de configuración pero lo controladores deben ser cargados por el kernel que es la parte central de sistema operativo.

3.- Balanceo de cargas de los servidores a replicar

Debe procurarse que los servidores a replicarse tengan similar carga de procesamiento, esto con el fin de tener un replicación simétrica en ambos servidores.

4.- Localizar posibles cuellos de botella en la red.

Debe evitarse en servidores candidatos a replicar datos, el contar con sistemas de tipo NFS (Network File System por sus siglas en inglés), el tener este tipo de sistemas de archivos en servidores que van a replicar datos a grandes distancias impacta de gran manera el desempeño de una aplicación, cuando se piensa en replicar datos debe considerarse que todos los sistemas de archivos sean locales a un servidor.

5.- Procurar tiempos cortos de transferencia de servicios y procesos.

Muchas aplicaciones requieren ser nuevamente operacionales en cuestión de segundos cuando una falla ocurre, si a esto le agregamos que el incremento de tiempo en la reanudación de operaciones significa un incremento en la pérdida de dinero para la empresa, nos daremos cuenta de la importancia de este punto, por lo tanto cuando implementemos

replicación de datos deberemos procurar tener tiempos de transferencia de servicios y procesos muy cortos de un servidor a otro.

6.- Recuperación ante un desastre.

La corrupción de una base de datos es un desastre, pero que hay acerca de las intrusiones físicas en el centro de cómputo, fuego, inundaciones, u otro tipo de desastres naturales ? Con un centro de cómputo completamente replicado y listo para funcionar podemos estar nuevamente en línea de manera rápida y sin necesidad de utilizar respaldos de cintas ni reconfiguraciones de sistema. Para lograr esto se requiere una cuidadosa coordinación con la gente involucrada como son desarrolladores de sistema, administradores, operadores y otros, para asegurarnos de que todos los puntos del plan de acción están cubiertos.

Técnicas de Replicación

La replicación puede tratarse de algo tan simple como el tomar un respaldo de un equipo y recuperarlo en otro sistema o puede llegar a algo tan complicado como replicar un dato o proceso que se encuentra en memoria hacia varios servidores a través de la red y que esta copia preste servicio en caso de tener una carga excesiva o una falla. Existen 5 categorías en las técnicas de replicación, las cuales se listan a continuación de la más simple a la más complicada:

1.- Replicación de sistemas de archivos.

Esta es la forma más rápida, fácil y barata de replicar un dato, pero también tiende a reportar fallas, como copias incompletas y otros problemas. Este tipo de copias se pueden hacer con los comandos de sistema operativo como ftp, tar, dump, rdist.

2.- Propagación de escrituras mediante drivers.

La característica principal de este tipo de replicación radica en el bloqueo de las transacciones en el servidor original hasta que las copias remotas hayan terminado.

3.- Replicación de unidades de disco.

Esta técnica utiliza una controladora de discos, para enviar los datos actualizados a otro arreglo de discos, la desventaja de esta técnica radica en que la copia se realiza bloque por bloque y tiene serios impactos en el desempeño de la red.

4.- Replicación transaccional.

En este escenario, las transacciones que vienen de una base de datos son distribuidas a dos o más bases de datos. Algunas aplicaciones de bases de datos pueden realizar la distribución en una sola transacción y en una segunda fase realizar una actualización para asegurar que la transacción es aplicada en todos los sistemas.

5.- Replicación de estado de nivel-proceso.

Las primeras cuatro técnicas solamente se aseguran que los datos sean copiados a múltiples lugares. En ambientes en los cuales se cuenta con grandes volúmenes de información, es necesario guardar una copia en memoria de los datos actualizados, la cual puede ser propagada a varios servidores. La replicación de estado de nivel-proceso se encarga de enviar el dato requerido a cada aplicación. Esta técnica también comprende un área de registro, la cual se encarga de actualizar los datos mas importantes a disco y en caso de sufrir el servidor una falla, no es necesario verificar la integridad de varios gigas de información, las últimas transacciones fueron guardadas en disco y son actualizadas al momento en que inicia el equipo.

Replicación de sistemas de archivos

La replicación de sistemas de archivos es la técnica más común. Se elige la replicación de sistemas de archivos cuando en una empresa se cuenta con una cantidad considerable de desarrolladores que demandan capacidad de cómputo en múltiples servidores como: manuales, librerías, compiladores, etc. El contar con múltiples servidores nos provee el primer nivel de redundancia contra una falla en un servidor, y puede remover cuellos de botella al distribuir las cargas en varias máquinas que cuenten con una copia de los datos.

La replicación tipo lectura-escritura es utilizada principalmente para "Recuperación ante Desastres", para el cual debemos contar con un centro de cómputo remoto que es conocido como "centro de cómputo alterno o secundario" y este contiene una copia lo mas

idéntica posible de un centro de cómputo primario. El utilizar una copia remota permite recuperar un servicio en cuestión de minutos, ya que es más rápido hacer uso de esta información que el tener que recuperar un respaldo con gran cantidad de información. Esta información normalmente es utilizada en caso de una contingencia, con frecuencia en caso de una inundación, un incendio. La replicación tipo lectura-escritura también es utilizada para actualizar múltiples sistemas de archivos en línea, aquí se debe contar con un servidor maestro, el cual cuenta con la copia más actual de un dato y que replicará a varios servidores esclavos.

Una idea errónea que algunos administradores tienen, es que la replicación nos permite eliminar un esquema de respaldos, la replicación guarda una copia fiel de los datos actuales, pero si el director de nuestra empresa requiere un correo que borro hace dos días, forzosamente tendremos que recuperarlo de un respaldo, ya que si el dato es borrado en servidor maestro también será borrado en los servidores esclavos.

Distribución de archivos

Una manera muy simple de replicar datos es aplicando técnicas de respaldos, con el fin de crear una imagen completa de un sistema de archivos, el cual será actualizado mediante respaldos incrementales, se recomienda que estos equipos trabajen por redes privadas de respaldo con el fin de no impactar nuestra red.

No es una mala idea el que nuestras unidades de respaldo sean rápidas y contengan varios dispositivos de lectura-escritura de cintas, debemos ver que el contar con unidades DLT y no DDS es una inversión y no un gasto. La parte crucial en este tipo de replicación no radica en al momento de realizar el respaldo, radica principalmente al momento de recuperar los datos y más aún si la información a extraerse es del orden de cientos de gigas.

Es sensato el considerar que los respaldo sean multiniveles, esto con el fin de realizar lo menos posible respaldos completos, que tal seguramente tardarán muchas más horas en terminarse, una vez realizado un respaldo completo se pueden realizar varios respaldos incrementales cada 30 minutos, si es que la información cambia constantemente. También es viable mas no recomendable el utilizar herramientas de sistema operativo como tar, cpio y zip para realizar esto propósitos.

Lo siguiente en complejidad son las utilerías de distribución como *rdist* o en su versión segura ***sdist*** el cual utiliza un canal cifrado de comunicación entre dos servidores. *rdist* compara los tiempos del servidor local y el remoto y solo enviará los archivos que requieran actualizarse, cuidando así el ancho de banda de nuestra red, también posee reglas que nos

permiten especificar los sistemas de archivos que se requieren replicar e inclusive únicamente los archivos de nuestro interés. Sin embargo, se debe tener un especial cuidado en los problemas de la red, una interrupción de pocos segundos puede echar a perder un replicación de varias horas, los permisos de los archivos o directorios pueden causar también problemas al momento de la replicación. Las herramientas que nos permiten verificar los tamaños de los archivos como tripwire nos ayudan de gran forma a verificar la integridad de los archivos que vamos a replicar y por ultimo se recomienda tener especial cuidado si se van a replicar archivos de mas de 2 gigas, para esto los sistemas de archivos debieron haberse creado y montado con la característica de archivos largos proporcionada por el software Veritas File System.

Puesta a punto de los servidores de replicación

Como ya hemos mencionado, la técnica de replicación que utilizaremos es la de replicación de sistemas de archivos a través de la red. Por razones de costos no es posible utilizar una técnica más confiable ya que esto implicaría que la universidad gastara una buena cantidad de dinero tanto en software como en hardware. Para replicar los datos del servidor web de la UNAM utilizaremos el servidor llamado dragón. Dado que la replicación de datos requiere tener dos servidores lo más idénticos posibles tanto en hardware como en software, lo primero que haremos es una comparación entre servidor newton al servidor dragon, y posteriormente tratar de que ambos servidores queden lo mas parecidos que se pueda tanto en hardware como en software. Debemos mencionar que la aplicación de servidor web no se instalará en el servidor newton.

Hardware de dragón y de newton

A continuación listamos las características principales de hardware de dragón y de newton.

Nombre de dominio	Dragon
Tipo de sistema	SUNW,Ultra-Enterprise-10000
Arquitectura del kernel	Sun4u
Tarjetas de sistema utilizadas	1 (tarjeta de sistema 1)
Procesadores	4 (a 334 MHZ c/u)
Memoria RAM	2 GB
Tarjetas de entrada/salida de datos	3 (1 qfe para red, 1 ultra wide combo SCSI para discos D1000 y una fcal para discos que usan fibra óptica
Arreglos de discos	2 A5000 y un D1000

Tabla 4.1 hardware del servidor dragón.

Nombre de dominio	Newton
Tipo de sistema	SUNW,Ultra-Enterprise-10000
Arquitectura del kernel	Sun4u
Tarjetas de sistema utilizadas	1 (tarjeta de sistema 0)
Procesadores	4 (a 400 MHZ c/u)
Memoria RAM	2 GB
Tarjetas de entrada/salida de datos	4 1 qfe para red, 1 fcal para arreglo A5200, 2 ultra wide SCSI para d1000.
Arreglos de discos	a) 1 A5200 y 1 D1000

Tabla 4.2 hardware del servidor dragón.

La información anterior nos dice que el servidor web y el servidor newton son muy parecidos en cuanto a hardware, ya que los dos están sobre una plataforma Sun Ultra-Enterprise-10000; cada uno de ellos consta de una tarjeta de sistema que tiene cuatro procesadores, aunque no son a la misma velocidad si nos servirán para conseguir lo que queremos; en cuanto a memoria RAM también tenemos la misma cantidad en ambos servidores; también en cuanto a tarjetas de entrada/salida de datos, la tarjeta de red es igual en ambos servidores, la tarjeta para conectar discos a través de fibra óptica también es igual, y la tarjeta para conectar discos SCSI tenemos una en dragón y dos en newton, aunque realmente solo utilizamos la de dragón. En lo que se refiere a los arreglos de discos, tenemos que dragón cuenta con 2 arreglos A5000 y un D1000 que se liberó con la reestructuración que se hizo en el capítulo 3, mientras que newton tiene un arreglo A5200 y un D1000. Realmente las diferencias de hardware que tiene los dos servidores no nos afectan demasiado, ya que lo que vamos hacer es una replicación de sistemas de archivos a través de la red y newton solo tendrá el control de los procesos en caso de que falle dragón, los datos replicados serían de utilidad en el caso de que se perdiera cierta información en el servidor web, o que existiera cierta inconsistencia en los datos, o en el caso de que ocurriera un desastre en el centro de cómputo de la UNAM, tomando en consideración los escenarios anteriores podríamos realizar una migración de servicios hacia el servidor newton en el momento en que nosotros lo consideremos pertinente. Bajo este contexto resulta importante tener el espacio de disco suficiente en el arreglo de discos A5200 para crear la misma estructura de discos tanto de sistema operativo como de datos que se creó en el servidor web, aunque debemos mencionar que los discos para los volúmenes de datos que se crearán en este servidor solo nos alcanzarán para un plex, es decir, este site de secundario no tendría redundancia en disco.

Software de dragón y de newton

En cuanto al software de dragón y de newton tenemos lo siguiente:

Nombre de dominio	dragon
Sistema Operativo	Solaris versión 2.8
Cluster de parches	i)
Nivel de kernel	ii) Generic_105181-33
Versión de Soltice Disk Suite	4.2.1
Versión de Veritas Volume Manager	3.04
Versión de Veritas File System	3.3.3

Tabla 4.3. Software actual del servidor dragón.

Nombre de dominio	newton
Sistema Operativo	Solaris versión 2.7
Cluster de parches	
Nivel de kernel	Generic_106541-23 (64-bit)
Versión de Soltice Disk Suite	4.2
Versión de Veritas Volume Manager	No tiene
Versión de Veritas File System	3.3.2

Tabla 4.4. Software actual del servidor newton

En lo referente al software de los dos servidores existen diferencias mas notables, la versión de sistema operativo es menos actual en newton, al igual que los parches instalados y la versión del kernel, además la versión de Disk Suite también es menos reciente, al igual que veritas file system, este servidor no tiene instalado veritas volume manager, siendo esto último muy importante ya que la configuración de discos de datos a la que queremos estará basada en volúmenes generados con volume manager. Hay que tomar en cuenta que respecto a dragón se están tomando las versiones que quedaron después de los cambios que se hicieron en el capítulo 3.

Tomando en cuenta lo anterior, optamos por reinstalar y configurar todo el software del servidor newton, se reinstalará el mismo software y las mismas versiones que se instalaron en el servidor web, y de esta manera dejar idénticos ambos servidores en cuanto a software se refiere.

Configuración actual de discos de newton

Antes que otra cosa analizaremos como están configurados los discos de newton, tanto de forma física como de forma lógica. Tenemos 2 arreglos, un D1000 que tiene solo tres discos de 18 GB y un arreglo A5200 que tiene 10 discos de 18 GB.

El arreglo D1000 está configurado para que la mitad de discos de la caja pertenezcan a una controladora, que es "c0", y la otra mitad a otra controladora diferente, que en este caso es "c1", es decir los discos de la posición 0 a la 5 con manejados por la controladora c0, mientras que los discos de la posición 6 a la 11 por la controladora c1. A continuación se muestra el arreglo D1000 mencionado.

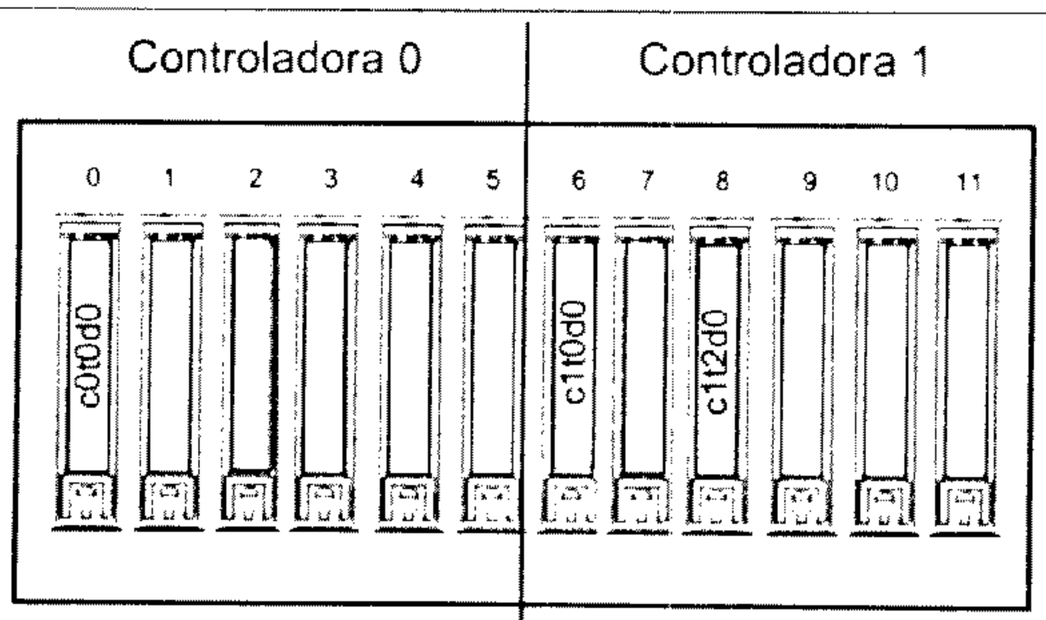


Figura 4.2. Discos del arreglo D1000 del servidor newton

Como vimos en el capítulo 3, el arreglo A5200 es muy parecido al arreglo A5000, solo que tiene capacidad hasta para 22 discos en lugar de 14. El arreglo A5200 que está conectado al servidor newton, se llama también newton y tiene actualmente 10 discos como lo muestra la figura siguiente.

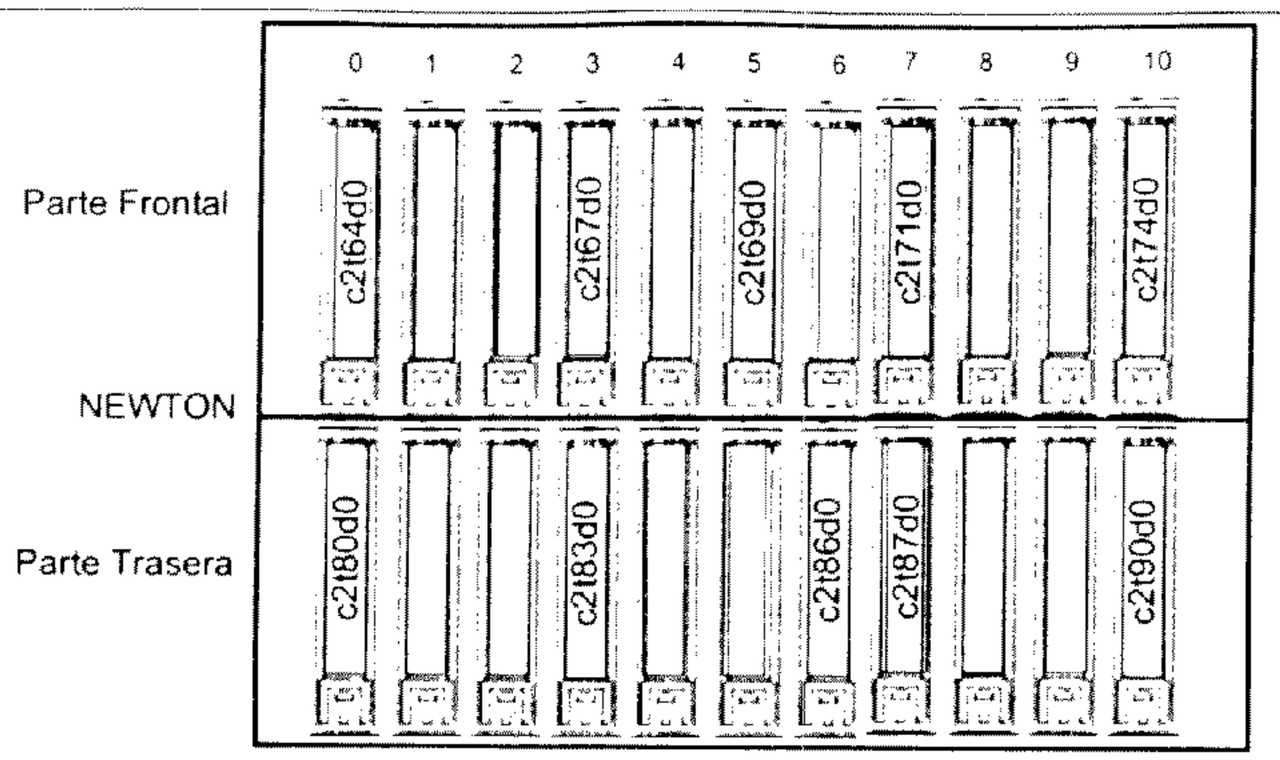


Figura 4.3. Discos del arreglo newton del servidor newton

Ahora bien, el sistema operativo está instalado en los dos discos del arreglo D1000 y está configurado bajo Soltice Disk Suite, tenemos un metaespejo llamado d1 que corresponde a / y que tiene un subespejo formado por la partición 0 del disco c0t0d0 y otro subespejo formado por la partición 0 del disco c1t0d0; un segundo metaespejo d2 correspondiente a swap, que a su vez se conforma de los subespejos d12 y d13, d12 está en la partición s1 del disco c0t0d0 y d13 en la partición s1 del disco c1t0d0; el tercer metaespejo es el d3 que representa a /usr, el cual se compone de d30 y d31, d30 esta formado por la partición s3 del disco c0t0d0 y d 31 por la partición s3 de c1t0d0; el metaespejo d5 de /opt está formado por d50 y d51, d50 a su vez se forma con la partición 5 del disco c0t0d0 y la misma partición del disco c1t0d0; y finalmente el metaespejo d6 que corresponde a /var, está formado por d60 y d61, d60 formado por la partición s6 del disco c0t0d0 y del c1t0d0. Además del sistema operativo en estos discos tenemos el metaespejo d7 que apunta al sistema de archivos /home que contiene información de usuarios, este metaespejo se compone de los subespejos d70 y d71. Respecto al disco c1t2d0 no se estaba utilizando, así que se destinó para la creación del disco alterno de newton que se describe en el capítulo2. Veamos la siguiente figura.

