33

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

LA ALTA DISPONIBILIDAD EN EL SISTEMA OPERATIVO WINDOWS NT UTILIZANDO LA TECNOLOGÍA CLUSTER

TESIS

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A :

ARTURO DÍAZ ARCEO

DIRECTORA:
ING. LAURA SANDOVAL MONTAÑO

MÉXICO, D.F. 2002

PAGINACIÓN DISCONTINUA

AGRADECIMIENTOS:

DEDICO ESTA OBRA A TODAS LAS
PERSONAS QUE ESTÁN, ESTUVIERON
Y ESTARÁN EN MI VIDA.

PORQUE POR ELLOS FUÍ, SERÉ
Y SOY LO QUE AHORA SOY.

INTRODUCCIÓN

En la actualidad, debido al gran desarrollo tecnológico, es ya indispensable para todas las compañías contar con un sistema de cómputo. La mayoría de los negocios cuentan con al menos una PC, y si hablamos de grandes empresas, todas requieren de una estructura de red, y de un sistema seguro en donde almacenar la información. También es prácticamente indispensable que quien adquiere una PC o quien ya la tiene, cuente con alguna versión de Windows. Esto ha ocasionado que el ambiente de Windows sea muy familiar al usuario, y como consecuencia desde que Microsoft propuso su versión de Windows para estaciones de trabajo y servidores, muchas compañías lo han adoptado como base de su sistema operativo. La versión de Windows NT 4.0 es la más difundida e implementada en una gran cantidad de empresas. Muchos otros negocios tienen una combinación de sistemas operativos como UNIX, Novell o Windows. Ésta es la razón por la cual elegí el sistema operativo Windows NT para realizar el estudio de la configuración de la tecnología de cluster.

El objetivo es estructurar un documento que defina de manera clara qué es un Cluster, qué elementos requiere para un buen funcionamiento y cuándo es conveniente contar con dicha tecnología si se tiene como sistema operativo Windows NT.

El capítulo I describe cómo ha ido creciendo la importancia de proteger la información y cómo ha evolucionando la tecnología para poder satisfacer las necesidades de los negocios, sus usuarios, clientes, etc. y comprender la posición del cluster. Se mencionan también algunas tecnologías de cluster de diversas compañías y las características que se requieren para evaluarlos.

Posteriormente, en los capítulos II y III se describe de manera general lo que es la tecnología de cluster. Encontraremos aquí la definición de cluster, así como los términos frecuentemente utilizados al hablar de dicha tecnología. También se analiza el funcionamiento del cluster, al igual que los componentes necesarios (hardware y software) para llevar a cabo una instalación del sistema con base en esto se describen las características tomadas en cuenta para clasificar los cluster así como una introducción para comprender las diferentes configuraciones que existen al implementarlos.

En el Capitulo IV, me enfoco a explicar con detenimiento el cómo hay que construir un cluster bajo el sistema operativo Windows NT 4.0 Enterprise Edition; aquí se detalla cómo se estructuran las posibles configuraciones de hardware (paso a paso), para posteriormente instalar el software de cluster, y por último describo cómo hay que instalar y configurar los grupos y recursos del cluster. También describo a detalle cómo es que funciona este software sus características y sus limitaciones.

CAPÍTULO I.- ANTECEDENTES

Existe en los negocios un factor importante que ya es considerado como mercancía de gran valor, y es la información. Ésta se ha convertido en materia prima de las empresas aunado a que ha ido adquiriendo mayor poder hasta convertirse en un aspecto determinante en la competencia empresarial.

Durante los últimos años ha habido un impresionante crecimiento de las tecnologías de información. A principios de la década anterior no existía el cómputo de escritorio y ahora el sistema cliente/servidor ya es una parte del trabajo normal. Basándonos en esta evolución debemos enfocar las estrategias de mercado y proponer soluciones informáticas que se ajusten a la forma de hacer negocios mediante el uso de la información en general como en la cultura empresarial.

1.1 EVOLUCIÓN TECNOLÓGICA

1.1.1 TECNOLÓGICA DE INFORMACIÓN

La tecnología de la información (IT) ha sido diseñada para servir a clientes en sus requerimientos de procesamiento de información, por tal motivo está relacionada con el desarrollo de *clusters*.

Un *cluster* puede definirse como un conjunto de máquinas que virtualmente trabajan como si fueran un solo servidor.

Un fenómeno interesante dentro del mercado de la IT es que cada nueva generación en la arquitectura de cómputo no parece desplazar totalmente a los sistemas existentes. Cada nueva tecnología es típicamente empleada para resolver una serie de requerimientos del cliente. A continuación mencionaré algunas de estas tecnologías.

Los *mainframes*, por ejemplo, representan el ambiente tradicional de procesamiento centralizado donde los usuario emplean terminales "tontas", este término se debe a que todos los cálculos y otros procesos (como son búsquedas, validación de datos, seguridad de datos, etc.) se llevaban a cabo en el *mainframe* al que estaban conectadas.

Cuando surgieron las minicomputadoras, el poder de cómputo se trasladó más cerca de los usuarios, ya que éstas podían ser situadas en un departamento individual o como parte de un grupo con ciertos requerimientos de procesos. A diferencia de los *mainframe*, los cuales se encontraban en un ambiente aislado y eran compartidos por toda la organización.

Las estaciones de trabajo y las computadoras personales trajeron consigo el "poder del *mainframe*" directo a los usuarios. Aunque limitados en el poder de procesamiento, estos dos sistemas brindan al usuario acceso inmediato e ininterrumpido tanto al CPU como a las aplicaciones personales.

Con todos estos distintos tipos de poder de procesamiento instalados en una organización, las compañías requerían hacer un sistema más rápido uniendo todo a través de una red, ya sea Red de Área Amplia (Wire Area Network, WAN), Red de Área Local (Local Area Network, LAN), o la combinación de ambas.

Con la arquitectura de cliente/servidor estos diversos sistemas trabajan en conjunto para proveer un mejor desempeño en el ambiente de la IT (tecnología de la información).

1.1.2 ARQUITECTURA CLIENTE / SERVIDOR

La arquitectura de cliente/servidor está más cerca de mantener el poder de proceso al nivel de usuario mientras maximiza el desempeño de una colección de sistemas instalados.

La arquitectura cliente/servidor tiene las siguientes características:

- Es una repuesta a las necesidades de la información tecnológica
- Esta diseñada para dividir el trabajo de cómputo entre dos o más computadoras.
- Tanto los componentes de hardware como de software son compartidos y/o distribuidos a través de la red.
- Este esquema de distribución permite a un usuario o un grupo de trabajo, contar con las capacidades de un *mainframe* o minicomputadora pero a un menor costo

1.2 SISTEMAS SEGUROS

A raíz de los problemas de seguridad que se presentaban en las empresas con sistemas de cómputo, la alta dependencia sobre los mismos y a la necesidad de evitar problemas a causa de la seguridad, se creó un área dedicada a atender este requerimiento.

El objetivo principal fue el evitar desastres y llevar a cabo labores de recuperación en caso de alguna caída en el sistema, así como asegurar que todas las aplicaciones se mantengan siempre al mismo nivel. Lo importante es que las posibles caídas no afecten al cliente, los usuarios deben ser los menos culpables de cualquier falla tecnológica.

La alta disponibilidad en los sistemas de cómputo es un requerimiento para las empresas de hoy en día, si lo que se busca es reducir significativamente los costos de tener un equipo fuera de funcionamiento.

1.2.1 *DOWNTIME* Y CAUSAS

Downtime: Tiempo que un servicio de cómputo no se encuentra brindando servicio, por diversas causas.