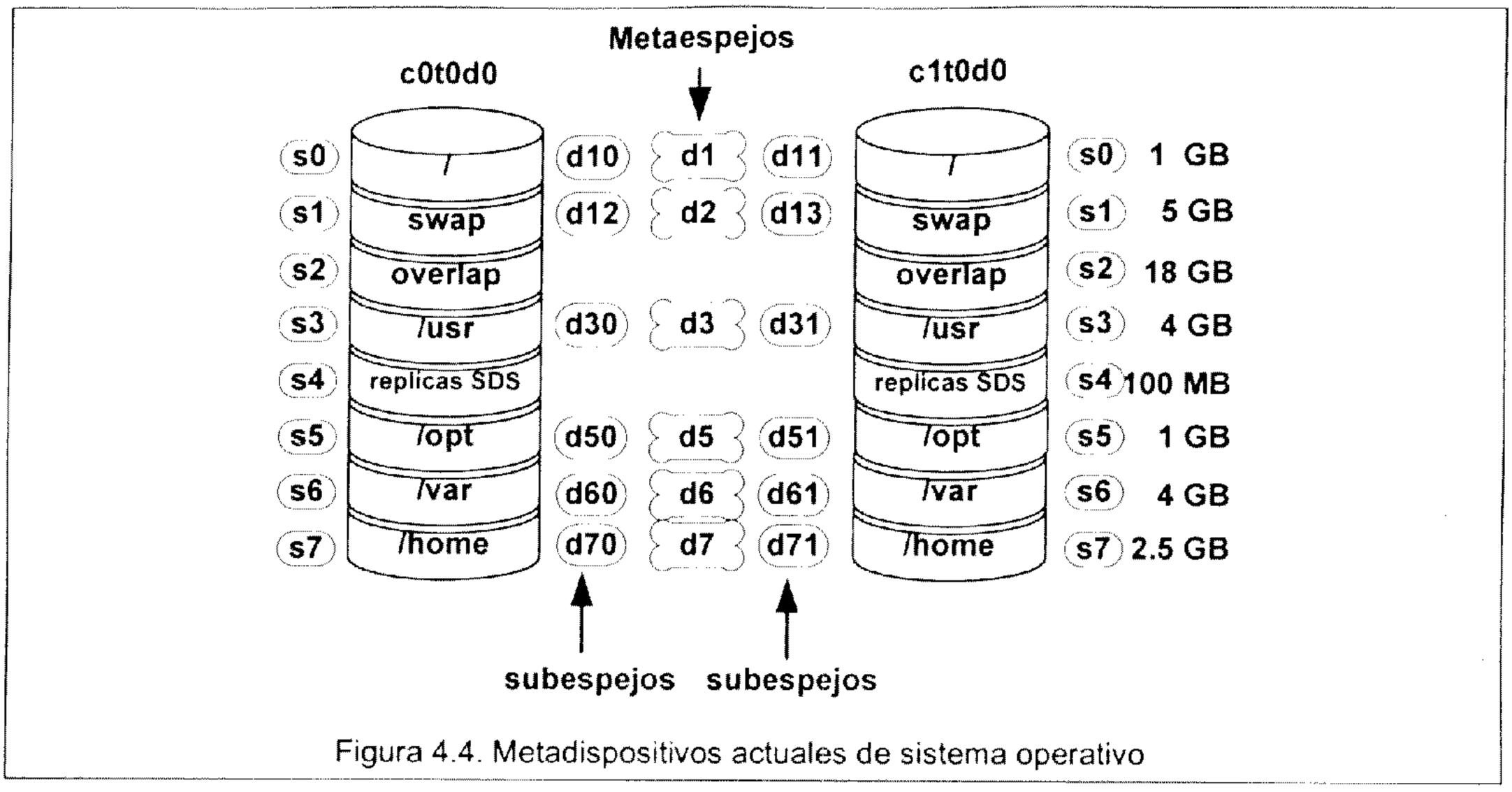


Figura 4.4. Metadispositivos actuales de sistema operativo

Por otro lado los discos que contienen información de los científicos y de los usuarios también se encuentran configurados bajo Disk Suite, solo que para ello utiliza el arreglo A5200. Solo son tres metadispositivos los que están en estos discos y no se está aprovechando todo el espacio disponible. Podemos ver en la figura siguiente que el metadispositivo d8 es un metadispositivo simple en stripe formado por los discos c2t64d0s1 y el c2t69s0s1. El metadispositivo d9 es un metaespejo formado por el subespejo d90 y d91, al subespejo d90 le corresponde el disco c2t74d0s1 y al d91 el disco c2t80d0s1. El tercer metadispositivo es el d66 y también es un metadispositivo simple en stripe formado por los discos c2t90d0s0 y c2t86d0s0.

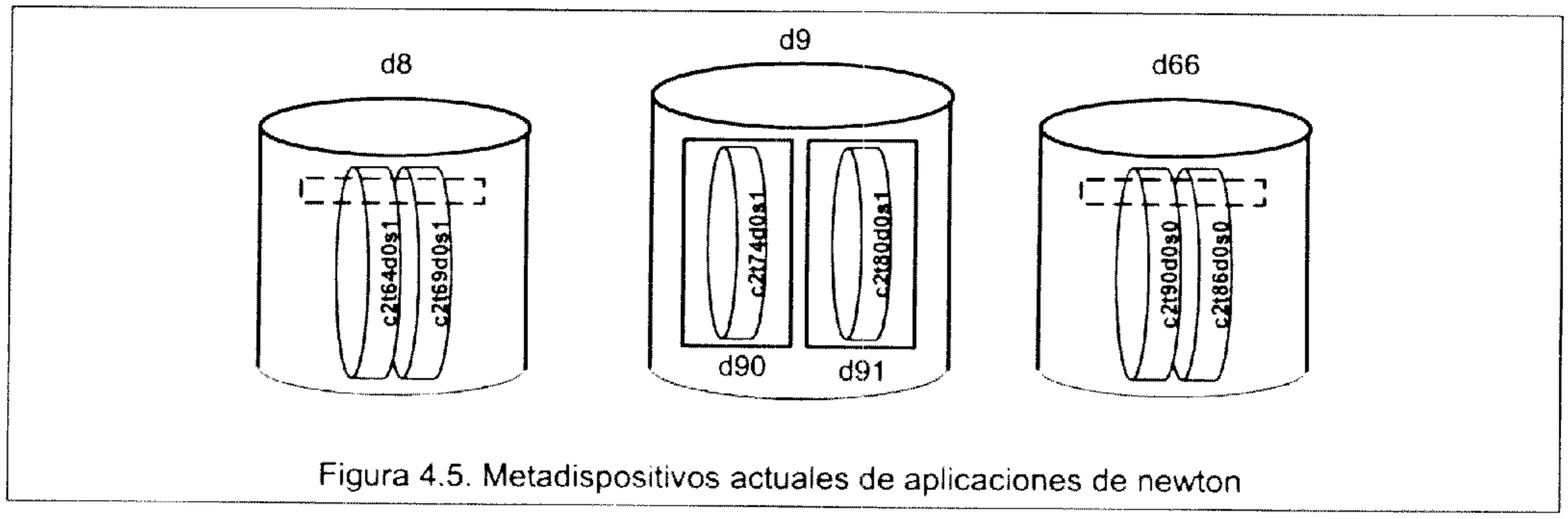


Figura 4.5. Metadispositivos actuales de aplicaciones de newton

De esta forma de los 10 discos que tiene el arreglo newton solo se están ocupando 6, por lo tanto la reestructuración que se haga debe contemplar esto para evitar el desperdicio de recursos.

Nueva Configuración de newton

Con el objetivo de que el servidor newton quede los mas parecido al servidor web, se reinstalará el sistema operativo, el software Soltice Disk Suite, Veritas Volume Manager y Veritas File System, y obviamente se harán las configuraciones necesarias que permitan tener un esquema de discos similar al de dragón. En esencia los procedimientos para hacer lo que pretendemos son los mismos que se utilizaron para dragón solo que hay algunas pequeñas variantes como pueden ser los nombres de los discos y entre otras.

Respaldo de datos actuales

Antes que nada lo primero que haremos es respaldar de alguna manera la información de los científicos y usuarios que actualmente tiene newton para recuperarla posteriormente cuando tengamos la nueva estructura de disco. Como recordaremos en el servidor web quedó un arreglo D1000 libre, este arreglo contiene 4 discos de 9 GB y 3 de cuatro GB, espacio suficiente para almacenar de forma temporal los aproximadamente 34 GB de datos que contienen los sistemas de archivos de dragón /home, /home/users00, /home/users01, /oracle y /home que se encuentra en los mismos discos donde está el sistema operativo, así que para no perder esta información a la hora de que se reinstale también la copiaremos.

El arreglo D1000 está conectado a dragón, así que creamos allí un grupo de volume manager llamado tmpnewtondg, le agregamos los discos del arreglo y creamos un volumen de un plex con los 3 discos de 4 GB y los 4 discos de 9 GB, creamos sus sistema de archivos de 45 gigabytes y generamos un directorio parra cada uno de los cuatro sistema de archivos de newton. Después hicimos un ftp al servidor newton para transferir la información hacia dragón. Cuando la información terminó de transferirse verificamos la integridad de la información y comenzamos la reconfiguración del servidor newton.

Con respecto al sistema operativo se hizo un respaldo a cinta de 4 mm por si se requiriera.

Este procedimiento para respaldar los datos de dragón no lo pusimos de forma detallada porque no es objetivo de este trabajo, solo basta mencionar de forma breve lo que se hizo.

Instalación de sistema operativo

La versión del sistema operativo solaris que se tenía en el servidor newton era la 2.7, la versión a la que se actualizó fue la 2.8, esto se hizo con el propósito de tener la versión del sistema operativo que tiene el servidor web y al mimo tiempo una versión mas reciente y que por lo mismo tuviera menos bugs o errores de software. El sistema operativo se instaló en el disco cotodo del arreglo D1000. Es importante mencionar que la instalación del sistema operativo solo se hizo en el disco cotodo, no será necesario como tal hacer una instalación en el disco c1t0d0, ya que esto se hará a través del software de Disk Suite. En esta ocasión por falta de discos en el arreglo A5200 utilizaremos los discos del arreglo D1000 de newton para instalar el sistema operativo.

Los sistemas de archivo de sistema operativo se definieron del mismo tamaño que los del dragón y quedaron de la siguiente forma.

```
/dev/dsk/c1t0d0s0    2056211  676656 1317869    34%    /
/dev/dsk/c1t0d0s3    4131866  908055 3182493    23%    /usr
/dev/dsk/c1t0d0s4    4131866 1337807 2752741    33%    /var
/dev/dsk/c1t0d0s5    4131866  413741 3676807    11%    /opt
swap                 3034584     152 3034432     1%    /tmp
```

Instalación y configuración de Soltice Disk Suite

En esencia este procedimiento es el mismo que se utilizó en el capítulo 3 para la instalación y configuración de Soltice Disk Suite, lo único que cambian son los discos que se utilizan.

El software se encuentra en el CD 2/2 de Solaris 8 en la ruta siguiente:

```
<CD2>/Solaris_8/EA/products/DiskSuite_4.2.1/sparc/Packages
```

1. Instalr todos los paquetes de Disk Suite versión 4.2.1.

```
newton # pwd
newton # <CD2>/Solaris_8/EA/Products/DiskSuite_4.2.1/sparc/Packages
newton # pkgadd -d .
```

2. Terminada la instalación se instaló el parche recomendado número 108693-07.

```
newton# patchadd 108693-07
```

3. Crear las particiones en el disco espejo de sistema operativo para los metadispositivos de Disk Suite.

3a. Crear una réplica de la tabla de particiones del sistema operativo del disco c0t0d0 que contiene el sistema operativo, al disco que será el espejo c1t0d0.

```
newton # prtvtoc /dev/rdsk/c0t0d0s0 | fmthard -s - /dev/rdsk/c1t0d0s0➔sistema de archivos  /
newton # prtvtoc /dev/rdsk/c0t0d0s1 | fmthard -s - /dev/rdsk/c1t0d0s1➔sistema de archivos  swap
newton # prtvtoc /dev/rdsk/c0t0d0s3 | fmthard -s - /dev/rdsk/c1t0d0s3➔sistema de archivos  /usr
newton # prtvtoc /dev/rdsk/c0t0d0s4 | fmthard -s - /dev/rdsk/c1t0d0s4➔sistema de archivos  /var
newton # prtvtoc /dev/rdsk/c0t0d0s5 | fmthard -s - /dev/rdsk/c1t0d0s5➔sistema de archivos  /opt
newton # prtvtoc /dev/rdsk/c0t0d0s7 | fmthard -s - /dev/rdsk/c1t0d0s7➔ réplicas de Disk Suite

fmthard:  New volume table of contents now in place.
```

Con esto el disco c1t0d0 está dividido en las mismas particiones y del mismo tamaño que el disco c0t0d0 del arreglo D1000 donde se instaló el sistema operativo.

3b. Ejecutar el script hecho en shell de unix para crear las réplicas de las bases de datos de configuración propias de disk suite y los metadispositivos. Se explicará línea por línea este script.

```
newton # cd /usr/sbin

### De las siguientes dos líneas la primera crea tres réplicas en la partición  c0t0d0s7, y la
### segunda  crea otras tres réplicas en la partición c1t0d0s7.

metadb -a -f -c 3 c0t0d0s7
metadb -a -f -c 3 c1t0d0s7

### De las siguientes tres líneas la primera indica la creación de un subespejo llamado d10 con
la partición c0t0d0s0,  ###  la segunda línea indica la creación de un segundo subespejo llamado
d20 con la partición c1t0d0s0, y la tercer ### línea inicializa el metaespejo d0 utilizando el
subespejo d10 que corresponde al sistema de archivos / de sistema
### operativo.

metainit -f d10 1 1 c0t0d0s0
metainit -f d20 1 1 c1t0d0s0
metainit d0 -m d10

### De las siguientes tres líneas la primera indica la creación de un subespejo llamado d11 con
la partición c0t0d0s1,  ### la segunda línea indica la creación de un segundo subespejo llamado
d21 con la partición c1t0d0s1, y la tercer ### línea inicializa el metaespejo d1 utilizando el
subespejo d11 que corresponde al swap del sistema
```

```
### operativo.

metainit -f d11 1 1 c0t0d0s1
metainit -f d21 1 1 c1t0d0s1
metainit d1 -m d11

### De las siguientes tres líneas la primera indica la creación de un subespejo llamado d12 con
la partición c0t0d0s3,  ### la segunda línea indica la creación de un segundo subespejo llamado
d22 con la partición c1t0d0s3, y la tercer ### línea inicializa el metaespejo d2 utilizando el
subespejo d12 que corresponde al sistema de archivos /usr de
### sistema operativo.

metainit -f d12 1 1 c0t0d0s3
metainit -f d22 1 1 c1t0d0s3
metainit d2 -m d12


### De las siguientes tres líneas la primera indica la creación de un subespejo llamado d13 con la partición c0t0d0s4, ### la segunda línea indica
la creación de un segundo subespejo llamado d23 con la partición c1t0d0s4, y la tercer ### línea inicializa el metaespejo d3 utilizando el
subespejo d13 que corresponde al sistema de archivos /var de
### sistema operativo.

metainit -f d13 1 1 c0t0d0s4
metainit -f d23 1 1 c1t0d0s4
metainit d3 -m d13

### De las siguientes tres líneas la primera indica la creación de un subespejo llamado d14 con
la partición c0t0d0s5,  ### la segunda línea indica la creación de un segundo subespejo llamado
d24 con la partición c1t0d0s5, y la tercer  ### línea inicializa el metaespejo d4 utilizando el
subespejo d14 que corresponde al sistema de archivos /opt de
### sistema operativo.

metainit -f d14 1 1 c0t0d0s5
metainit -f d24 1 1 c1t0d0s5
metainit d4 -m d14

### La partición donde están guardadas las copias de las réplicas de Disk Suite no se espejea.
### Con el siguiente comando se modifica el archivo de configuración /etc/vfstab para que la
siguiente vez el
### servidor inicalize de metadispositivos de disk Suite y no de particiones de disco simples.

metaroot d0
```

3c. Verificar que los metadispositivos y las réplicas de DiskSuite fueron creados correctamente

```
### Debemos tener 3 réplicas en la partición c0t0d0s7 y 3 mas en la partición c1t0d0s7, ya que
fueron las que
### debieron de haberse creado con la segunda línea del script anterior, usemos el siguiente
comando:

newton # metadb -i

        flags           first blk       block count
     a m  p  luo        16              1034            /dev/dsk/c0t0d0s7
     a    p  luo        1050            1034            /dev/dsk/c0t0d0s7
     a    p  luo        2084            1034            /dev/dsk/c0t0d0s7
     a    p  luo        16              1034              /dev/dsk/c1t0d0s7
     a    p  luo        1050            1034            /dev/dsk/c1t0d0s7
     a    p  luo        2084            1034            /dev/dsk/c1t0d0s7
```

```
o - replica active prior to last mddb configuration change
u - replica is up to date
l - locator for this replica was read successfully
c - replica's location was in /etc/lvm/mddb.cf
p - replica's location was patched in kernel
m - replica is master, this is replica selected as input
W - replica has device write errors
a - replica is active, commits are occurring to this replica
M - replica had problem with master blocks
D - replica had problem with data blocks
F - replica had format problems
S - replica is too small to hold current data base
R - replica had device read errors


# Para comprobar que los metaespejos se hayan hecho correctamente usemos el comando siguiente:

newton # metastat -p

d0 -m d10 d20 1       ### Indica que el metaespejo d0 está formado por los subespejos d10 y d20
d10 1 1 c0t0d0s0      ### Indica que el suespejo d10 está formado por  c1t0d0s0 y pertenece a d0
d1 -m d11 d21 1       ### Indica que el metaespejo d1 está formado por los subespejos d11 y d21
d11 1 1 c0t0d0s1      ### Indica que el suespejo d11 está formado por  c1t0d0s1 y pertenece a d1
d2 -m d12 d22 1       ### Indica que el metaespejo d2 está formado por los subespejos d12 y d22
d12 1 1 c0t0d0s3      ### Indica que el suespejo d12 está formado por  c1t0d0s3 y pertenece a d2
d3 -m d13 d23 1       ### Indica que el metaespejo d3 está formado por los subespejos d13 y d23
d13 1 1 c0t0d0s4      ### Indica que el suespejo d13 está formado por  c1t0d0s4 y pertenece a d3
d4 -m d14 d24 1       ### Indica que el metaespejo d4 está formado por los subespejos d14 y d24
d14 1 1 c0t0d0s5      ### Indica que el suespejo d14 está formado por  c1t0d0s5 y pertenece a d4

d20 1 1 c1t0d0s0     ### Las 4 líneas de la izquierda muestran los subespejos que no están
sincronizados
d22 1 1 c1t0d0s3     ### con su respectivo metaespejo.
d23 1 1 c1t0d0s4
d24 1 1 c1t0d0s5
```

4. Editar el archivo /etc/vfstab para incluir las rutas de los metadispositivos y comentar las de los dispositivos fisicos.

```
### Se recomienda respaldar este archivo y editar el archivo original, dando de alta los
metadispositivos. Al finalizar ### el archivo deberá de quedar de la forma siguiente:

* Las letras en negritas son las líneas que se agregan al editar el vfstab

#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
#/dev/md/dsk/d0 /dev/md/rdsk/d0 /         ufs     1       yes     - ### Esta línea ya se había
agregado
fd      -       /dev/fd fd      -       no      -
/proc   -       /proc   proc    -       no      -
/dev/md/dsk/d0  /dev/md/rdsk/d0 /       ufs     1       no      -
swap    -       /tmp    tmpfs   -       yes     -
/dev/md/dsk/d1  -       -       swap    -       no      -
/dev/md/dsk/d2  /dev/md/rdsk/d2 /usr    ufs     1       no      -
/dev/md/dsk/d3  /dev/md/rdsk/d3 /var    ufs     1       no      -
/dev/md/dsk/d4  /dev/md/rdsk/d4 /opt    ufs     2       yes     -

### Este procedimiento requiere que se reinicialize el sistema.
```

```
newton # sync          ### Sincroniza los datos de memoria a disco
newton # sync          ### Sincroniza los datos de memoria a disco
newton # init 6         ### Reinicializa el sistema
```

5. Por último, una vez que el sistema está funcionando nuevamente, es necesario asociar los subespejos que están desasociados a sus metaespejos correspondientes para que la información se comience a sincronizar. Para ello se emplea el script siguiente.

```
### Con las siguientes lineas de comandos lo que estamos haciendo es asociar cada subespejo que
estaba
### desasociado  con su subespejo y metaespejo correspondiente para  iniciar la sincronización de
sus datos.

metattach d0 d20
metattach d1 d21
metattach d2 d22
metattach d3 d23
metattach d4 d24

### Verificar el avance de la sincronicación

# metastat |grep -i progress

### Cuando la sincronozación haya terminado el proceso de instalación y configuración de Solstice
DiskSuite estará
### concluido.
```

De esta manera ya tenemos dos discos con el mismo sistema operativo, uno en la posición 0 del arreglo yodo y el otro en la posición 0 del arreglo yodo2. La gráfica siguiente muestra como quedaron los metaespejos del sistema operativo del servidor web de la UNAM.

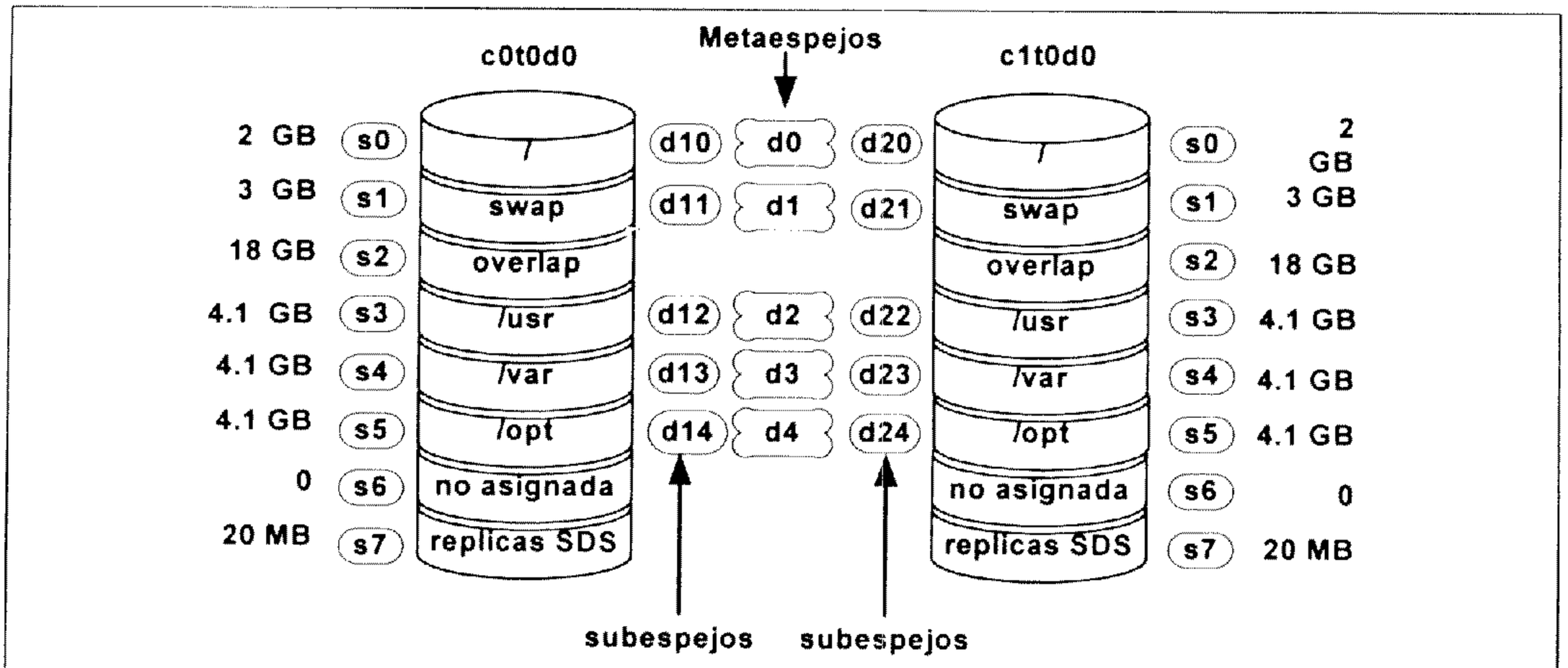


Figura 4.6. Metaespejos de disk suite de sistema operativo de newton

Instalación y configuración de Veritas Volume Manager y Veritas File System

El siguiente paso es la instalación y configuración de Veritas Volume Manager y veritas File System. El objetivo de esto es que para los discos de datos de las aplicaciones y usuarios tengamos una configuración RAID 0+1 igual a la del servidor dragón, aunque debemos señalar que solo tendremos un plex para cada volumen ya que no tenemos discos suficientes para formar volumenes con dos plexes. El plex que tendrá cada volumen tendrá una distribución de banda como en dragón.

Antes que nada haremos otro respaldo a cinta del sistema operativo, ya teníamos un respaldo pero este no incluye todo lo que se hizo a partir de la instalación de Disk Suite, por lo tanto haremos otro. También desasociaremos los subespejos d20, d21, d22,d23 y d24 de sus respectivos metaespejos para que los nuevos cambios de configuración solo se hagan en los subespejos d10,d11,d12,d13 y d14, y de esta manera tener un segundo respaldo de sistema operativo en disco como medida de contingencia.

Se hizo un movimiento en la posición de los discos del arreglo newton, se movió el disco de la posición 6 de la parte trasera a la posición 5, de tal forma que los discos quedaron distribuidos como se muestra a continuación.

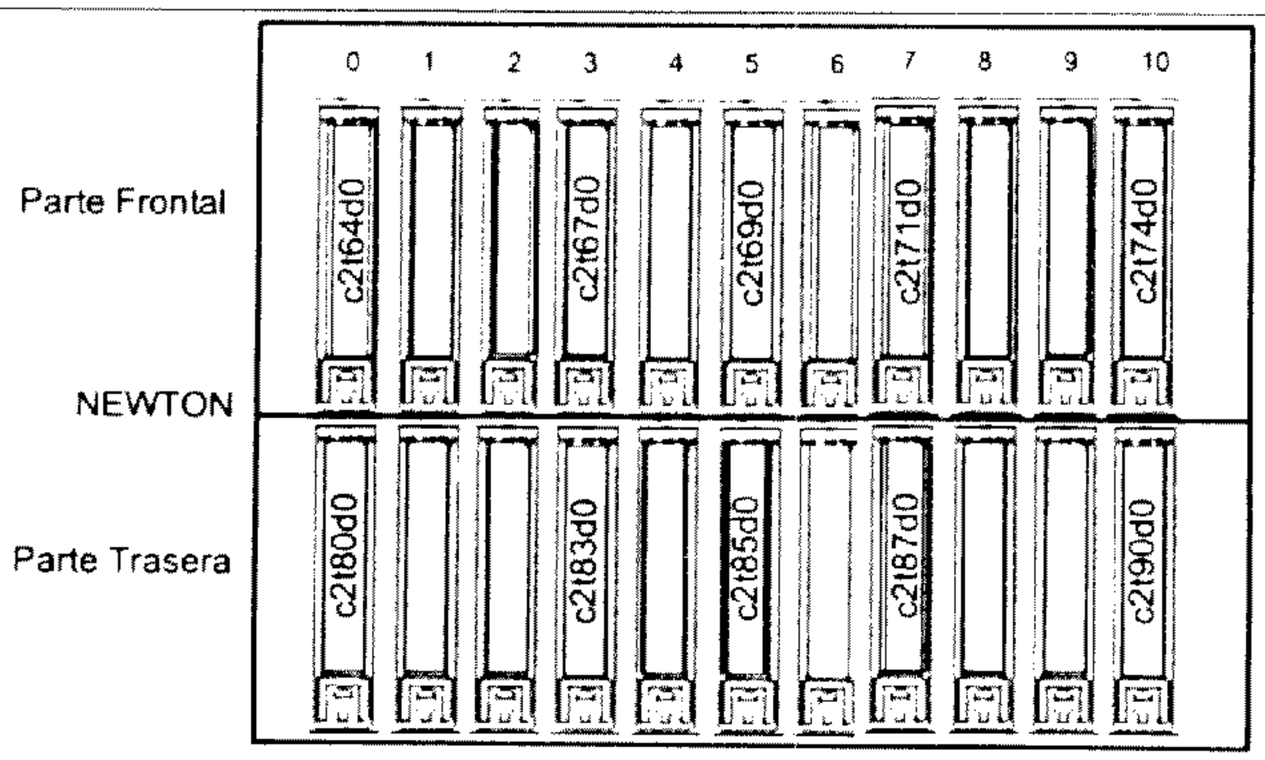


Figura 4.7. Discos del arreglo newton del servidor newton

Los discos que utilizaremos para volume manager son los de la posición 0,3,5,7, y 10 de la parte frontal, y los de las posiciones 0,3,5,7 y 10 de la parte trasera. Solo se crearán los volumenes con un solo plex.

Sigamos el siguiente procedimiento para instalar y configurar Veritas Volume Manager versión 3.0.4., este procedimiento incluye también la instalación de Veritas File Systems aunque respecto a este último no hay que hacer grandes cambios, solo hay que instalar su paquete correspondiente y comenzar a usarlo para crear sistemas de archivos.

1.- Instalación de los paquetes de Veritas.

Una vez descomprimido el directorio que contiene los paquetes de veritas, pasarse a él e instalarlos, estos paquetes contienes los archivos binarios, las páginas de manual, documentación y ambiente gráfico.

Las partes de la instalación de los paquetes y de la instalación de los parches las podemos excluir, ya que son exactamente igual a la del capítulo 3, solo mencionamos sus puntos correspondientes.

2.- Instalación de los parches.

3.- Una vez que el servidor está funcionando nuevamente se procede con la configuración de volume manager.

```
### Inicializar el proceso o de configuración de volume manager  vxconfigd, el cual mantiene
configuraciones de
### discos de volume manger y de sus grupos de discos.

newton# vxconfigd

###  Crear el archivo /etc/vx/volboot, este archivo contiene un identificador del servidor que es
utilizado por volume ### manager  para establecer a quién perpetencen los discos físicos.

newton # vxdctl init

### Agregar licencias de volume manager y file system. El agregar una licencia de veritas implica
introducir un
### número de varias cifras lo cual nos va a permitir hacer uso de este software de manera legal.
En este caso por
### razones de seguridad este número de licencia lo denotaremos con "#".

### Para Veritas Volume Manager

newton # vxlicense -c
...
...
#### #### #### #### #### ####
...
```

```
### Para Veritas File System

newton # vxlicense -c
...
...
#### #### #### #### #### ####
...

###  Crear el grupo de discos rootdg. Antes de crear cualquier grupo de discos primero debe
inicializarse el grupo ### rootdg  ya que es indispensable para la configuración de volume
manager.

newton# vxdg init rootdg

### Inicializar el disco rootdg01 y agregarlo al grupo rootdg, en este caso el disco que usaremos
para el grupo rootdg ### es el c1t20d0 del arreglo de discos yodo.

newton # vxdisksetup -i c2t85d0
newton # vxdisk -f init c2t85d0s2
newton # vxdg -g rootdg adddisk rootdg01=c2t85d0s2

### Habilitar el manejador de discos Volume Manager

newton # vxdctl enable

### Remover archivo install-db. Este archivo se crea automáticamente cuando se instala volume
manager, y al estar ### creado lo que hace es evitar que volume manager inicialize cuando el
servidor inicia.

newton # cd /etc/vx/reconfig.d/state.d
newton # rm install-db

### En este caso no se configurará el espejo del disco del grupo rootdg por falta de discos.

#### Se reinicializa el servidor dragón  para verificar que volume manager funcione de manera
#### automática y de forma adecuada.
```

4.- Creación de volúmenes y sistemas de archivos del grupo rootdg.

El único sistema de archivos que vamos a tener sobre el grupo rootdg es el de /respaldos que nos servirá para guardar respaldos de sistema operativo, como recordarán la creación de este sistema de archivos a la cual se hace referencia en el capítulo 2.
Crearemos el volumen con un plex sobre el disco rootdg01.

```
### Crear el volumen respaldos y su sistema de archivos; crear punto de montura y montarlo.

newton # vxassist -g rootdg make repaldos 17g rootdg01
newton # mkfs -F vxfs -o  largefiles,bsize=8192 /dev/vx/rsdk/rootdg/respaldos
newton # fsck -F vxfs /dev/vx/rdsk/rootdg/respaldos
newton # mkdir /respaldos
newton # mount -F vxfs /dev/vx/dsk/rootdg/respaldos /respaldos

### Este plex de volumen no tendrá espejo.
```

5.- Configuración del grupo de discos para datos de aplicaciones y usuarios.

Necesitamos 148 GB de capacidad en el arreglo de discos para recrear la estructura de sistemas de archivos de datos de dragón. Ese espacio lo cubrimos con los 9 discos de 18 GB que tenemos en el arreglo newton descontando el disco que se usó para crear el grupo rootdg. Como mencionamos en este capítulo anteriormente los discos que utilizaremos para volume manager son los de la posición 0,3,5,7, y 10 de la parte frontal, y los de las posiciones 0,3,5,7 y 10 de la parte trasera y solo se crearán los volumenes con un solo plex.

Sigamos los siguientes pasos para la creación de los volúmenes de las aplicaciones.

Primero creamos e inicializamos el grupo de discos de las aplicaciones. Al igual que en dragón decidimos nombrar al grupo como "datosdg" para hacer referencia a la información que tendrá. Con el propósito de que el nombre del disco de volume manager fuera ilustrativo, decidimos nombrar a cada disco empezando por el nombre del arreglo, después una "f" si el disco está en la parte frontal de la caja o una "t" si el disco está en la parte trasera, después le sigue un número que indica la posición del disco en la caja. Obviamente este nombre de disco hace referencia al nombre físico del disco como se vio en la parte teórica de volume manager.

```
###Con la siguiente linea inicializamos el grupo datosdg con el disco "newtonf0".

# vxdg init datosdg newtonf0=c2t64d0
```

El siguiente paso es inicializar cada disco físico y agregarlos al grupo de discos datosdg. Para esto hicimos dos scripts y los ejecutamos. El disco c2t64d0 ya no se toma en cuenta porque ya fue puesto bajo el control de volume manager en el paso anterior.

```
### Script que inicializa los discos del arreglo yodo

/etc/vx/bin/vxdisksetup -i c2t67d0
/etc/vx/bin/vxdisksetup -i c2t69d0
/etc/vx/bin/vxdisksetup -i c2t71d0
/etc/vx/bin/vxdisksetup -i c2t74d0
/etc/vx/bin/vxdisksetup -i c2t80d0
/etc/vx/bin/vxdisksetup -i c2t83d0
/etc/vx/bin/vxdisksetup -i c2t87d0
/etc/vx/bin/vxdisksetup -i c2t90d0

### Script que nombra  los discos del arreglo yodo que usará el grupo datosdg

/etc/vx/bin/vxdg -g datosdg adddisk newtonf3= c2t67d0
/etc/vx/bin/vxdg -g datosdg adddisk newtonf5= c2t69d0
/etc/vx/bin/vxdg -g datosdg adddisk newtonf7= c2t71d0
```

```
/etc/vx/bin/vxdg -g datosdg adddisk newtonf10=c2t74d0
/etc/vx/bin/vxdg -g datosdg adddisk newtont0= c2t80d0
/etc/vx/bin/vxdg -g datosdg adddisk newtont3= c2t83d0
/etc/vx/bin/vxdg -g datosdg adddisk newtont7= c2t87d0
/etc/vx/bin/vxdg -g datosdg adddisk newtont10=c2t90d0
```

6.- Creación de volúmenes del grupo datosdg.

Ya que tenemos dados de alta los discos con sus nombres respectivos, creamos los volúmenes con un plex. También aquí utilizamos un pequeño script.

```
### Script que crea los volúmenes con su primer plex

 vxassist -g datosdg make users00 7g  layout=stripe ncolumn=9 newtonf0 newton3 newtonf5 newtonf7
newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make users01 7g  layout=stripe ncolumn=9 newtonf0 newton3 newtonf5 newtonf7
newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make users02 7g  layout=stripe ncolumn=9 newtonf0 newton3 newtonf5 newtonf7
newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make users03 7g  layout=stripe ncolumn=9 newtonf0 newton3 newtonf5 newtonf7
newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make users04 4g  layout=stripe ncolumn=9 newtonf0 newton3 newtonf5 newtonf7
newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make home   31g layout=stripe ncolumn=9 newtonf0 newton3 newtonf5 newtonf7
newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make mirrorhome   31g layout=stripe ncolumn=9 newtonf0 newton3 newtonf5
newtonf7 newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make mirrorusers00 7g  layout=stripe ncolumn=9 newtonf0 newton3 newtonf5
newtonf7 newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make homelog 31g layout=stripe ncolumn=9 newtonf0 newton3 newtonf5 newtonf7
newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make sybase   7g  layout=stripe ncolumn=9 newtonf0 newton3 newtonf5 newtonf7
newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make raid   4g layout=stripe ncolumn=9 newtonf0 newton3 newtonf5 newtonf7
newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make raid2   4g layout=stripe ncolumn=9 newtonf0 newton3 newtonf5 newtonf7
newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make pgsql   2g layout=stripe ncolumn=9 newtonf0 newton3 newtonf5 newtonf7
newtonf10 newtont0 newtont3 newtont7 newtont10
 vxassist -g datosdg make usrsybase   5g layout=stripe ncolumn=9 newtonf0 newton3 newtonf5
newtonf7 newtonf10 newtont0 newtont3 newtont7 newtont10
```

7.- Creación de los sistemas de archivos de los volúmenes.

Una vez creados los volúmenes es necesario crear sus correspondientes sistemas de archivos, es aquí donde se hace uso del software instalado Veritas File System. Veamos el script correspondiente.

```
### Este ciclo for de shell de unix, crea un sistema de archivos para cada volumen que creamos en
el punto ### anterior, nótese que no se está especificando el tamaño de cada sistema de archivos,
debido a que toma el ### tamaño de cada volumen. También en este ciclo le estamos indicando que
los sistemas de archivo van a ### ser de tipo  vxfs, es decir de veritas file system, y además
que van a soportar tamaños largos de archivos.

for i in users00 users01 users02 users03 users04 home mirrorhome mirrorusers00 homelog sybase
raid raid2 pgsql usrsybase homenewton
> do
> mkfs -F vxfs -o  largefiles,bsize=8192 /dev/vx/rsdk/datosdg/$i
```

8.- Creación de los puntos de montura y montado de sistemas de archivo.

```
### El siguiente script crea los puntos de montura para cada sistema de archivos. Es importante
que estos
### puntos de montura tengan el nombre que tenían originalmente

mkdir /home/users00
mkdir /home/users01
mkdir /home/users02
mkdir /home/users03
mkdir /home/users04
mkdir  /mirror/home
mkdir /mirror/users00
mkdir /home
mkdir /home/log
mkdir /sybase
mkdir /raid
mkdir /raid2
mkdir /usr/local/pgsql
mkdir /usr/sybase

### Para montar los sistemas de archivos se utilzó el siguiente script.

mount -F vxfs /dev/vx/dsk/datosdg/users00 /home/users00
mount -F vxfs /dev/vx/dsk/datosdg/users01 /home/users01
mount -F vxfs /dev/vx/dsk/datosdg/users02 /home/users02
mount -F vxfs /dev/vx/dsk/datosdg/users03 /home/users03
mount -F vxfs /dev/vx/dsk/datosdg/users04 /home/users04
mount -F vxfs /dev/vx/dsk/datosdg/home    /home
mount -F vxfs /dev/vx/dsk/datosdg/mirrorhome   /mirror/home
mount -F vxfs /dev/vx/dsk/datosdg/mirrorusers00   /mirror/users00
mount -F vxfs /dev/vx/dsk/datosdg/homelog /home/log
mount -F vxfs /dev/vx/dsk/datosdg/sybase /sybase
mount -F vxfs /dev/vx/dsk/datosdg/raid    /raid
mount -F vxfs /dev/vx/dsk/datosdg/raid2  /raid2
mount -F vxfs /dev/vx/dsk/datosdg/pgsql   /usr/local/pgsql
mount -F vxfs /dev/vx/dsk/datosdg/usrsybase   /usr/Sybase
```

9.- Automatizar el montado de sistemas de archivos.

Agregar los sistemas de archivos creados en su correspondiente volumen al archivo /etc/vfstab. Se agregan tanto los volúmenes del grupo rootdg como los del grupo datosdg. Esto se hace con la finalidad de que cuando el servidor reinicie monte de manera automática estos sistemas de archivos ya que este archivo de configuración es leído siempre que un servidor inicia su funcionamiento. Las letras en negritas son las lineas que se agregaron. Recordemos que anteriormente también se agregaron las líneas correspondientes a disk suite.

```
### En seguida se muestra el contenido del archivo /etc/vfstab


#device         device          mount           FS      fsck    mount   mount
#to mount       to fsck         point           type    pass    at boot options
#
fd      -       /dev/fd fd      -       no      -
swap    -       /tmp    tmpfs   -       yes     -
/proc   -       /proc   proc    -       no      -
/dev/md/dsk/d0  /dev/md/rdsk/d0 /       ufs     1       no      -
/dev/md/dsk/d1  -       -       swap    -       no      -
/dev/md/dsk/d2  /dev/md/rdsk/d2 /usr    ufs     1       no      -
/dev/md/dsk/d3  /dev/md/rdsk/d3 /var    ufs     1       no      -
/dev/md/dsk/d4  /dev/md/rdsk/d4 /opt    ufs     2       yes     -
/dev/vx/dsk/rootdg/respaldos    /dev/vx/rdsk/rootdg/respaldos   /respaldos vxfs    3     yes
-
/dev/vx/dsk/datosdg/users01 /dev/vx/rdsk/datadg/users01        /users01   vxfs    3       yes
-
/dev/vx/dsk/datosdg/users02 /dev/vx/rdsk/datadg/users02        /users02   vxfs    3       yes
-
/dev/vx/dsk/datosdg/users03 /dev/vx/rdsk/datadg/users03        /users03   vxfs    3       yes
-
/dev/vx/dsk/datosdg/users04 /dev/vx/rdsk/datadg/users04        /users04   vxfs    3       yes
-
/dev/vx/dsk/datosdg/home /dev/vx/rdsk/datadg/home       /home   vxfs    3       yes     -
/dev/vx/dsk/datosdg/mirrorhome /dev/vx/rdsk/datadg/mirrorhome       /mirror/home   vxfs    3
yes     -
/dev/vx/dsk/datosdg/mirrorusers00 /dev/vx/rdsk/datadg/mirrorusers00       /mirror/users00   vxfs
3       yes     -
/dev/vx/dsk/datosdg/homelog /dev/vx/rdsk/datadg/homelog        /home/log   vxfs    3       yes
-
/dev/vx/dsk/datosdg/sybase /dev/vx/rdsk/datadg/sybase       /sybase   vxfs    3       yes     -
/dev/vx/dsk/datosdg/raid /dev/vx/rdsk/datadg/raid       /raid   vxfs    3       yes     -
/dev/vx/dsk/datosdg/raid2 /dev/vx/rdsk/datadg/raid2       /raid2   vxfs    3       yes     -
/dev/vx/dsk/datosdg/pgsql /dev/vx/rdsk/datadg/pgsql       /usr/local/pgsql   vxfs    3       yes
-
/dev/vx/dsk/datosdg/usrsybase /dev/vx/rdsk/datadg/usrsybase       /usr/sybase   vxfs    3
yes     -
```

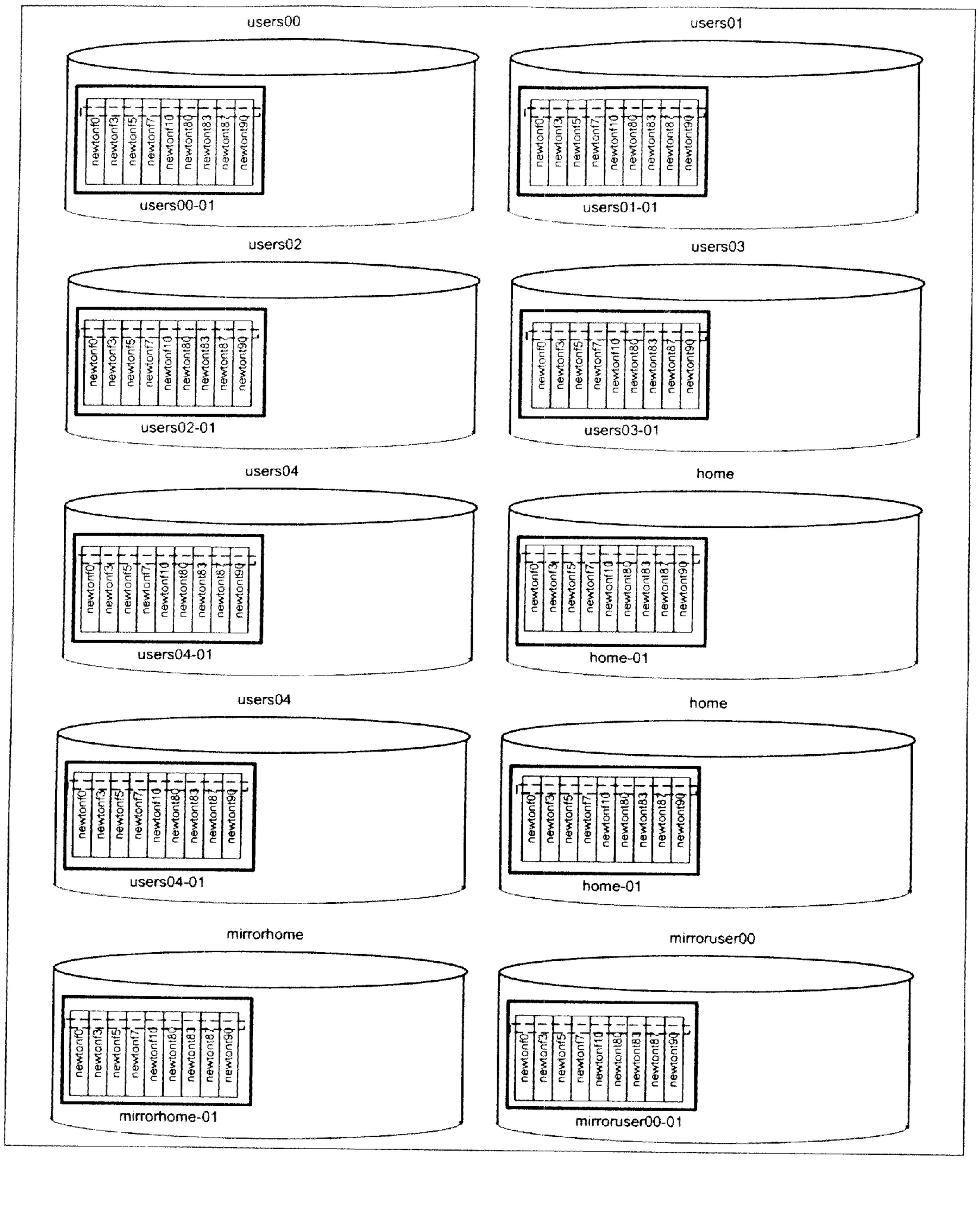
Hasta aquí, hemos creado los volúmenes con un plex en cada uno de ellos, y a su vez su correspondiente sistema de archivos y también los hemos montado, todo esto en el servidor newton. El siguiente paso es recuperar la información de las aplicaciones y de los usuarios, la idea era que finalmente esa información quedara en los discos del arreglo D1000 de dragón en el que estaba, así que transferimos la información vía ftp a newton utilizando temporalmente el sistema de archivos /home y el /homelog que se habían creado y que medían cada uno 31 GB, una vez que la transferencia acabó y que se verificó la integridad de la información, se desconectó el arreglo D1000 de dragón y se trasladó al centro de cómputo de Pitágoras, se conectó a newton a través de una tarjeta SCSI que no se estaba usando y se crearon metadispositivos correspondientes a /home, /home/users00, /home/users01, /oracle y /home a lo largo de sus siete discos y se movió allí la información previamente transferida a /home y /homelog, posteriormente estos dos últimos sistemas de archivos de vaciaron para que estuvieran disponibles para lo que fueron creados que es la replicación de datos.