Muchas aplicaciones, como pueden ser bases de datos y sistemas de servicio (por ejemplo, servidores de archivos, servidores de correo electrónico, etc.), son críticas para los negocios, por tal razón se desea proveer acceso ininterrumpido y consistente a los servicios y a los datos contenidos en disco dentro de un ambiente de red. Sin embargo, cualquier sistema operativo puede presentar fallas y haber pérdidas de información en cualquier momento. Las causas son muy variadas, pueden ser provocadas por temblor, incendio o simplemente por una baja programada o no programada del sistema que provee los recursos.

La siguiente lista muestra las razones más comunes por las que se presentan fallas en los sistemas que representan pérdidas en tiempo:

- Falla en hardware o software, caídas del sistema que interrumpen el acceso a la red, a los discos y aplicaciones que el sistema proporcionaba.

- Operaciones de administración como son instalaciones, mantenimiento y respaldos que pueden causar que el sistema no brinde los datos y aplicaciones.
- Fallas de red y de I/O (Input/Output Entrada/Salida) que provocan que los datos y aplicaciones no se encuentren disponibles.
- Error del operador.
- Desastres metropolitanos (terremotos, incendios, etc.)

1.2.2 ALTA DISPONIBILIDAD

Un ambiente de servicio disponible (Available Server Environment, ASE) es un esquema integrado por sistemas y discos localizados en un *bus* compartido que juntos proveen datos y software altamente disponibles a los clientes del sistema. Este ambiente al mantener aplicaciones y datos disponibles reduce significativamente el tiempo de *downtime* causado por fallas de hardware o software.

En un principio a este software se le llamó DECSafe ASE (Digital Equiptment Corporation Safe Available Server Environment) y más adelante se renombró como *TruCluster*. En él se configuran servicios para aplicaciones o discos de datos que se desea hacer altamente disponibles, el proceso de *failover del Cluster* mantiene dichas aplicaciones y discos independientes de la disponibilidad de cualquier sistema en particular. En sí, este software permite desmontar los dispositivos de un nodo que falló por medio del proceso de *failover*, ejecutando acciones preprogramadas, con el fin de evitar corrupción de datos, y posteriormente monta los dispositivos en otro nodo que contenga los requerimientos necesarios, en el cual el servicio continúa disponible al cliente. Las aplicaciones se instalan en cada nodo del *cluster* y los discos son compartidos, por lo que cualquier nodo podrá correr las aplicaciones y acceder los datos. Esto habilita a los clientes a tener virtualmente acceso ininterrumpido a los recursos.

El cambio de servicios de una máquina a otra del *cluster*, cuando no está involucrado un servidor de bases de datos, es de 15 segundos a un minuto, aproximadamente. Si existe una bases de datos, el tiempo dependerá del tamaño de dicha base.

El software de *Cluster* provee acceso *multihost* a los discos de tecnología SCSI y un mecanismo de *failover* para aplicaciones basadas en red y servicios del sistema. La *figura 1.1* muestra un ambiente de alta disponibilidad que incluye dos

nodos con dos *buses* SCSI compartidos, discos conectados a cada *bus* y dos interfaces de red. Cada nodo está corriendo dos servicios.

El software del *Cluster* puede detectar las siguientes fallas y/o eventos de hardware y software:

* Caídas del Sistema Operativo.
* Apagado o reinicio del sistema.
* Fallas de red, como son fallas en interfaces de red y desconexión de cables.
* Fallas de I/O, detección de falla en los controladores SCSI y desconexión de cables.