Este procedimiento no lo describiremos, solo basta saber que nuevamente los datos de las aplicaciones y usuarios están disponibles en el servidor newton, de esta forma ya tenemos la estructura de sistemas de archivos igual a la de dragón en el servidor newton lista para usarse en la replicación de datos.

Configuración final de discos de aplicaciones y usuarios

De esta forma terminamos la configuración de volúmenes con el manejador de discos Veritas Volume Manager. Todos los volúmenes quedan con un plex el cual es del mismo tamaño que los plexes de los volúmenes de dragón.

La nueva configuración de los discos de las aplicaciones del servidor newton de la UNAM queda como se muestra a continuación.


users00
newtonf0
newtonf3
newtonf5
newtonf7
newtonf10
newton80
newton83
newton87
newton90
users00-01
users01
users01-01
users02
users02-01
users03
users03-01
users04
users04-01
home
home-01
users04
users04-01
home
home-01
mirrorhome
mirrorhome-01
mirroruser00
mirroruser00-01

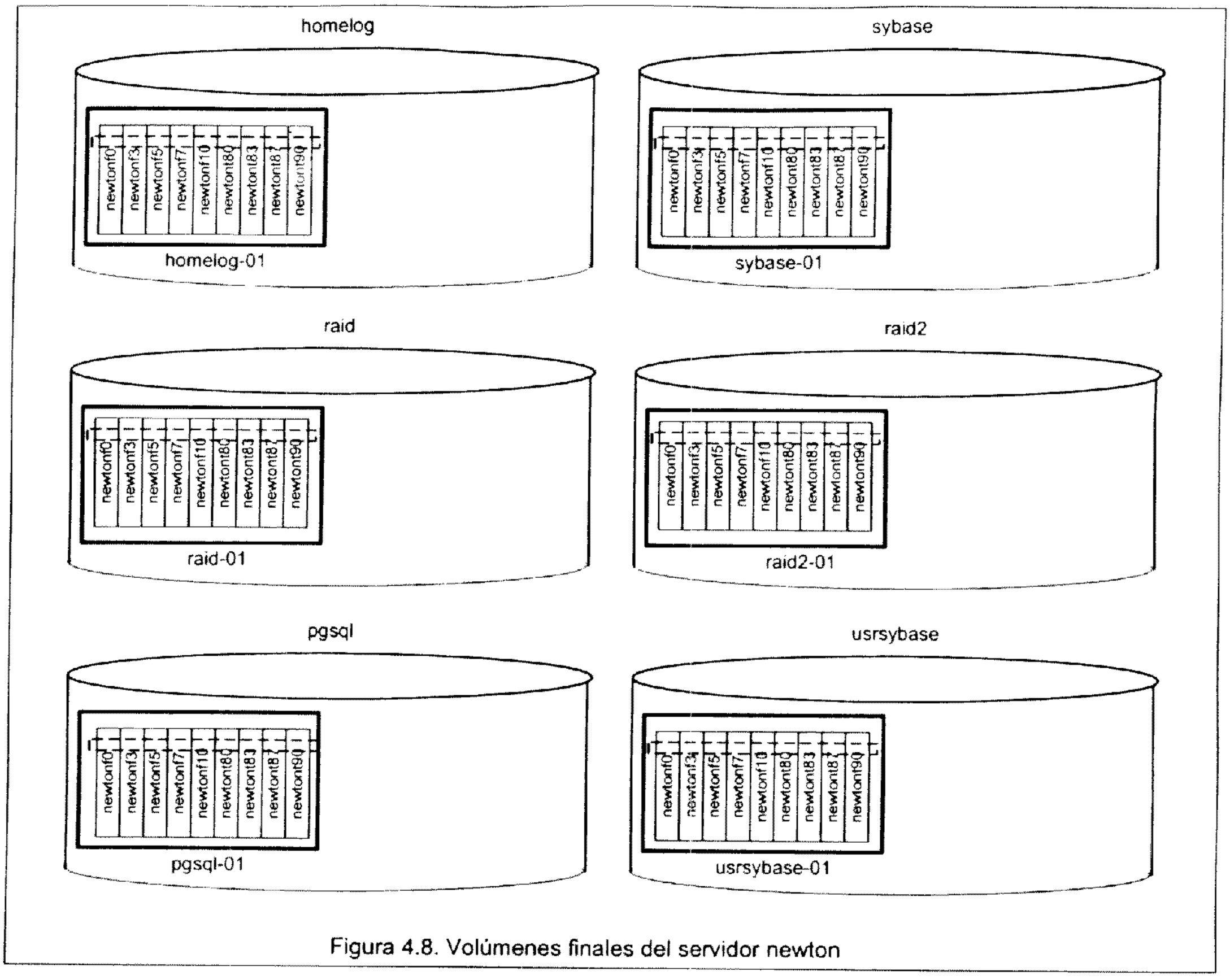


Figura 4.8. Volúmenes finales del servidor newton

Proceso para replicación de datos del web de la UNAM

Hoy día la pagina principal de una UNAM resulta sumamente importante para la comunidad universitaria, y mas aun si se toma en consideración que también las universidades pasan por épocas difíciles como las que sufrió nuestra universidad hace algunos años durante la huelga, el haber contado con un programa de DR en aquel entonces nos hubiera facilitado la migración de un servicio en cuestión de minutos. La técnica que se usará para replicar los datos es conocida como *distribución de archivos*; esta técnica se basa en utilizar aplicaciones que normalmente están incluidas en el sistema operativo en nuestro caso Solaris cuenta con varias aplicaciones las cuales pueden ayudarnos en nuestra tarea como ufsdump, tar, rdist , esta ultima es la mas apropiada para la universidad que nos permitirá replicar los datos de los servidores ubicados en la DGSCA a un site ubicado a varios kilómetros fuera de la universidad en la Colonia del Valle. Se opta por este tipo de replicación ya que es una replicación hasta cierto punto barata, con la cual la universidad no tendrá que pagar miles de dólares en licencias y soporte.

El comando *rdist* nos ayuda a sincronizar archivos ordinarios, directorios y archivos especiales y puede ser utilizado para sincronizar datos entre dos servidores que se encuentren en el mismo site o entre servidores que se encuentran a miles de kilómetros; si por alguna razón el servidor primario sufre alguna falla los datos sincronizados en el servidor secundario puede continuar dando el servicio, esta aplicación tiene la ventaja que es una herramienta de replicación bidireccional, esto quiere decir que detecta cambios en ambos servidores. Para estos fines es sumamente importante la configuración a nivel de red del ambos servidores ya que de esto depende el tiempo de sincronización de los datos, los datos que necesitamos son las salidas del comando ifconfig y un cat del archivo /etc/hosts de ambos servidores.

Configuración a nivel de red de dragon

dragon # ifconfig -a

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 132.248.10.7 netmask ffffff00 broadcast 132.248.10.255
        ether 0:0:be:a6:6b:41
```

dragon # cat /etc/hosts

```
#
# Internet host table
#
127.0.0.1       localhost
132.248.10.7    dragon.dgsca.unam.mx   dragon  loghost
132.248.10.15   dragon-ssp

#
# CST
#
132.248.115.7   plata   middleware
```

Configuración a nivel de red de newton

newton # ifconfig -a

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
qfe0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 132.247.12.12 netmask ffffff00 broadcast 132.247.12.255
        ether 0:0:be:a7:0:3a
```

newton # cat /etc/hosts

```
# Internet host table
#
127.0.0.1       localhost
132.247.12.12   newton  newton.servidores.unam.mx      loghost
132.247.12.10   pitagoras1
132.248.10.1    servidor.unam.mx
132.248.255.191         telecom1

#
# CST
#
132.248.115.7    plata   middleware
```

Si observamos la configuración actual el servidor dragon esta dando servicio por un interfaz hme0 conectada a la red 132.248.10.0, este equipo cuenta con una tarjeta qfe (quad fast ethernet), la cual nos permite tener 4 interfaces de red que trabajan a 100 Mbs. Lo que se hará es tener acceso a la red 132.247.12.0 vía una interfaz qfe, esto con el fin de obtener un mejor performance, ya que toda la replicación se hará por medio de esta interfase y no por la red de producción que es la 132.248.10.0, también se ocupará otra interfaz qfe en el servidor newton con el fin de no saturar la interfaz ya existente, todo este proceso se realizo de la manera siguiente:

Reconfiguración de interfaces de dragon

dragon # ifconfig -a

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 132.248.10.7 netmask ffffff00 broadcast 132.248.10.255
        ether 0:0:be:a6:6b:41
```

1. Con el comando ifconfig se da de alta la nueva interfaz, en este caso la qfe3 y se le asigna la dirección ip 132.248.12.10

```
dragon # ifconfig qfe3 plumb
dragon # ifconfig qfe3 inet 132.247.12.10 netmask 255.255.255.0 broadcast + up
dragon # ifconfig -a

lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 132.248.10.7 netmask ffffff00 broadcast 132.248.10.255
        ether 0:0:be:a6:6b:41
qfe3: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 132.248.12.10 netmask fffff00 broadcast 132.248.12.255
        ether 0:0:be:a6:6c:3c
```

2. Como el comando ifconfig habilita la interfaz para que sea vista por el kernel, en el momento en que reinicialize el servidor esta configuración se perderá, por lo cual debemos habilitar la interfaz vía archivos de configuración, esto lo hacemos editando el archivo /etc/hosts y /etc/hostaname.qfe3 (este archivo no existe, se debe crear).

dragon # vi /etc/hosts

```
#
# Internet host table
#
127.0.0.1       localhost
132.248.10.7    dragon.dgsca.unam.mx  dragon loghost
132.248.10.15   dragon-ssp

# Linea agregada para interfaz qfe3 - Replicacion de Datos -
132.247.12.10   dragon-replica
132.148.12.16   newton-replica

#
# CST
#
132.248.115.7   plata   middleware
```

dragon # vi /etc/hostname.qfe3

```
dragon-replica
```

Reconfiguración de interfaces de newton

newton # ifconfig -a

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
qfe0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 132.247.12.12 netmask ffffff00 broadcast 132.247.12.255
        ether 0:0:be:a7:0:3a
```

1. Damos de alta una interfaz extra con el fin de no saturar las interfaces de producción y que esta interfaz solo sirva para replicar datos, en este caso por cuestiones de administración ocupamos la interfaz qfe3.

```
newton # ifconfig qfe3 plumb
newton # ifconfig qfe3 inet 132.247.12.16 netmask 255.255.255.0 broadcast + up
newton # ifconfig -a

lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
        inet 127.0.0.1 netmask ff000000
qfe0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 132.247.12.12 netmask ffffff00 broadcast 132.247.12.255
        ether 0:0:be:a7:0:3a
qfe3: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
        inet 132.247.12.16 netmask ffffff00 broadcast 132.247.12.255
        ether 0:0:be:a7:0:3c
```

2. Actualizamos los archivos /etc/hosts y creamos el archivos /etc/hostname.qfe3 con el fin de hacer la configuración permanente.

newton # vi /etc/hosts

```
# Internet host table
#
127.0.0.1       localhost
132.247.12.12   newton  newton.servidores.unam.mx      loghost
132.247.12.10   pitagoras1
132.248.10.1    servidor.unam.mx
telecom1
132.247.12.10   dragon-replica


# Linea agregada para interfaz qfe3 - Replicacion de Datos -
132.247.12.16   newton-replica

#
# CST
#
132.248.115.7    plata   middleware
```

newton # vi /etc/hostname.qfe3

```
newton-replica
```

Con lo realizado hasta ahora ya contamos con dos interfaces hacia la misma red que nos permitirán replicar los datos mediante interfaces dedicadas, gráficamente tenemos algo como lo siguiente:

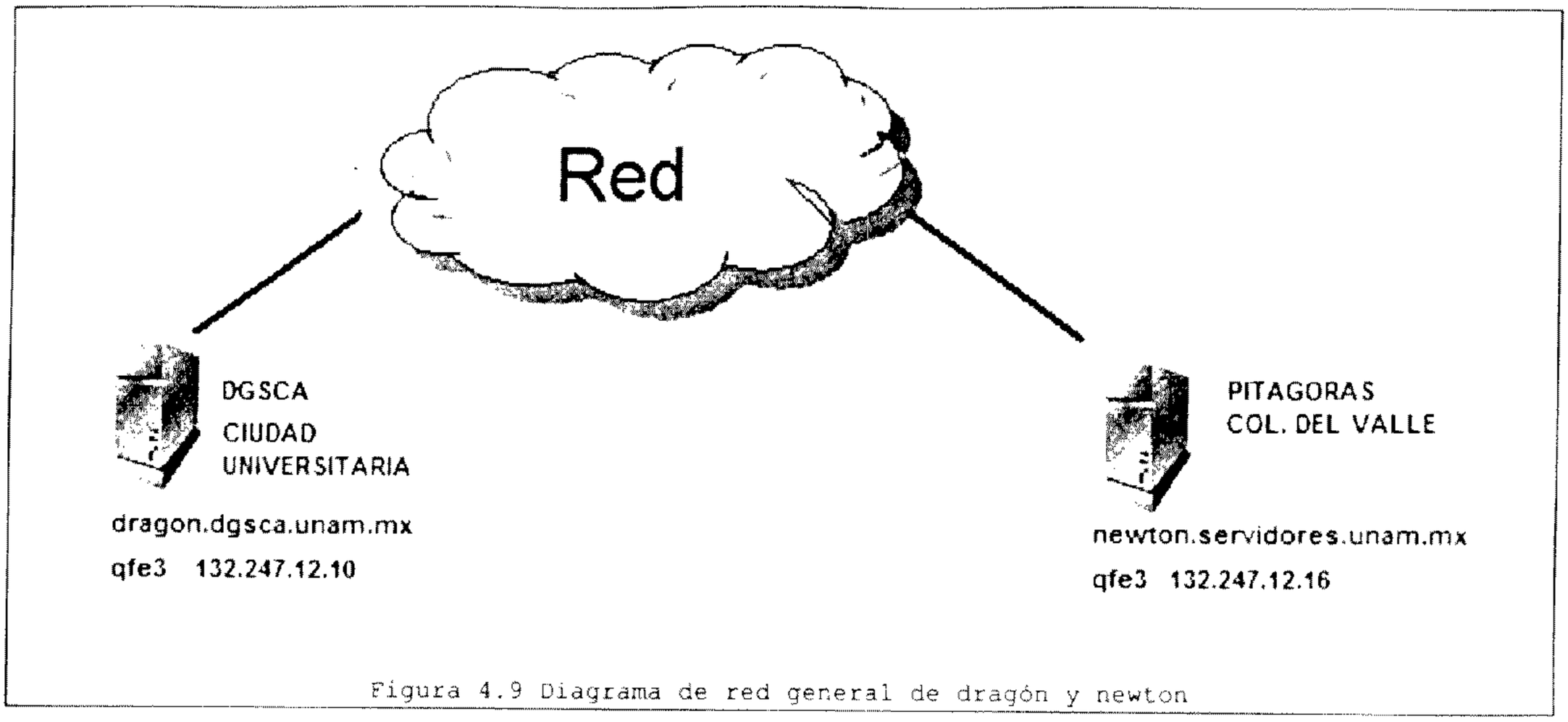


Figura 4.9 Diagrama de red general de dragón y newton

Ya que tenemos lo necesario a nivel de red para tener un buen desempeño en la sincronización lo único que nos hace falta es un script que nos permita realizar la replicación automáticamente, como lo habíamos mencionado antes utilizaremos la herramienta *rdist*, con lo cual estaremos utilizando una solución de distribución de archivos, los sistemas de archivos que se tienen que replicar en el servidor newton son los siguientes:

dragon # df -k

```
Filesystem kbytes used avail capacity Mounted on
/dev/md/dsk/d0 617275 327903 233818 59% /
/dev/md/dsk/d2 4131866 2646600 1443948 65% /usr
/proc 0 0 0 0% /proc
fd 0 0 0 0% /dev/fd
/dev/md/dsk/d3 3099287 2162276 875026 72% /var
/dev/md/dsk/d4 2056211 466484 1528041 24% /opt
swap 3034584 152 3034432 1% /tmp
/dev/vx/datosdg/dsk/mirrorhome 30945914 27291825 3344630 90% /mirror/home
/dev/vx/datosdg/dsk/homelog 30945914 12978909 17657546 43% /home/log
/dev/vx/datosdg/dsk/users00 6186810 4193040 1931902 69% /mirror/users00
/dev/vx/datosdg/dsk/users01 6186810 3778354 2346588 62% /home/users01
/dev/vx/datosdg/dsk/users02 6186810 3492786 2632156 58% /home/users02
/dev/vx/datosdg/dsk/users03 6186810 5857917 267025 96% /home/users03
/dev/vx/datosdg/dsk/users04 3871954 2069039 1764196 54% /home/users04
/dev/vx/datosdg/dsk/sybase 6186810 2881346 3243596 48% /sybase
/dev/vx/datosdg/dsk/raid 3871954 1961986 1871249 52% /raid
/dev/vx/datosdg/dsk/raid2 3871954 1100983 2732252 29% /raid2
/dev/vx/datosdg/dsk/pgsql 1527116 1140770 325262 78% /usr/local/pgsql
/dev/vx/datosdg/dsk/sybase 4743974 3615240 1081295 77% /usr/sybase
/dev/vx/datosdg/dsk/home 30957590 28151616 2496399 92% /home
/dev/vx/datosdg/dsk/users00 6199998 3354000 2783999 55% /home/users00
```

Antes de ocupar la herramienta rdist es necesario contar con los siguientes requerimientos:

1. El usuario con el que se va a replicar la información debe existir en ambos servidores, en este caso se ocupa el usuario root, así mismo se debe crear un archivo en llamado /.rhosts en el servidor que va a recibir los datos en nuestro caso el servidor se llama newton y dentro del archivo se pone el nombre del servidor dragon por la interfaz qfe3 (dragon-replica).

newton # vi /.rhosts

```
dragon-replica
```

2. Se verifica que el servicio de shell este habilitado en el archivo /etc/inetd.conf, en nuestro caso el servicio esta habilitado.

newton # cat /etc/inetd.conf

```
# Shell, login, exec, comsat and talk are BSD protocols.
#
shell    stream  tcp     nowait  root    /usr/sbin/in.rshd       in.rshd
login    stream  tcp     nowait  root    /usr/sbin/in.rlogind    in.rlogind
exec     stream  tcp     nowait  root    /usr/sbin/in.rexecd     in.rexecd
comsat   dgram   udp     wait    root    /usr/sbin/in.comsat     in.comsat
talk     dgram   udp     wait    root    /usr/sbin/in.talkd      in.talkd
#
```

3. Se realizan las pruebas de conectividad desde dragon utilizando rsh, la prueba consiste en hacer un remote shell a la maquina newton-replica preguntando por la fecha.

dragon # rsh newton-replica date

```
Mon Jul  9 13:50:40 CST 2003
```

Hasta este punto ya prácticamente esta todo listo para hacer la replicación de los datos, solo nos hace falta hacer un script con el cual se copiaran todos los del servidor dragon a newton, debemos recordar que está copia se realizará con el comando rdist que viene con el sistema operativo solaris, estos scripts son los siguientes:

```
#
#  Script para replicacion del sistema  de archivos /home
```

```
#

HOSTS = ( Newton-replica )
FILES = ( /home )

${FILES} -> ${HOSTS}
install;

#
#  Script para replicacion del sistema  de archivos /home/log
#

HOSTS = ( Newton-replica )
FILES = ( /home/log )

${FILES} -> ${HOSTS}
install;

#
#  Script para replicacion del sistema  de archivos /home/users00
#

HOSTS = ( Newton-replica )
FILES = ( /home/users00 )

${FILES} -> ${HOSTS}
install;

#
#  Script para replicacion del sistema  de archivos /home/users01
#

HOSTS = ( Newton-replica )
FILES = ( /home/users01 )

${FILES} -> ${HOSTS}
install;

#
#  Script para replicacion del sistema  de archivos /home/users02
#

HOSTS = ( Newton-replica )
FILES = ( /home/users02 )

${FILES} -> ${HOSTS}
install;

#
#  Script para replicacion del sistema  de archivos /home/users03
#

HOSTS = ( Newton-replica )
FILES = ( /home/users03 )

${FILES} -> ${HOSTS}
install;

#
#  Script para replicacion del sistema  de archivos /home/users04
#

HOSTS = ( Newton-replica )
FILES = ( /home/users04 )
```

```
${FILES} -> ${HOSTS}
install;

#
#  Script para replicacion del sistema  de archivos /usr/sybase
#

HOSTS = ( Newton-replica )
FILES = ( /usr/Sybase )

${FILES} -> ${HOSTS}
install;

#
#  Script para replicacion del sistema  de archivos /usr/local/pgsql
#

HOSTS = ( Newton-replica )
FILES = ( /usr/Sybase )

${FILES} -> ${HOSTS}
install;
```

Estos scripts es todo lo que necesitamos para realizar la replicación de los datos del dragon hacia newton, una vez concluida la replicación los sistemas de archivos en newton se ven de la siguiente manera, resultando la replicación un éxito:

newton # df -k

```
Filesystem kbytes used avail capacity Mounted on
/dev/md/dsk/d0 617275 327903 233818 59% /
/dev/md/dsk/d2 4131866 2646600 1443948 65% /usr
/proc 0 0 0 0% /proc
fd 0 0 0 0% /dev/fd
/dev/md/dsk/d3 3099287 2162276 875026 72% /var
/dev/md/dsk/d4 2056211 466484 1528041 24% /opt
swap 3034584 152 3034432 1% /tmp
/dev/vx/datosdg/dsk/homelog 30945914 12978909 17657546 43% /home/log
/dev/vx/datosdg/dsk/users01 6186810 3778354 2346588 62% /home/users01
/dev/vx/datosdg/dsk/users02 6186810 3492786 2632156 58% /home/users02
/dev/vx/datosdg/dsk/users03 6186810 5857917 267025 96% /home/users03
/dev/vx/datosdg/dsk/users04 3871954 2069039 1764196 54% /home/users04
/dev/vx/datosdg/dsk/sybase 6186810 2881346 3243596 48% /sybase
/dev/vx/datosdg/dsk/raid 3871954 1961986 1871249 52% /raid
/dev/vx/datosdg/dsk/raid2 3871954 1100983 2732252 29% /raid2
/dev/vx/datosdg/dsk/pgsql 1527116 1140770 325262 78% /usr/local/pgsql
/dev/vx/datosdg/dsk/sybase 4743974 3615240 1081295 77% /usr/sybase
/dev/vx/datosdg/dsk/home 30957590 28151616 2496399 92% /home
/dev/vx/datosdg/dsk/users00 6199998 3354000 2783999 55% /home/users00
```

Capítulo 5. Conclusiones

Del presente trabajo podemos concluir que la alta disponibilidad va más allá de la implementación de un cluster, donde existen dos máquinas idénticas con acceso a los mismos discos de datos y en donde una de ellas tiene siempre el control y el funcionamiento de las aplicaciones. Mucha gente tiene la idea errónea de que con el simple hecho de contar con un cluster en su site ya tiene alta disponibilidad en sus sistemas; la experiencia nos dice que para contar con alta disponibilidad primero debemos tener con una buena administración, así mismo se debe incluir como parte de nuestros best practices con un esquema de respaldos adecuado, una vez que se cuenta con lo anterior debemos buscar redundancia en nuestros discos, y finalmente si la aplicación lo requiere hay que tratar de contar con un cluster, si en la empresa para la que trabajamos se considera de vital importancia la aplicación que manejamos, se debe buscar la implementación de la replicación de datos y este será el nivel mas alto de disponibilidad que existe en la actualidad y que se enfoca a la recuperación ante desastres.