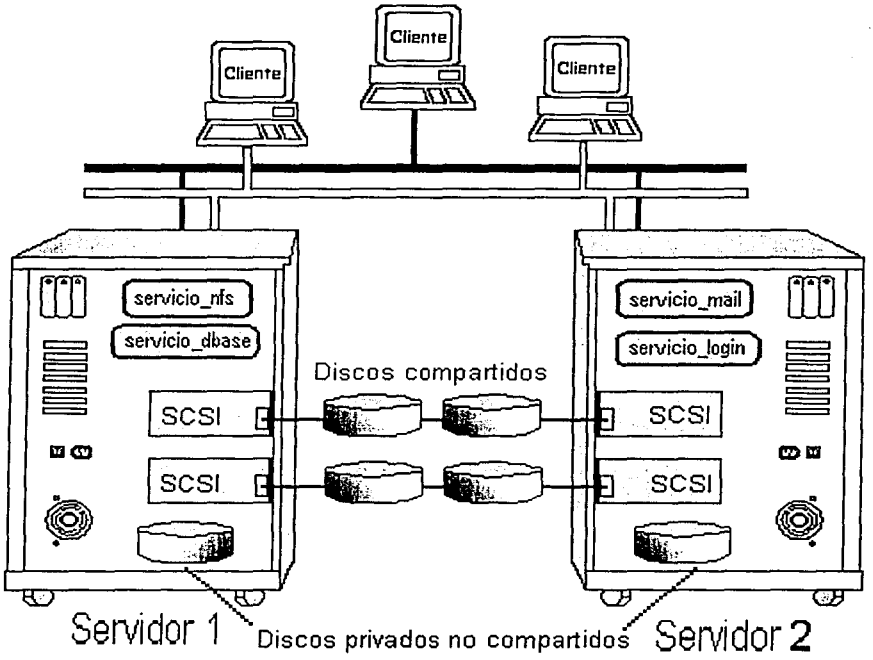


Figura 1.1 Visualización sencilla de un ambiente de alta disponibilidad.

Si uno de estos eventos ocurre el *Cluster* automáticamente tomará las acciones apropiadas para asegurar la recuperación de las aplicaciones y la continua disponibilidad hacia los clientes. Por ejemplo, si un sistema que está corriendo una aplicación falla, el software del *Cluster* se encargará de reiniciar la aplicación en un sistema viable y disponible.

1.3 MÉTODOS DE TOLERANCIA A FALLAS (*FAULT TOLERANCE*)

Se han introducido una gran variedad de tecnologías para satisfacer la creciente necesidad de servidores de alta disponibilidad. La tecnología más simple de alta disponibilidad que es el "espejo de discos" (disk mirror), como se muestra en la *figura 1.2*, duplica de manera continua toda la información escrita en disco, a un conjunto de discos espejo. Ésta es sin duda una buena protección de información, sin embargo no puede detectar todo tipo de fallas de hardware y software.

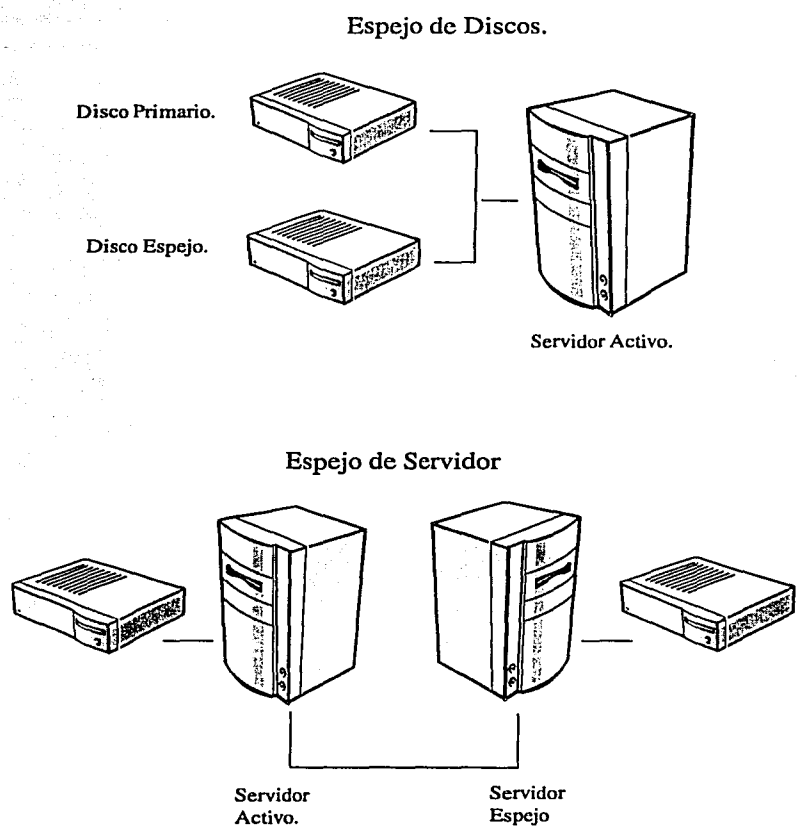


Figura 1.2 - Espejo de discos y Espejo de servidores.