Así mismo, el nivel de disponibilidad que hayamos elegido está directamente ligado con los costos, ya que entre mayor sea la disponibilidad requerida los gastos serán mayores debido a que se deberá invertir en hardware, software, centros de cómputo, capacitación, y pruebas por citar solo algunos.

Durante el desarrollo de este proyecto encontramos que existen ciertos lineamientos que se deben seguir cuando se busca que nuestros sistemas sean altamente disponibles; el seguir estos lineamientos, los cuales no son recetas de cocina de cómo levantar un cluster por citar alguno, nos ayudaran a ahorrar dinero, incrementar la disponibilidad de nuestro equipo y a tener a nuestros usuarios felices, estos lineamientos son los siguientes:

1.- Invierte dinero ..., pero no demasiado.

La calidad cuesta dinero, nuestro trabajo como diseñadores de sistemas confiables, radica en explicarle al cliente los costos y beneficios de cada nivel de protección y desarrollar juntos una solución adecuada al nivel de criticidad de su servicio. El que un cliente gaste muchos millones de dólares no implica que se vaya contar con sistemas altamente confiables y por ende disponibles.

2.- No asumas nada.

Si tu compras el equipo mas poderoso en el mercado, este no viene con la alta disponibilidad incluida, se deben hacer pruebas de redundancia en el equipo, pruebas con las aplicaciones, se deben provocar fallas comunes, se debe comprobar que los procesos aun son validos para esta nueva plataforma y desarrollar un plan de migración y créanme que todas estas pruebas no las hacen los vendedores por nosotros, no asumas que toda va a funcionar de maravilla y que tus problemas están resueltos con la compra de este nuevo equipo.

3.- Remueve puntos únicos de falla (SPOF por sus siglas en ingles Single Point of Failure).

Un punto único de falla es un componente (hardware, firmware, software) el cual puede causar degradación en el servicio y en casos más críticos la caída del sistema, para eliminar estos puntos únicos de falla se debe buscar redundancia en los componentes que conforman nuestra operación. Actualmente los nuevos equipos tienen redundancia en fuentes de poder, ventiladores, CPUs, discos, tarjetas controladoras, etc. Al buscar los puntos únicos de falla debemos estar familiarizados con nuestra operación y analizar desde el servidor, las aplicaciones, la operación y terminar con el entorno, es decir, el site. De nada sirve tener un servidor con fuentes redundantes si nuestro site solo cuenta un UPS.

4. Hay que mantener nuestros sistemas seguros.

Existen gran cantidad de libros que hablan sobre el tema, seguramente se han escrito muchas tesis en la universidad y se seguirán escribiendo muchas, pero realmente son pocas las empresas que se preocupan por mantener la integridad de sus servidores. El objetivo es mantener la integridad de los datos y prevenir accesos no autorizados a nuestros equipos que puedan llevarnos a downtimes muy extensos.

5. Consolida tus servidores.

El crecimiento no planeado, aumenta la complejidad de nuestros sistemas, en este caso lo más probable es que tengamos un inmenso centro de cómputo que contiene una gran cantidad de sistemas que no comparten recursos. Las empresas deben de contar con equipos para desarrollo, prueba y producción y algunos con más recursos incluso cuentan con equipos para capacitación. Además, es posible que se cuenten con múltiples sistemas operativos, cada uno con sus propias licencias de software, herramientas administrativas, sistemas de seguridad, etc. Cuando te sientas un momento y analizas tu site, te das cuenta que es muy complejo, lo que necesitas es consolidar tus servidores, se deben agrupar aplicaciones y

servidores que hagan tareas similares, ¿por qué tener varios equipos pequeños destinados a manejar bases

de datos, si puedes adquirir un equipo de gran potencia que soporte todos esos servidores? Seguramente al contestar preguntas como esta te darás cuenta que puedes simplificar lo que hasta este momento era muy complejo y así consolidar tu centro de cómputo.

6. Documenta todo.

La documentación nos provee inicialmente guías para los nuevos administradores una vez que lleguen a nuestra empresa. Una buena documentación debe incluir todo lo que tu consideres necesario entre mas sea mejor, una documentación inicial puede incluir: configuración técnica sobre nuestro servidor, numero de cpus, memoria, configuración de discos, sistema operativo, crones, interfaces de red, usuarios, aplicaciones que ejecuta el server, etc., es necesario incluir procedimientos, manuales de operación, guías de errores, etc. Y asegúrate que tener una copia en papel y en electrónico.

7. Prueba todo.

En la vida diaria de un administrador hacemos algo que no esta considerado como una buena practica, ya que probamos todo cuando tenemos alguna contingencia. Cada seis meses en necesario hacer pruebas de restauración de aplicaciones, si contamos con clusters es necesario hacer failovers de nuestras aplicaciones y si contamos con esquemas de disaster recovery debemos migrar las aplicaciones al site alterno, en todas estas pruebas deben participar los usuarios y apegarnos lo mas posible al esquema actual de producción.

La conclusión más importante de este proyecto radica en que el servidor WEB de la universidad más importante de México ha aumentado la disponibilidad en sus datos. Esto último como resultado de un mejor esquema de respaldos, lo cual implicó hacer algunas modificaciones al que ya existía; se mejoró también la redundancia en los discos de sistema operativo y de los datos de aplicación, esto se logró con una reestructuración de la configuración actual y con el software de manejo de discos Veritas Volume Manager y Solstice Disk Suite; además se hizo replicación de datos hacia un segundo servidor.

Es importante mencionar que hay algunos aspectos que nos hubiera gustado mejorar, uno de ellos es la implementación de un cluster, pero desafortunadamente era necesario adquirir hardware y software adicional lo cual obviamente implicaba un gasto extra para la UNAM, por tal motivo no lo pudimos implementar, sin embargo en este capítulo vamos a dejar un esquema de un cluster por si en determinado momento las condiciones de la universidad permitieran ponerlo en práctica, este esquema de cluster en determinado momento podría orientarse a un esquema de replicación de datos mas robusto y confiable del que

implementamos en el capítulo 4. Primero veremos la teoría y posteriormente lo que se haría directamente en el servidor.

Definición y Características cluster y failover

Diagnósticar una falla, puede tomar horas y en algunos casos dias. Una vez que la falla es diagnosticada, si el problema es el hardware, un reemplazo de la parte que ha fallado debe ser obtenido, y después alguien con los conocimientos necesarios debe hacer el cambio. Si el problema es el software un parche para la aplicación o el sistema operativo debe ser aplicado.

Si en el servidor corren aplicaciones criticas, esta interrupción de horas o días es simplemente inaceptable. Una solución práctica es implementar una configuración compuesta por dos o mas servidores ligados entre sí, de esta manera si un servidor falla, el otro podrá tomar los procesos y servicios y por lo tanto mantendrá el sistema funcionando, a esto se le conoce como **cluster**. Para garantizar la consistencia de los datos y una rápida recuperación, los servidores deben tener conectados los mismos discos de datos, es decir los discos deben ser compartidos.

En una configuración de cluster a la migración de los servicios de un servidor a otro se le conoce como **failover.**

Como mínimo la migración de los servicios durante un failover debe tener las siguientes características.

Transparente. El failover no debe interferir con los clientes que accesan a los servicios de un servidor, tal vez sea necesario para los usuarios volver a conectarse a su aplicación, pero solo en algunos casos como lo son bases de datos primarias.

Rápido. Un failover no debe de tomar más de cinco minutos, idealmente menos de dos minutos. Si un reboot completo es requerido para realizar el failover, el tiempo se incrementara y tal vez pueda llevarse a cabo en una hora o más.

Intervención manual mínima. Idealmente, la intervención humana no debe de ser requerida del todo; el proceso entero debe ser automático. Algunos sites o aplicaciones requieren inicialización manual para el failover, pero eso no es recomendable. Bajo ninguna circunstancia un failover debe requerir reboot del servidor que tomara el control.

Acceso de datos garantizado. Después de un failover, el hosts activo deberá poder acceder a los mismos datos críticos que el hosts original. En una configuración de cluster los servidores deben también de estar comunicados de manera continua, de esta manera cada

sistema sabe el estado en que se encuentra su compañero. Esta comunicación es llamada hearbeat.

Cuando un failover ocurre, tres elementos críticos deben ser movidos del servidor que ha fallado al servidor activo:

1. Identidad de la red. En un ambiente de red ethernet, la identificación de la red se refiere la dirección IP y en algunos casos a la dirección MAC.

2. Acceso a discos compartidos. Esencialmente la tecnología de sistemas operativos prohíbe el acceso de múltiples servidores a los mismos discos al mismo tiempo. En una configuración de discos compartidos, el acceso lógico es restringido a un servidor al mismo tiempo.

3. Reasignación de procesos. Una vez que los discos han sido migrados al servidor pasivo, todos los procesos asociados con los datos deberán ser reiniciados.

Componentes de un Cluster

Un cluster es un arreglo de dos más servidores los cuales tiene acceso a los mismo arreglos de disco y cuyo propósito principal es reducir los lapsos de tiempos muertos, cuando una falla ocurre. De esta forma cuando algún servidor tiene una falla existe un servidor que se encarga de realizar el trabajo que estaba haciendo el servidor que falló.

Nos enfocaremos un poco mas a los clusters integrados por dos servidores, ya que la propuesta de este capitulo está orientada a la implementación de una configuración de cluster formada justamente por dos servidores.

Los componentes necesarios para la implantación de un cluster incluyen lo siguiente.

Servidores

Fundamentalmente cuando se diseña un cluster debemos de tener al menos dos servidores, los cuales idealmente deben ser idénticos tanto en hardware como en software. Los servidores deben tener la misma versión de sistema operativo con la misma versión de programas de aplicación y la misma revisión de parches. También deben de tener la misma cantidad de memoria, el mismo número de procesadores a la misma velocidad, las mismas tarjetas de red y de conexión a los discos y además una carga de trabajo similar. Los servidores

deben de ser tan idénticos como sea posible. En una configuración de cluster los servidores reciben también el nombre de nodos.

En un cluster de dos nodos, un servidor es conocido como servidor primario y el otro como servidor secundario.

Servidor Primario. Es el servidor principal, cuando el cluster empieza a funcionar este es el servidor que tendrá el control de las aplicaciones. En ese momento este servidor también se conoce como servidor activo.

Servidor Secundario. También se conoce como servidor pasivo. Este servidor no está realizando trabajo cuando el cluster comienza a trabajar, solo está en espera de que alguna falla en el servidor primario ocurra para entrar en acción, cuando esto ocurre toma el control de las aplicaciones y pasa de ser servidor pasivo a servidor activo, y en contraparte el servidor primario que era el servidor activo hasta antes de la falla, pasa a ser el servidor pasivo.

Redes

Existen tres tipos diferentes de redes que son requeridos para implementar un cluster: red heartbeat, red pública o de servicio y red administrativa. Lógicamente cada una de estas redes tiene una dirección IP asignada y no debe ser la misma para ninguno de los tres tipos.

Red heartbeat

Las red de heartbeat es aquella mediante la cual los nodos de un cluster tienen comunicación uno con el otro. Fundamentalmente los servidores se están monitoreando uno a otro a través de los links de este tipo de interfaces. Los paquetes de heartbeat pueden ser muy complejos y generalmente contienen información del estado en que se encuentra cada servidor. Podemos tener mas de una red de este tipo en un cluster, aunque es recomendable tener cuatro o cinco tal vez no tendría caso, normalmente son dos las redes heartbeat que se utilizan en un cluster de dos nodos.

Las redes heartbeat pueden ser implementadas sobre cualquier tarjeta confiable. Muy frecuentemente, las tarjetas ethernet 10 base T son el medio seleccionado para redes heartbeat. Usar una red más rápida o más compleja de este tipo no tendrá un valor significativo para el sistema en lo absoluto, entonces para que gastar más dinero. Redes mas rápidas son ideales cuando se tienen que mandar largas cadenas de datos, pero los mensajes

de heartbeat son cortos, por lo tanto sería poco rentable utilizar tarjetas de mayor velocidad para dichas redes.

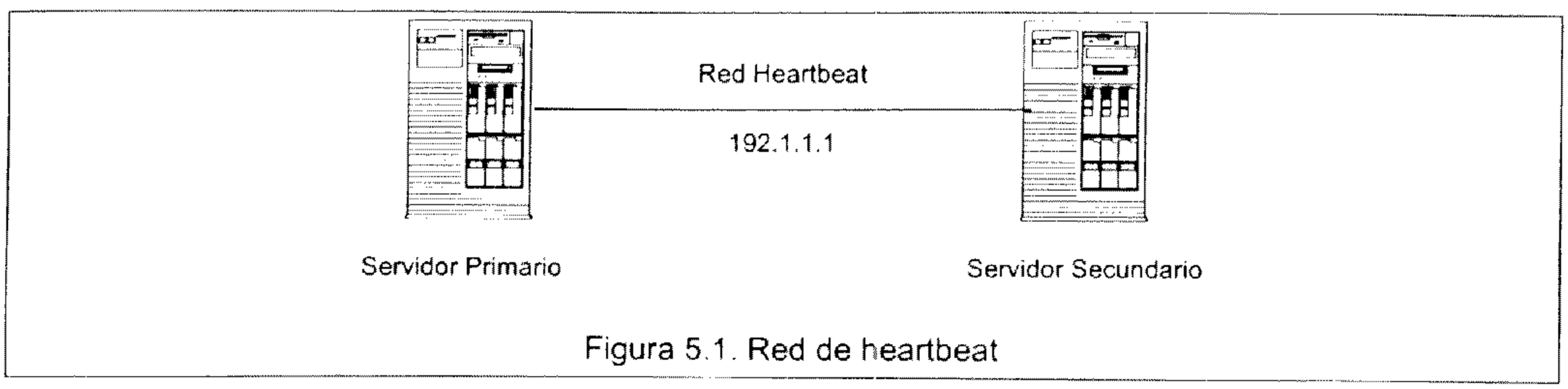


Figura 5.1. Red de heartbeat

Red Pública

Con el propósito de brindar el servicio para el cual fue implementado, el cluster necesita estar conectado al menos a una red de servicio público. Esta red debe ser la misma a la que los clientes deben conectarse para acceder las aplicaciones críticas que están corriendo en el servidor. La red pública es la red visible del cluster. Por supuesto, los servidores seguramente podrán brindar acceso a sus clientes a través de más de una red.

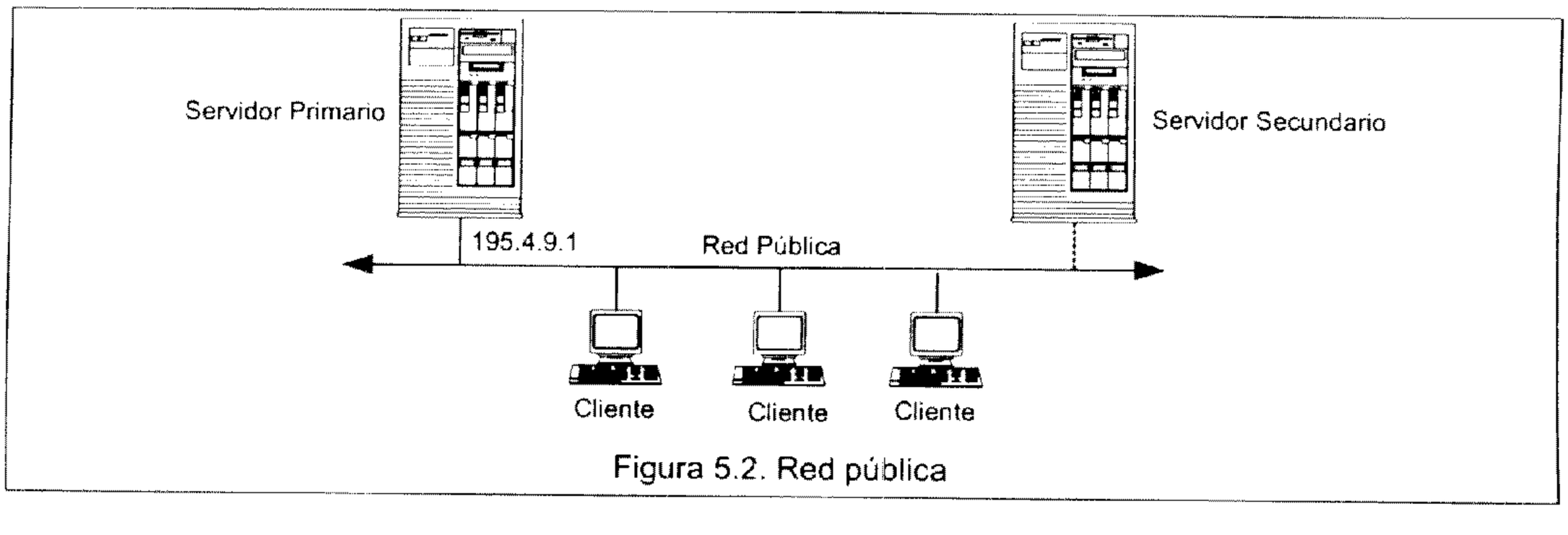


Figura 5.2. Red pública

Red Administrativa

En algunas configuraciones de clulster, el servidor secundario iniciará sin conectividad a la red pública, solo con las conexiones de heartbeat de su servidor compañero en el cluster.

El efecto de la falta de conectividad de la red pública al iniciarse el servidor puede ser seriamente perjudicial. Los servicios de nombres fallarán, así como el e-mail, la impresión de archivos y otros servicios que implique el uso de esta red. Si la interfase de red pública no está presente, la implementación inicial será perjudicada en gran medida porque esos servicios

necesitarán estar deshabilitados a la hora del inicio del servidor y solo estarán arriba cuando el servidor reciba un failover.

Desde este punto de vista es mejor configurar los servidores de un cluster con una interfase adicional a la red pública. Podemos llamar a esta conexión adicional red administrativa porque es utilizada solamente para propósitos administrativos. Además de proveer conectividad de red básicos y servicios, esta conexión es una garantía para los administradores del sistema hacia un servidor particular, mas que un servidor anónimo brindando un servicio particular. La interfase administrativa permite al administrador entrar al servidor que ha fallado e investigar la causa del failover antes de que este sea puesto nuevamente en producción o stand by.

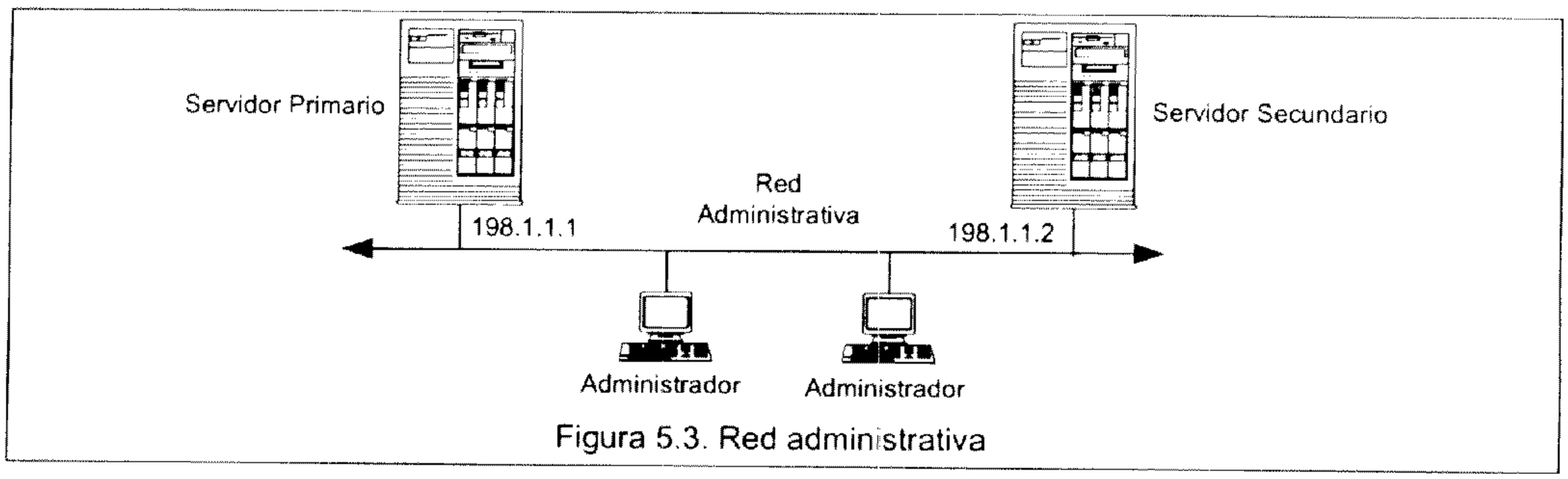


Figura 5.3. Red administrativa

Discos

Hay dos tipos diferentes de discos requeridos para un cluster. Los discos no compartidos o privados y los discos compartidos o públicos.

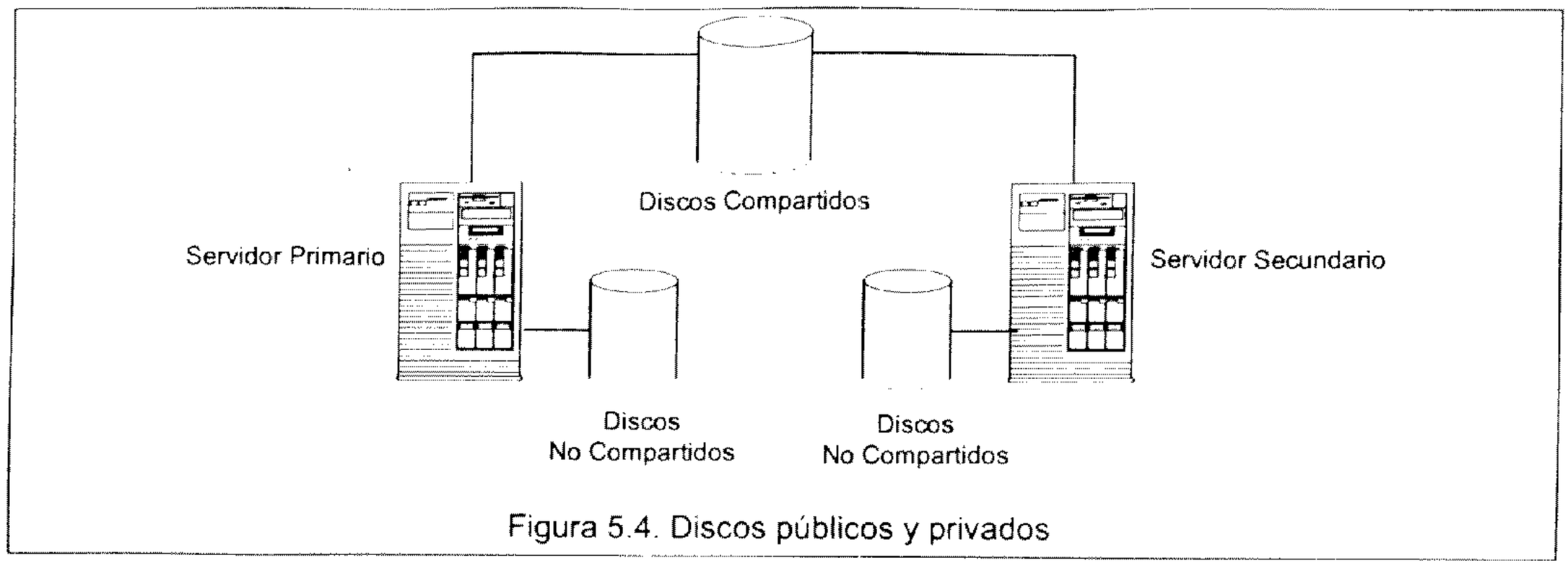


Figura 5.4. Discos públicos y privados

Discos No Compartidos

Los discos no compartidos son propios de cada servidor y son aquellos que contienen el sistema operativo y otros archivos que son requeridos para su operación, incluyendo algún software necesario para iniciar y manejar el proceso de failover.

Estos discos generalmente están localizados en el interior de cada servidor, aunque ese no es un requerimiento estrictamente necesario. En realidad, es mejor que los discos privados estén localizados físicamente al exterior de los servidores.

Los discos privados por definición no deben ser accesados por los dos servidores, solo un servidor debe hacer uso de los datos contenidos en ellos. Además todo los datos de los discos privados deberá ser espejeado utilizando algún software manejador de discos o incluso de manera física.

Muchos archivos de administración deberán mantenerse en perfecta sincronización en los dos servidores del cluster. No existe herramienta de ayuda en este tipo de sincronización; esto deberá hacerse automáticamente. De tal forma que es muy importante que cuando se haga un cambio en un servidor de algún archivo de configuración, se realice lo mismo en el otro, de lo contrario esto ocasionará que los failover no se hagan de manera correcta.

Discos Compartidos.

Los discos compartidos son aquellos donde reciden los datos de aplicación crítica. Estos son los discos que se transfieren entre los servidores del cluster cuando ocurre un failover y deben de ser accesibles por los dos servidores, aunque solo deben ser accesados por un

servidor a la vez ya que si los dos servidores tratan de escribir en los discos compartidos al mismo tiempo, será inevitable la corrupción de datos.

Al igual que los discos no compartidos, todo el contenido de los discos compartidos debe de tener alguna clase de redundancia como medida de protección ante pérdida de información o canales de transmisión de datos.

Poniendo aplicaciones críticas sobre discos

Una de las cuestiones mas interesantes en el diseño de sistemas de HA es donde poner los ejecutables de aplicaciones críticas. Existen dos opciones: podemos ponerlos sobre los discos privados, con la información del sistema, o podemos ponerlos con los discos compartidos, con los datos asociados.

Si instalamos las aplicaciones sobre los discos compartidos, la buena noticia es que solo necesitaremos mantener una copia de los ejecutables y de los archivos de configuración asociados; el cambio deberá ser hecho dos veces; en otro caso un failover no garantiza un ambiente idéntico de la otra parte del cluster.

Aunque solo con una copia de las aplicaciones, es casi imposible instalar y actualizar las aplicaciones de manera consistente, y con la opción de un regreso (rollback). Con dos copias de la aplicación, y el sistema A activo, podemos hacer el upgrade sobre el sistema B, y failover hacia B. Si el upgrade es exitoso, entonces habrá un failover hacia B, corriendo la actualización de la aplicación correctamente, entonces podremos actualizar el sistema A. Si la actualización falla solo necesitamos regresarnos al sistema A y volver a realizar el upgrade en el sistema B.

Esta es una de las preguntas para las cuales no hay una sola respuesta correcta.

Depende del ambiente, de la frecuencia con la cual se realicen actualizaciones, y de la complejidad para manejar ambientes paralelos.

Portabilidad de la aplicación

En el diseño de un cluster es importante tomar en cuenta que las aplicaciones deben de poder corren en ambos servidores, y solo sobre un servidor a la vez. Frecuentemente la licencia es un impedimento para esto, a menudo el motivo es el gasto que ello implica. Si las aplicaciones no corren en los dos servidores, y solo sobre uno a la vez entonces no tendremos alta disponibilidad. Es necesario tener en cuenta este punto y llegar a un acuerdo con el proveedor de las aplicaciones.

Eliminación de puntos de falla

Este es un componente muy general que se refiere a cualquier elemento en el cluster. Si en el cluster hay un componente que falle esto causará seguramente que el o los servidores no estén disponibles, de esta manera no se tendrá alta disponibilidad.

Software de Manejo Failover

Una opción para implementar el manejo de failover es el software de failover que algunos administradores han escrito durante el tiempo que han estado trabajando con clusters de servidores. Aunque tiene algunas desventajas. Sistemas de manejo de failover comerciales ofrecen muchas ventajas sobre el que ha sido escrito por administradores, ya que entre otras cosas es maduro, probado y robusto. Tiene soporte las 24 horas del dia los siete dias de la semana. Se puede instalar fácilmente y comunicar con software de monitoreo de terceros (Tivoli, Solstice domain Manager, HP Open View, BMC Patrol, CA Unicenter , etc.), y también puede ser capaz de monitorear muchas aplicaciones populares de terceros, incluyendo bases de datos, servidores web, servidores de terceros, y servidores de archivos.

A continuación tenemos un breve resumen de algunos productos de Software Manejadores de Failover comerciales:

Veritas FirtsWatch. Disponible para Solaris y otros sitemas operativos. Soporta todas las redes comunes, todos los manejadores de datos y otras aplicaciones comunes. Soporta combinaciones de servidores SUN y discos de terceros, es fácil de instalar y de adaptar a nuevas aplicaciones.

Veritas Cluster Server (VCS). Disponible para Solaris y otros sistemas operativos. VCS hace todo lo que hace FirstWatch, pero también soporta ambientes SAN, failovers basados en reglas, y reglas de failover complejas en configuraciones de hasta 32 nodos.

Legato Fulltime HA+. Disponible para Solaris y otros sistemas operativos. Soporta todas las redes comunes, la mayoría de las bases de datos y otras aplicaciones. También soporta combinaciones de servidores SUN y discos de terceros.

Sun Cluster HA. Disponible solo para solaris. Soporta todos los servidores y arreglos de discos SUN. Algunos arreglos de discos de terceros también son soportados. Puede ser un poco complejo de instalar.

Tipos de configuración de Cluster

Cuando se implementa un esquema de failover podemos seleccionar de entre tres tipos diferentes:

Configuración de dos nodos, configuración de más de dos nodos y configuración no convencional. Cada una de ellas tiene sus características propias que la hacen diferentes de las demás en algún punto.

Independientemente del tipo de sistema o de método que seleccionemos, es indispensable que todos los nodos dentro del cluster estén sincronizados en el tiempo. Aplicaciones y sistema operativo pueden llegar a confundirse en caso de no ser así: el tiempo siempre debe fluir en la misma dirección, hacia adelante y a la misma velocidad. Si los sistemas no están sincronizados, es imposible para un sistema crear un archivo sobre otro sistema.

Configuración de dos nodos

El más común y el más simple tipo de configuración de failover es el de dos nodos. A su vez la configuración de dos nodos se dividen en dos tipos: asimétrica y simétrica. En la configuración asimétrica un nodo está activo y realizando el trabajo crítico, mientras su nodo compañero está solamente en stanby, listo para entrar en acción cuando el primer nodo falle. En la configuración simétrica, ambos nodos están realizando en trabajo crítico de forma independiente y en el caso de que un nodo falle, el nodo que en ese momento esté arriba realizará todo el trabajo hasta que el nodo afectado levante nuevamente.

Configuración asimétrica uno a uno

El cluster asimétrico es la base de una configuración de cluster. Cualquier configuración es solo una variación de este modelo. En una configuración asimétrica, hay un servidor primario o maestro el cual normalmente provee todos los servicios críticos en el cluster. La figura que se ve a continuación nos muestra un servidor primario el cual está conectado a un servidor de respaldos dedicado a través de dos redes Heartbeat dedicadas. En esta configuración, los heartbeat son simples redes punto a punto.

Los dos servidores están conectados a un conjunto de discos. Estos discos están idealmente divididos entre dos controladoras separadas y dos arreglos de discos también separados, y los datos están espejeados de una controladora a otra. (Por supuesto que la

configuración trabaja bien si los espejos están sobre una controladora única o en un arreglo de discos único). Un disco particular o file system solo debe ser accesado por un servidor a la vez. Los dos servidores están también conectados a la misma red pública, con los clientes en producción sobre ella. Ellos comparten una dirección IP de red única la cual es transferida por el software manejador de failover de un servidor a otro. Esta dirección se conoce como dirección virtual, ya que no está asignada permanentemente a un hosts en particular, solo un servidor en el cluster puede tener dicha dirección a la vez. El otro servidor no tiene identidad en la red pública y por lo tanto no puede ser accesado por los usuarios. Las redes heartbeat se manejan de forma separada de las interfaces públicas, y deben ser configuradas para que no soporten ruteo. Servidores con una configuración de failover configurada propiamente tendrán también interfaces de red administrativas, lo cual permitirá a los administradores de sistemas acceder los servidores aunque ocurra un failover.

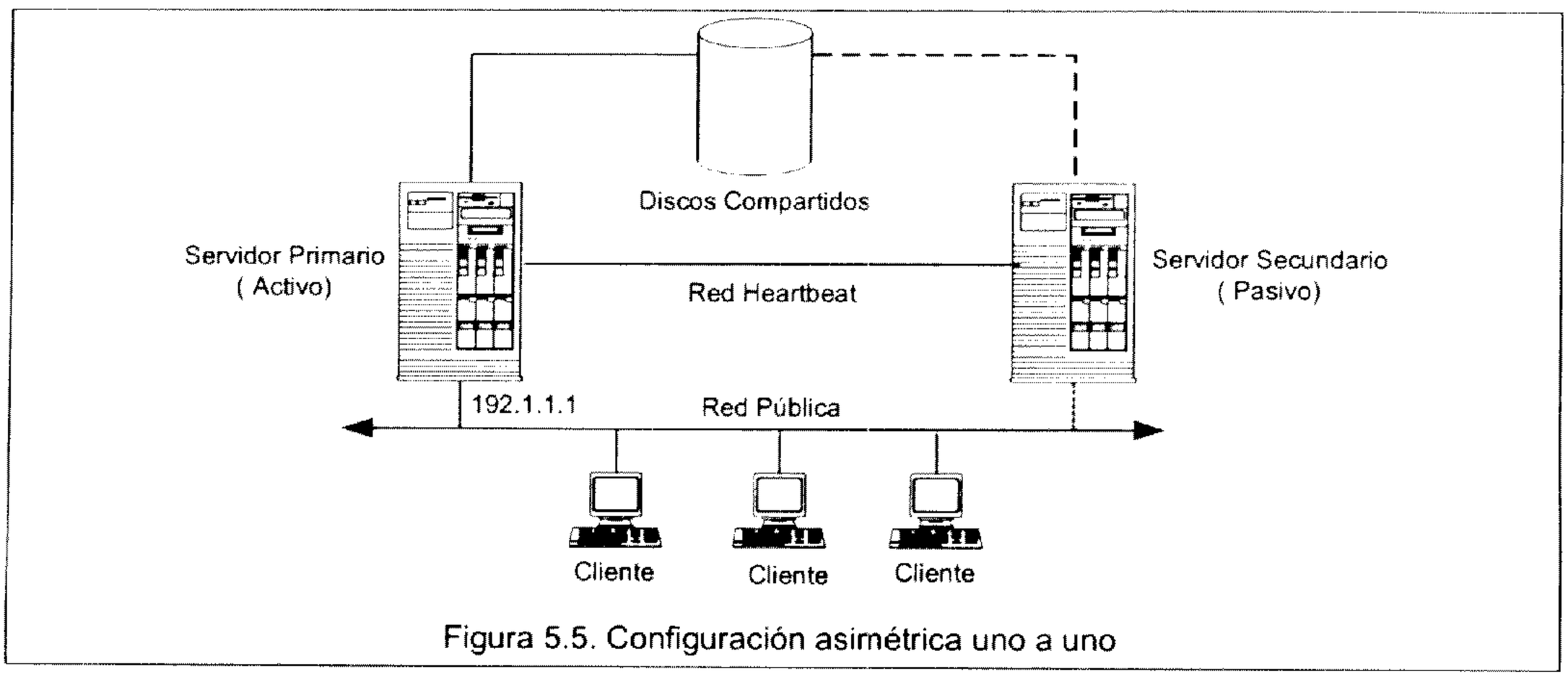


Figura 5.5. Configuración asimétrica uno a uno

Después de que se ha completado un failover asimétrico (figura siguiente), el servidor que toma el control es el que accesa los discos, el que es dueño de la dirección virtual y el que está corriendo los procesos críticos. No hay cambio en las conexiones físicas, todos los cambios son a nivel de software.

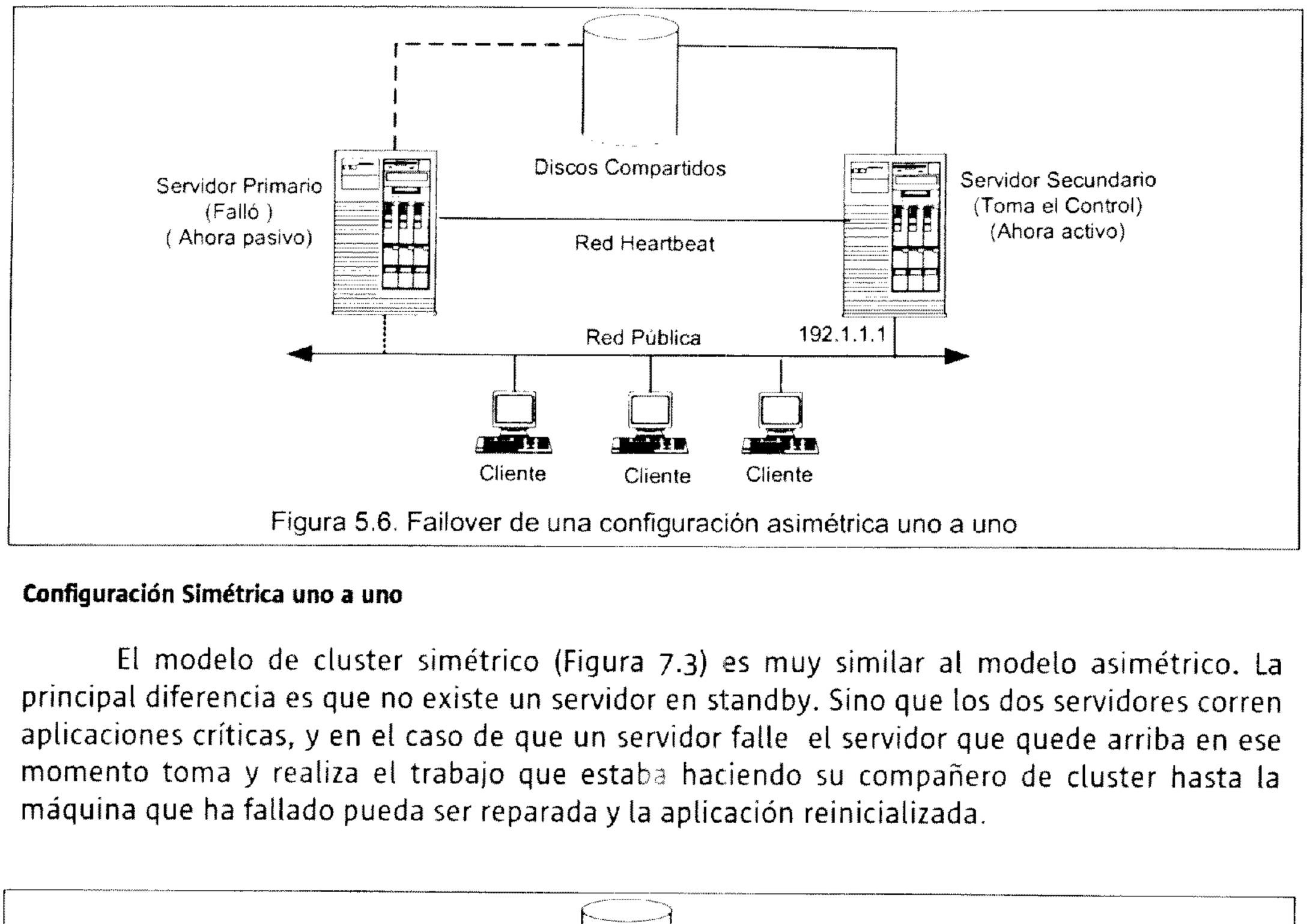


Figura 5.6. Failover de una configuración asimétrica uno a uno

Configuración Simétrica uno a uno

El modelo de cluster simétrico (Figura 7.3) es muy similar al modelo asimétrico. La principal diferencia es que no existe un servidor en standby. Sino que los dos servidores corren aplicaciones críticas, y en el caso de que un servidor falle el servidor que quede arriba en ese momento toma y realiza el trabajo que estaba haciendo su compañero de cluster hasta la máquina que ha fallado pueda ser reparada y la aplicación reinicializada.