Otra tecnología de alta disponibilidad es el espejo de servidor, en la *figura 1.2* se muestra un servidor espejo, como lo es el *Novell Server Fault Tolerance* (SFT), el cual, además de proteger la información, detecta fallas de forma automática al reiniciar aplicaciones seleccionadas. Esta tecnología presenta ciertos

inconvenientes como es que algunos proveedores imponen altos precios y debido a que su software trabaja sólo en sus plataformas, se convierten en vendedores únicos, además se requiere de un servidor "*standby*" que permanece inactivo hasta que el servidor principal falla.

En sistemas de "*Fault Tolerance*", como es el sistema *NonStop* de *Tandem*, detecta y recupera de manera casi inmediata en fallas sencillas de hardware o software. La mayoría de las transacciones en bancos corren en este tipo de sistemas. Estos sistemas también se encuentran a un costo elevado y cada solución se basa en un hardware de cierto vendedor.

Por último, hay otra tecnología que ofrece alta disponibilidad y protección de la información que es la de "*clusters*". Con un *cluster* es posible mantener aplicaciones de alta disponibilidad (Obteniendo hasta un 99.9% de servicio disponible), que gracias a su bajo costo comparado con implementaciones totalmente redundantes (*Stratus* y *Tandem NonStop*), son una alternativa inmediata para solucionar problemas de caídas de máquinas por exceso de carga, ofreciendo opciones de balanceo, así como bases para escalar el poder de cómputo. Los *clusters* no son recomendables para aplicaciones críticas como pueden ser los sistemas de tarjetas de crédito y tasación telefónica que requieren 100% de disponibilidad todos los días del año.

En el mercado varias empresas ofrecen software para implementar clusters, entre ellas se encuentran principalmente Hewlett Packard (HP), Data General, IBM, Tandem (recientemente adquirida por Compaq), Microsoft, que dio licencias del software de Digital para implementar *clusters* con Windows NT, y Digital, que es uno de los pioneros en el uso de *clusters* desde los 80's, iniciando primero en máquinas VMS y posteriormente integrando sus máquinas UNIX y más recientemente sus máquinas Windows NT. Dentro de las tecnologías de *Cluster*, encontramos las siguientes:

1.3.1 *TruCluster*

Adicionalmente al *TruCluster Available Server*, DEC liberó en 1996 una versión más completa denominada *TruCluster Production Server*, que ofrece la funcionalidad comentada anteriormente así como novedosas opciones de comunicación entre procesos de diferentes máquinas a través de la tecnología de Memory Chanel. Con Memory Chanel es posible crear áreas de memoria virtual para

paso de mensajes entre máquinas en forma eficiente, lo que sin lugar a dudas facilitará la implementación y mejorará el desempeño de manejadores de bases de datos paralelos que utilicen este esquema.

1.3.2 DEC DIGITAL CLUSTER FOR WINDOWS NT

Como se mencionó anteriormente el primer cluster que se desarrolló para Windows NT, fue creado por la compañía de *Digital Equipment Corporation* (ahora Compaq). Dicho *Cluster* está colocado en la cima de Microsoft Windows NT, comprende servidores duales activos que comparten una unidad de almacenamiento SCSI (*Small Computer System Interface*) y una GUI (*Graphical User Interface*) basada en la configuración de administrador de *Cluster*. Esta tecnología cuenta además con una segunda tarjeta de red dedicada a la conexión entre los dos servidores para evitar *failover* innecesarios y mejorar el desempeño cuando realmente haya una baja de uno de los servidores. Cuenta también con la opción de failback con la cual se migran las aplicaciones al servidor original una vez que éste se encuentra nuevamente en línea.