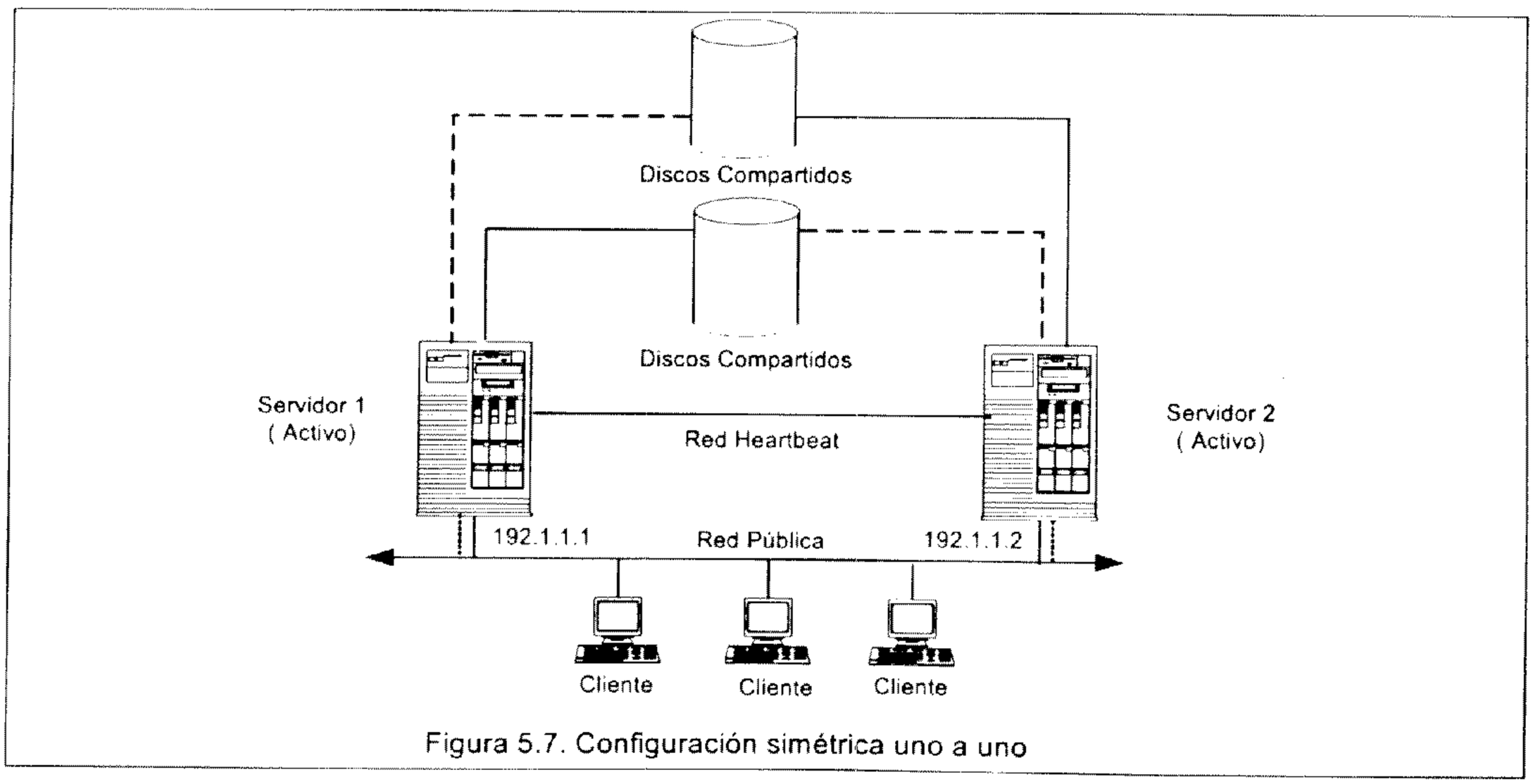


Figura 5.7. Configuración simétrica uno a uno

La siguiente figura muestra como en una configuración simétrica uno a uno, el servidor 1 toma el control después de que el servidor falló, al hacer esto dicho servidor toma la dirección IP del servidor 2, el control de sus discos y obviamente sus procesos y servicios, hasta que el servidor 2 este funcionando de forma correcta nuevamente.

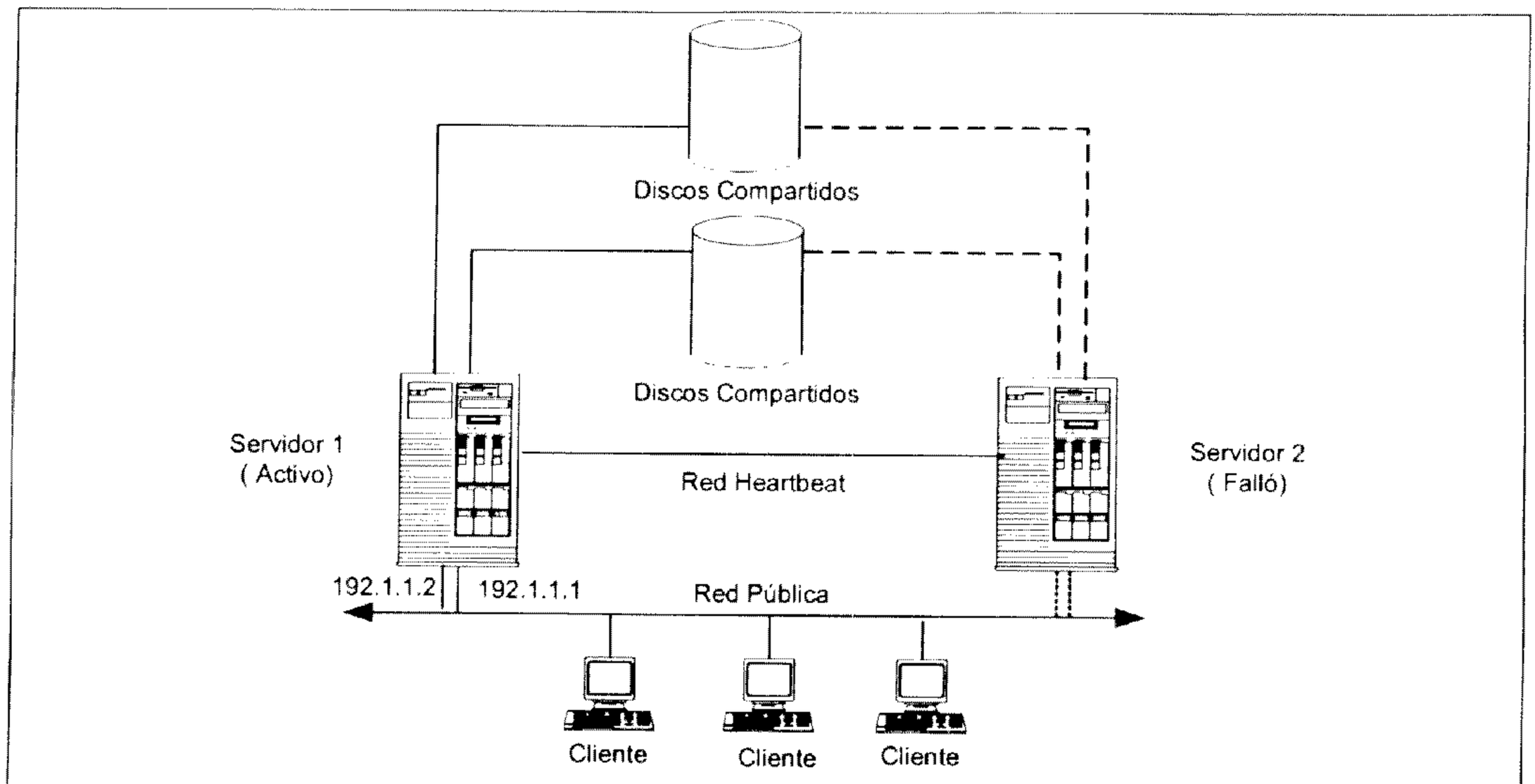


Figura 5.8. Failover de una configuración simétrica uno a uno

Desde una perspectiva de costos y rendimiento, la configuración de failover simétrica es el mejor camino, ya que hace uso de todo el hardware.

En una configuración simétrica existe el inconveniente del impacto inevitable del performance que tendrá uno de los dos servidores cuando alguno tome toda la carga de trabajo por falla del otro servidor, aunque la forma de contrarestar esto puede ser comprar CPU´s u memoria extras para ambos servidores, de tal forma que estos puedan manejar de forma mas fácil la carga adicional.

Elección de tipo de configuración

Llegado el momento tendremos que decidir, configuración simétrica o asimétrica? La configuración asimétrica es mejor para la alta disponibilidad, pero es más difícil de manejar.

La configuración simétrica es mas cara, pero en cambio aprovecha el hardwatre de mejor manera.

Configuración de mas de dos nodos

Por razones de minimizar las pérdidas de dinero, las compañías frecuentemente tratan de construir configuraciones de failover que involucren muchas máquinas en combinaciones complejas. Todas estas configuraciones trabajan virtualmente con el mismo modelo de failover uno a uno.

Un modelo de failover N a uno es aquel en el que existen máquinas corriendo aplicaciones críticas con todas las fallas dentro de un servidor único en standby. En este contexto estamos hablando específicamente de configuraciones tradicionales que no son de canales de fibra.

Las configuraciones N a uno son tradicionalmente limitadas por tres o cuatro servidores que hacen el failover a un servidor que está en standby. El servidor standby debe tener suficientes tarjetas para todas las redes y buses de disco que requerirán acceso.

Configuración Asimétrica N a Uno

En la figura 7.6 podemos ver un ejemplo clásico de una configuración de cluster cuatro a uno. Cada servidor primario en el grupo está conectado a sus propios discos y a su vez todos los discos se conectan al servidor standby. El servidor de standby puede tomar el lugar de alguno de los servidores que fallen. Aunque no existe relación directa entre los cuatro servidores primarios, son totalmente independientes.

La ventaja de esta configuración es que existen menos servidores sin hacer nada que si estuvieran en serie o en pares de configuraciones asimétricas. Pero así como hay ventajas, también hay desventajas. La más notoria es que el servidor standby no será utilizado la mayor parte del tiempo. Posiblemente pasen meses entre un failover y otro, y durante ese tiempo el servidor standby estará ocioso y con él todo el hardware que lo compone.
La otra desventaja de esta configuración es que si falla el servidor standby, los cuatro servidores restantes estarán sin protección.

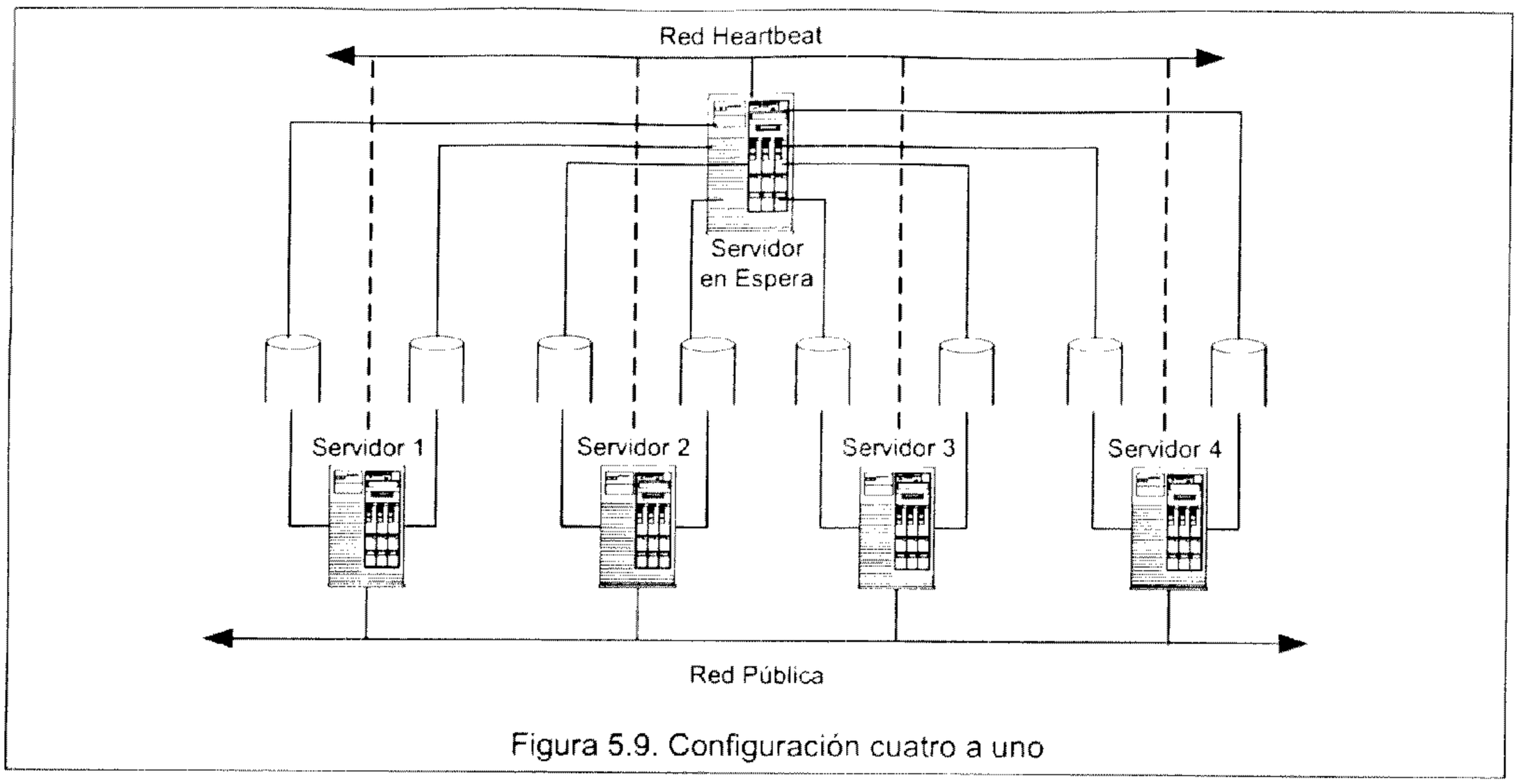


Figura 5.9. Configuración cuatro a uno

Configuración Multiservidor Mediante Red

Estas configuraciones representan un modelo mucho más robusto para múltiples hosts que un modelo N a uno. Estas configuraciones es llamada algunas veces N a N o N a M, porque cualquier hosts puede mandar sus servicios a otro. Asumiendo que el hosts superviviente tiene suficiente capacidad, y que se hizo una configuración apropiada, podríamos perder tres miembros de esta red y continuar proviendo servicios a través del cuarto.

Configuración no convencional

Este tipo de configuraciones puede variar, se caracteriza porque las personas que los diseñan hacen alguna variante de los tipos de configuraciones que conocemos rompiendo en algunos casos ciertas reglas, por ello se conocen como no convencionales.

Configuración Simétrica N a uno

Una configuración simétrica N a uno es muy similar a una configuración asimétrica N a uno. La única diferencia real es que el servidor standby también corre aplicaciones críticas, y cuando este falla migra a cualquiera de los otros servidores que ha sido configurado como predeterminado. Aunque este modelo elimina tiempos de ocio de servidor, carga con las desventajas de un modelo asimétrico N a 1.

Configuración Asimétrica uno a N

Este modelo que se muestra en la figura 7.10 se parece al modelo asimétrico N a uno. Aunque un servidor único es el backup de muchos, en esta configuración corremos todos nuestros servicios de aplicación críticos en un servidor primario único, y si es servidor falla, todos nuestros grupos de servicios son migrados a muchas máquinas más pequeñas.

Este modelo es de utilidad si deseamos usar algunos equipos viejos como servidores standby para una aplicación en particular más que para un conjunto de servicios. Aunque otra vez, los ahorros son falsos, ya que son absorbidos por la complejidad del modelo. Si decidimos emplear este modelo, debemos asegurarnos de que todos los servidores que seleccionemos estén corriendo la misma versión de sistema operativo y de que puedan correr la misma versión de binarios. Si hacemos esto con algunas estaciones de trabajo debemos estar seguros que todas las máquinas tengan la misma arquitectura de kernel. También deberemos asegurarnos de que tenemos suficientes slots en el backplane y de que todo nuestro hardware nuevo puede trabajar con los nuevos equipos.

Modelo Simétrico Round-Robin

De las configuraciones no convencionales, round-robin es definitivamente la mas interesante y la que probablemente tenga el mayor potencial. En una configuración round-robin, cada servidor corre una aplicación crítica y actúa como standby solo para otro servidor. En la configuración round-robin de cinco hosts que se muestra en la figura 7.11, ginger corre una aplicación crítica, la cual hará failover a baby en caso de ser necesario. Ginger es también el servidor standby para scary. Ginger no tiene relación con posh o sporty. Aunque, si sporty falla, posh tomará su lugar.

Lógicamente los servidores en una configuración de round-robin están arreglados en un anillo, una de las ventajas de este tipo de configuración es la escalabilidad. Aunque son necesarias algunas reconfiguraciones, no será tan difícil agregar un sexto servidor llamado spice al modelo. Solo necesitaríamos cambiar a sporty para que hiciera failover a spice o a posh y necesitaríamos decirle a posh que aceptara un failover de de spice o de sporty. Round-Robin también tiene cuidado de algunas desventajas que se presentan en otras configuraciones offbeat. Aquí no existen máquinas ociosas o sin hacer nada. Ningún hosts necesita demasiada cantidad de recursos de cómputo porque ningún hosts soportará mas de dos veces de su carga normal de trabajo. Cada hosts no requiere mucho mas hardware del que ocuparía en una configuración simétrica regular.

La otra ventaja que ofrece una configuración round-robin es que parece terriblemente inteligente ante el primer problema que se presente. La desventaja es que si lo implementamos es poco probable que alguno actualmente tenga soporte sobre uno anterior.

Propuesta de un cluster para servidor WEB

Idealmente un cluster debe estar configurado en servidores diferentes y en centros de cómputo separados geográficamente por varios kilómetros de distancia, bajo este esquema es factible la implementación de un cluster global del servidor web de la UNAM tomando como base las modificaciones que se han hecho a lo largo de este trabajo en el servidor web y en el servidor llamado newton. Es por esto que realizaremos una propuesta de un cluster para que en caso dado de que la universidad cuente con los recursos económicos necesarios para adquirir mas hardware y software, esta propuesta sea llevada a cabo. Es muy importante destacar que de ponerse en práctica esta propuesta la replicación de datos que se implementó en el capitulo 4, desaparecería para dar a lugar a la nueva configuración.

Los dos servidores que se usarían son el servidor web de la UNAM también conocido como dragón y cuya ubicación como sabemos está en el centro de cómputo de la DGSCA, y el servidor newton que se encuentra en el centro de cómputo de Pitágoras.

Reestructuración y reconfiguración de los servidores

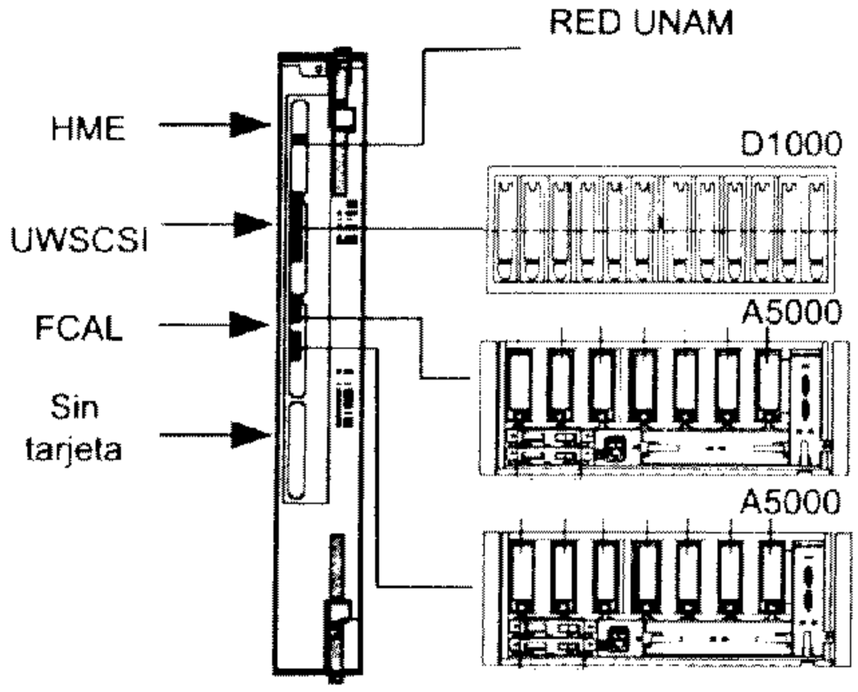
Lo primero que se haría sería diseñar la distribución de hardware de ambos servidores. En este caso los dos servidores quedarían formados por 2 tarjetas de sistema y cada uno en su plataforma actual. Las tarjetas de sistema que se podrían ocupar sería la 0 y la 1 de su plataforma respectiva.

Actualmente dragón tiene tres tarjetas de entrada salida, la primera está en la posición 0 del sbus 0 y es una tarjeta de red hme, esta tajreta se utiliza para la red pública; la segunda tarjeta está en la posición 1 del sbus 0 y es una ultra wide SCSI, se usa para conectar los discos del arreglo D1000; la tercera tarjeta está en el slot 0 del sbus 1 y mediante ella está conectado el servidor a los arreglos A5000.

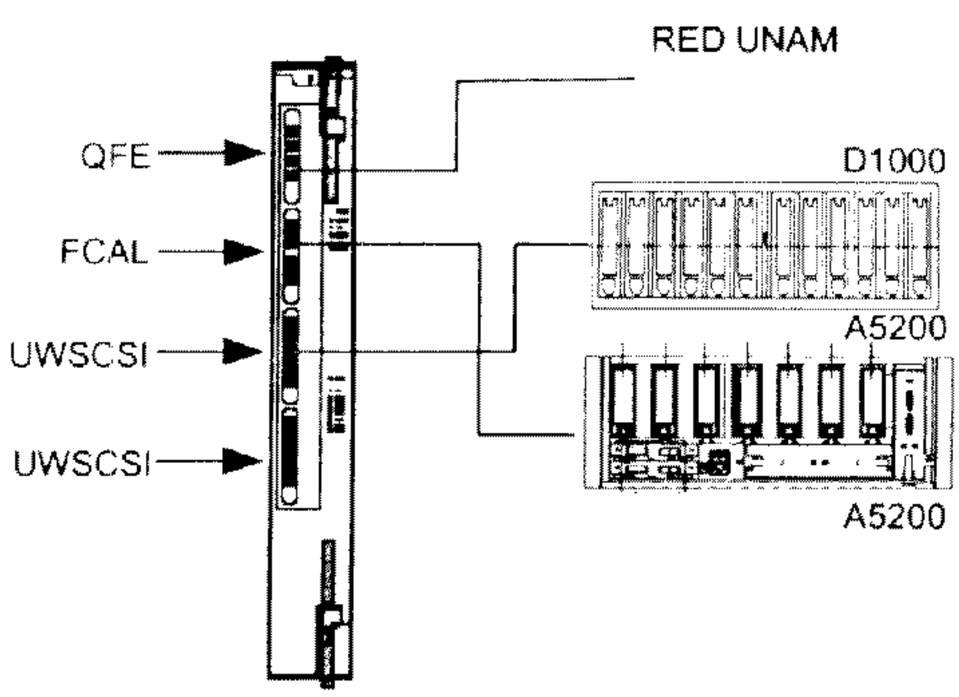
Por su parte newton tiene cuatro tarjetas de entrada salida de datos, la primera es una tarjeta de red qfe, esta tarjeta tiene capacidad para configurar hasta cuatro interfases de red, pero solo utiliza una para la red pública, está en el sloto del sbus 0; la segunda es una tarjeta de fibra fcal para conectar los A5200 al servidor y está en el spot 1 del sbus 0; y en el slot 0 y 1

del sbus 1 tenemos una tarjeta ultra wide SCSI en cada ranura, solo se está usando una para conectar los discos D1000.

Debemos mencionar también que dragón tiene cuatro procesadores a 333 MHZ y 2 GB de memoria, mientras que newton tiene cuatro procesadores a 400 MHZ.



Tarjeta de sistema 0 de silicio



Tarjeta de sistema 0 de cientificos

Figura 5.10. Componentes actuales de hardware de dragón y newton

Para la nueva configuración nosotros proponemos comprar o siguiente:

Comprar el siguiente hardware:

- 1 tarjeta hme
- 3 tarjetas qfe
- 2 tarjetas fcal
- 2 arreglos D1000 con capacidad para 12 discos, con
- 3 discos de 18 GB para los arreglos D1000
- 3 arreglos de discos A5200 con 12 discos de 18 GB c/u
- Adicionalmente comprar 2 discos mas para arreglos A5200
- 2 tarjetas de sistema, cada una con 4 GB de memoria y 4 procesadores a 400 MHZ
- Adicionalmente comprar 4 GB de memoria
- También comprar adicionalmente otros cuatro procesadores a 400 MHZ
- 2 cables SCSI
- 4 cables defibras

Comprar el siguiente software:

- Veritas Global Cluster

El hardware anterior y el existente se distribuiría de la siguiente forma:

Con el propósito de no interrumpir el servicio del web de la UNAM, la nueva configuración se haría primero en el servidor newton. Pretendemos que con la posible reestructuración tanto newton como dragón queden conformados por dos tarjetas de sistema, la tarjeta de la posición 0 y la de la posición 1, pero en el caso de la plataforma llamada cientificos donde está configurado newton, la posición de tarjeta de sistema número uno está siendo utilizada por el servidor einstein, el cual es un servidor que se mantiene ocioso la mayoría del tiempo, y que por lo tanto se puede cambiar de posición sin que esto le afecte, la posición a la que se cambiaría sería a la posición 3, de esta manera ya tendríamos disponible la posición 1 para que fuera parte de newton.
Una vez hecho lo anterior se daría de baja el servidor newton, se sacaría la tarjeta de sistema 0 y junto con una de las tarjetas de sistema nuevas se reestructurarían de la siguiente forma:

- A la tarjeta de sistema 0, se le sustituirían los cuatro procesadores que tiene y que son a 333 MHZ por los cuatro nuevos que son a 400 MHZ.
- Se le agregarían 2 G de memoria RAM también a la tarjeta de sistema 0 para que quedara con 4 GB.
- La posición 0 de la tarjeta de sistema 0 quedaría con una tarjeta ultra wide SCSI para que a través de ella el servidor se conectara a uno de los arreglos D1000; en la posición 1 quedaría una tarjeta qfe, de está tarjeta se configuraría una interfase para la red pública o de la UNAM; la posición 2 quedaría con una tarjeta hme la cual se usaría para configurar la red de heartbeat entre dragón y newton; y en la posición cuatro quedaría con una tarjeta fcal para conectar el servidor a un arreglo de discos A5200.
- En lo que respecta a la tarjeta de sistema número 1, en la primera posición quedaría una tarjeta ultra wide SCSI a través de la cual estaría conectado el otro arreglo de discos, de esta manera serían dos trayectorias por donde estaríamos viendo los dos arreglos y en caso de alguna falla física de una de las dos tarjetas SCSI o del cable tendríamos uno de reserva; en la posición 2 de esta misma tarjeta de sistema quedaría configurada una qfe que serviría para implementar una red administrativa entre ambos servidores; la posición 3 quedaría vacía; finalmente en la posición cuatro de la tarjeta de sistema 1 se insertaría una tarjeta fcal, de esta manera en el servidor tendríamos dos tarjetas fcal, cada una de las cuales estaría conectada a los dos arreglos, esto tendría la misma función que las tarjetas SCSI, es decir sería con el propósito de tener redundancia por si alguna falla ocurre en una de las dos tarjetas o en la fibra que conecta lar tarjetas con los arreglos.

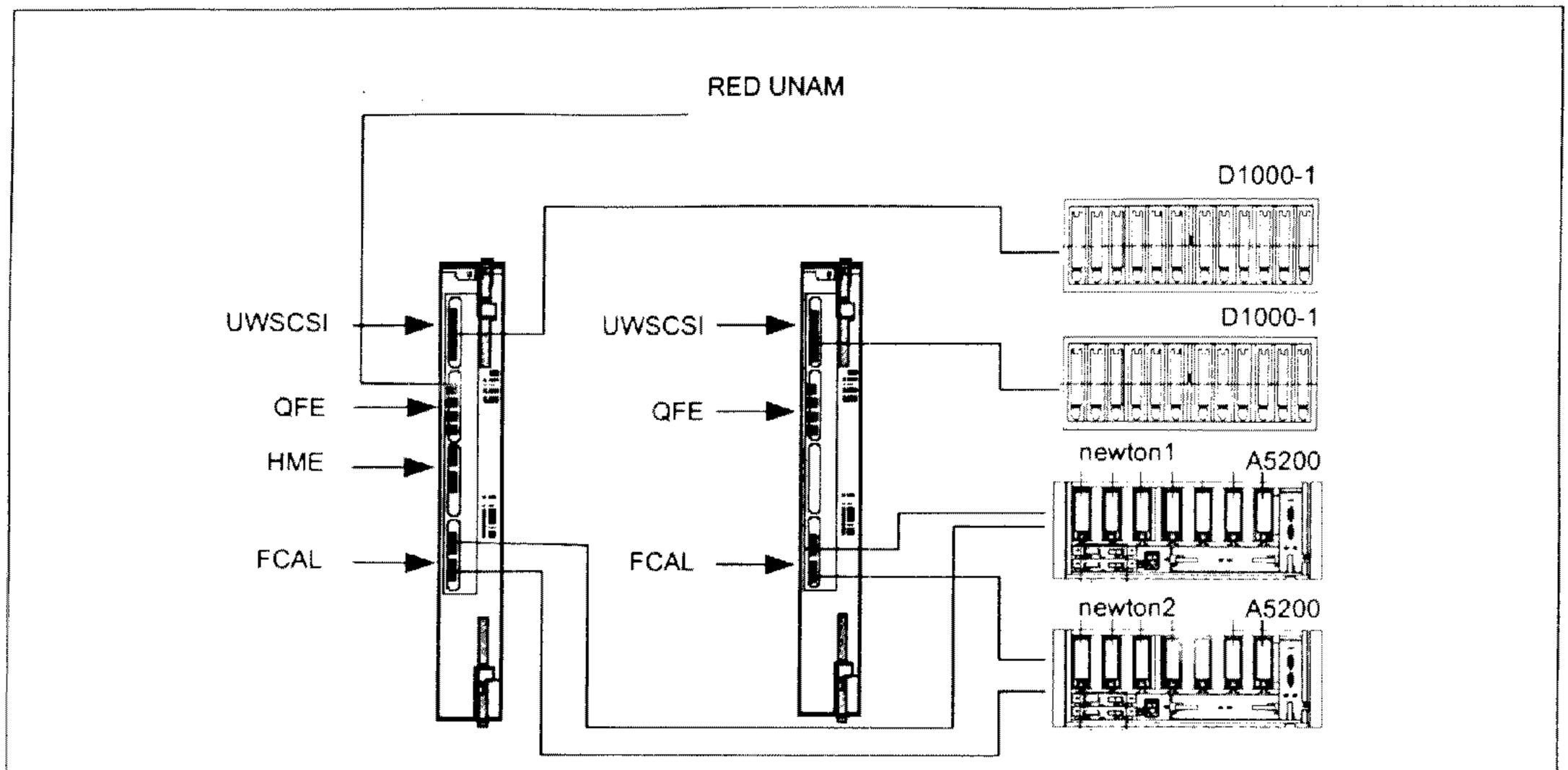


Figura 5.11. Distribución propuesta de hardware de newton y dragon

En lo que se refiere a como quedarían configurados los arreglos D1000 y A5200 tenemos lo siguiente:

De la configuración que se implementó en el servidor newton en el capítulo 4, sabemos que el sistema operativo se encuentra en el disco de la posición 0 del arreglo D1000 y su espejo se encuentra en el mismo arreglo pero en la posición 6, pero nosotros en la configuración que proponemos para el cluster incluimos un arreglo D1000 mas cada uno de los cuales está conectado a newton, para aprovechar esta redundancia lo que se haría sería deshacer los metaespejos de la configuración actual del disco D1000 y volverlos a crear, los primeros metaespejos seguirían estando en el disco actual de sistema operativo, pero los segundos metaespejos quedarían en el otro arreglo D1000, es decir quedarían en controladoras diferentes y de esta manera tendríamos redundancia por si alguna falla ocurriera con los arreglos o las tarjetas mediante las cuales se conecta al servidor. Además se configuraría un disco alterno que podría ser en la posición 6 del segundo arreglo D100. Los procedimientos para hacer esto serían los mismos que se usaron en el capítulo 2 y 3 para espejear los discos de sistema operativo y crear el disco alterno, solo cambiarían los discos. Veamos la siguiente gráfica.

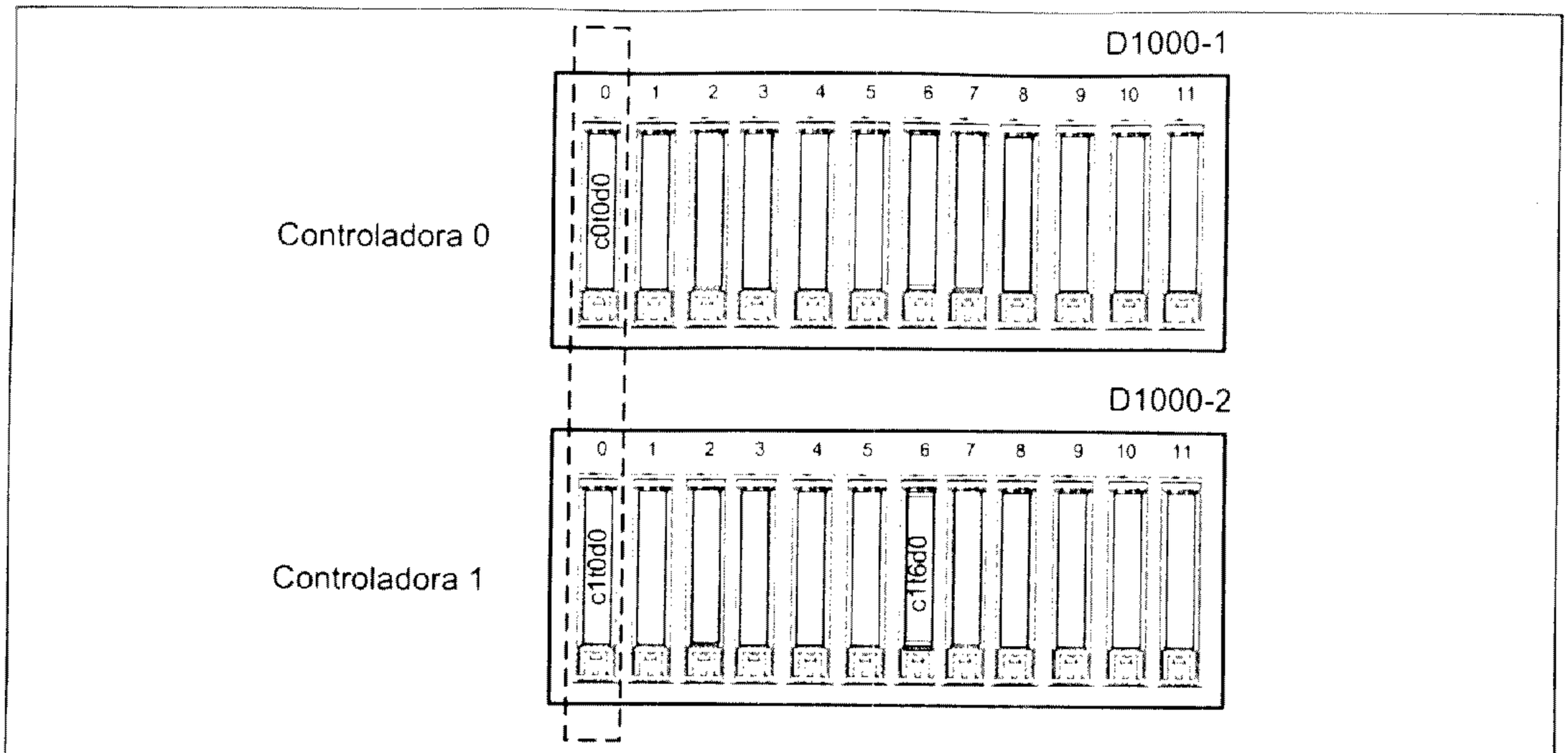


Figura 5.12. Discos del arreglo D1000 del servidor newton

Para los arreglos A5200 sabemos que la reestructuración que se hizo en el capítulo cuatro dejó distribuida la información de las aplicaciones en volúmenes con un plex a través de nueve discos, usando uno de los nuevos arreglos A5200 adquiridos y sus 12 discos lo que se haría primero sería acomodar sus discos de la manera que muestra la siguiente gráfica, después conectar el arreglo al servidor mediante unta tarjeta fcal y darlos de alta en volume manager, una vez hecho lo anterior crear el segundo plex para cada volumen solo que ahora la banda de datos se distribuiría a través de 11 discos, el disco de la posición 0de la parte trasera se ocupará como disco de reserva. Una vez terminada la creación del segundo plex desasociar el primer plex, desconectar el arreglo A5200 al cual ahora llamaremos newton2, agregarle 2 discos mas para que quede de 12 y distribuirlo de la misma manera que el arreglo newton1, posteriormente meter los discos del arreglo newton2 bajo el control de volume manager y crear el segundo plex. Con esto ya estaríamos ocupando los dos arreglos A5200 (newton1 y newton2), tendríamos entonces los mismos volúmenes que teníamos anteriormente con la misma información solo que ahora con dos plexes en cada uno de ellos y sobre discos diferentes.

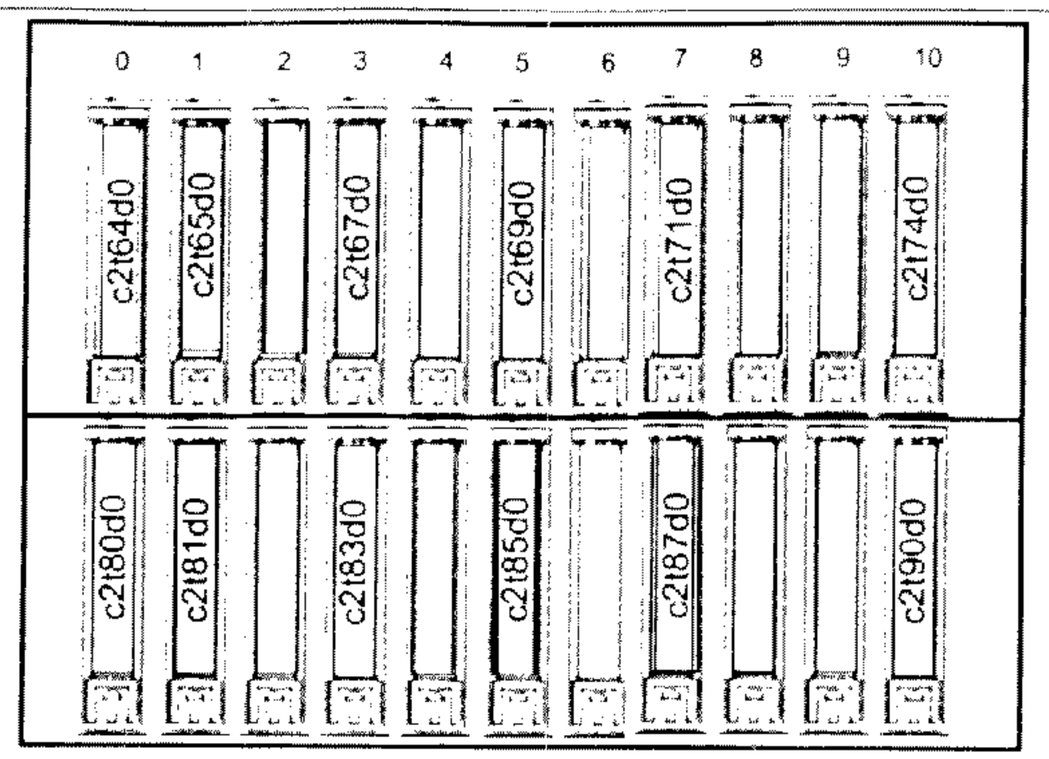


Figura 5.13 Distribución de discos del arreglo newton1

El siguiente paso sería la instalación de las aplicaciones y del software de Veritas Global Cluster, aquí sería necesaria la participación tanto de los administradores de dragón como de los proveedores de software para que llevarán esto a cabo y una vez realizado se harían una serie de pruebas preferentemente durante la noche las cuales consistirían en quitar de producción dragón y poner en producción newton, esto se haría con la finalidad de comprobar que realmente funcionaría newton como servidor web cuando ocurriera un failover.

Después de que las pruebas se hubieran hecho, entonces se programaría una fecha determinada y durante la noche se pondría en producción newton como servidor web y se haría la reestructuración de hardware en dragón, de tal forma que tanto tarjetas de sistema, tarjetas de entrada salida, distribución de discos y todo lo demás quedara exactamente igual que en newton.
A continuación mostramos una imagen de cómo quedaría configurado el hardware del cluster del servidor web de la UNAM.

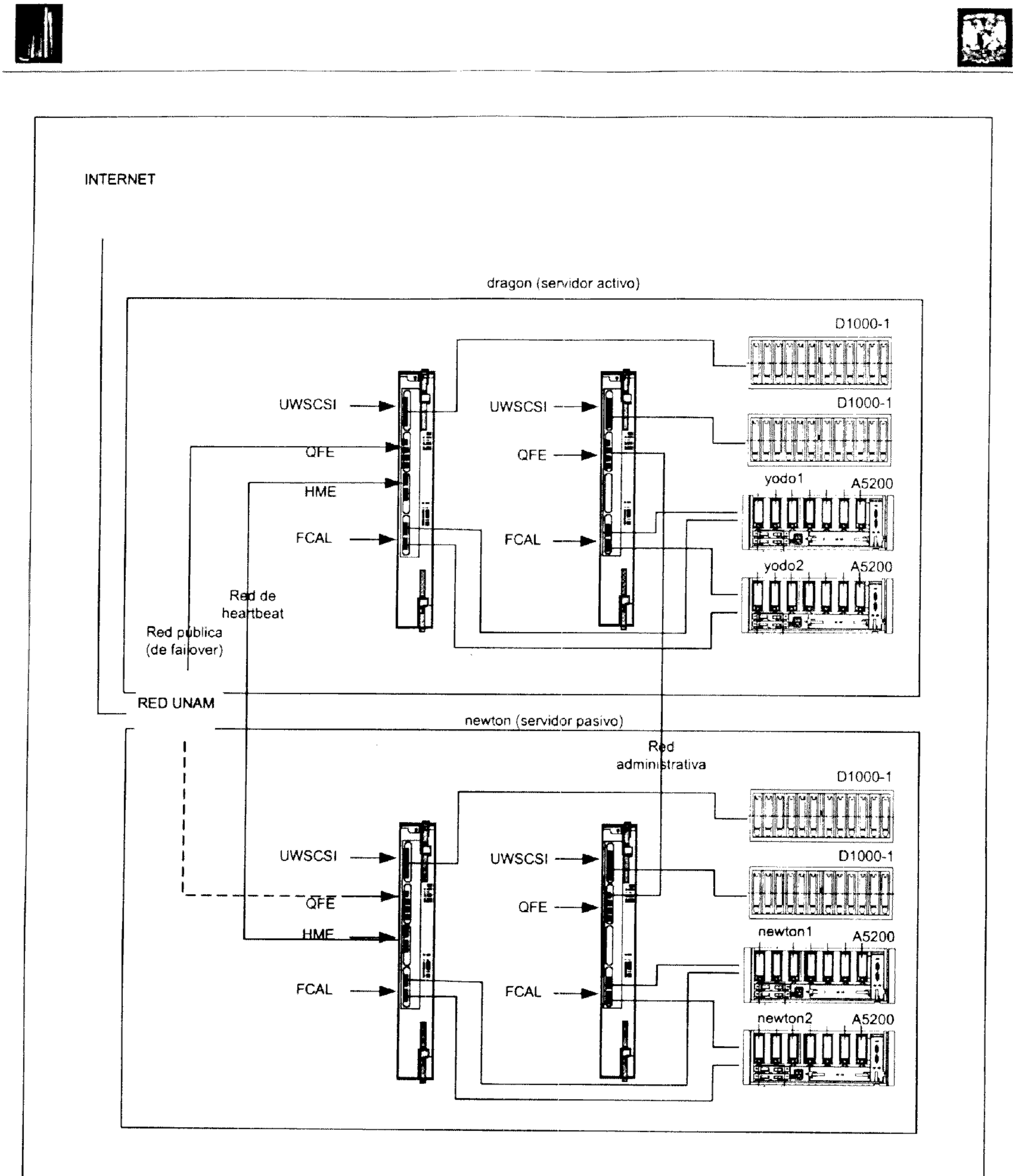


Figura 5.14.Propuesta de cluster del servidor WEB de la UNAM

Bibliografía

[1] Blueprints for High Availability: Designing Resilient Distributed Systems by Evan Marcus, Hal Stern, 2dn Edition 2003, Ed. Wiley

[2] Sun Cluster 3.0 Administration ES-333. Student Guide. Sun Microsystems. July 2001, Revisión B.

[3] Veritas Cluster Server for UNIX, Fundamentals. Bilge Gerrits, Siobhan Seeger, Dawn Walker. Veritas Software Corporation, 2003.

[4] Veritas Foundation Suite for Solaris: Administration and Troubleshooting, Volume I. Jade Arrington. Veritas Software Corporation,2002.

[5] Veritas Foundation Suite for Solaris: Administration and Troubleshooting, Volume II. Jade Arrington. Veritas Software Corporation,2002.

[6] Veritas Foundation Suite for Solaris: Administration and Troubleshooting, Volume III. Jade Arrington. Veritas Software Corporation,2002.

[7] Fundamentals of Solaris 8, Operating Enviroment for System Administrators, SA-118. Student Guide. Sun Microsystems. June 2001, revision B.1.

[8] Solaris 8, operating Environment, System Administration II, SA-288. Student Guide. Sun Microsystems. August 2001, Revisión B.2.

[9] High Availability: Design, Techniques and Processes by Floyd Piedad, Michael Hawkins Ed. Prentice Hall, 2001.

[10] Mission Critical Systems Management by Yuval Lirov Ed. Prentice Hall PTR; 1st edition.

[11] IT Systems Management: Designing, Implementing, and Managing World-Class Infrastructures by Rich Schiesser Ed. Prentice Hall, 2002.