1.3.3 IBM PC SERVER

El IBM PC Server de alta disponibilidad provee *failover* entre dos servidores IBM PC (320, 520 o 720) con la misma configuración de almacenamiento de disco. Como *Wolfpack*, está diseñado para compartir dispositivos comunes de almacenamiento de disco.

1.3.4 TANDEM SERVERNET

El *Cluster* de *Tandem* está basado en el concepto de SAN (*System Area Network*). La SAN está diseñada para archivar comunicaciones de cualquier componente a otro a una alta velocidad, confiable e interconexión con baja latencia. El hardware de estándar industrial es usado en los sistemas junto con tarjetas de comunicación del propietario.

1.3.5 MICROSOFT NT CLUSTERING

Dave Custler es el arquitecto de Microsoft NT *Clustering*, así como también de Digital VMS primera plataforma de *cluster* usada en negocios hoy en día. Algunas de las características básicas de Windows NT *Clustering* son la capacidad de entrada a dominios, las herramientas administrativas de monitoreo de

multisistemas, el monitoreo del desempeño y la habilidad de rutear peticiones vía redirector.

Debido a su atractiva interfaz gráfica, muchas empresas poco a poco han comenzado a migrar sus aplicaciones al sistema operativo Windows NT, pero cabe mencionar que hoy en día este sistema operativo no puede considerarse como un sistema *enterprise*[1], debido a que le faltan varias funcionalidades para llegar a serlo, una de esas funcionalidades es la implementación de métodos para tener alta disponibilidad, los *clusters* vienen a ser los proveedores de esta función.

Hoy en día los problemas que encaran las empresas que cuentan con escasos recursos para su infraestructura de cómputo es el ir balanceando la alta disponibilidad para minimizar los costos asociados con elementos que permiten alcanzar altos niveles de disponibilidad.

Muchas organizaciones, como mínimo, optan por adquirir servidores basados en tecnologías que cuenten con Memoria de tipo ECC (Error Corrected Code), RAID (Redundant Array of Inexpensive Disk), componentes redundantes tales como fuentes de poder y ventiladores, componentes *hot swap*[2]. A través de un mínimo de inversión adicional, el nivel de disponibilidad que se puede llegar a alcanzar es de un 99%.

Existen empresas que requieren de tener cómputo ininterrumpido (*fault-tolerant*), tales como un monitoreo médico, control de tráfico aéreo, y centros de control militar. Soluciones de empresas como TANDEM, se encuentran disponibles y pueden reducir los tiempos de un equipo de cómputo fuera de servicio (*downtime*), a menos de una hora al año. Generalmente estas soluciones son de tecnología propietaria y extremadamente costosas; una solución de este tipo tiene como mínimo un costo aproximado de $500,000.00 USD.

A un costo similar, obteniendo casi los mismos niveles de disponibilidad, existen tecnologías de *clustering* disponibles de Digital para los sistemas operativos OpenVMS y Digital Unix. Estas soluciones son demasiado costosas para muchas empresas y aplicaciones.

---

[1] Enterprise: sistema empresarial.
[2] Hot Swap.- Se entiende como componentes Hot Swap, como aquellos componentes que pueden ser removidos de un servidor sin tener que apagar el equipo.

Los usuarios en la actualidad, deben de aprender a encontrar un balance adecuado entre la alta disponibilidad y los costos, sin que queden encasillados dentro de una solución propietaria. Para la mayoría de las organizaciones, la escasez de productos que cumplan con estas necesidades crea un gran *"abismo"* en el mercado.

Poder elegir la plataforma adecuada para los usuarios en este *"abismo"* es la causa por la cual un gran porcentaje de cuentas del mercado cliente/servidor está migrando de Novell a Windows NT. Hasta el momento, una deficiencia de Windows NT ha sido la falta de capacidades empresariales, tal es el caso de la opción de alta disponibilidad. La solución de *cluster* para Windows NT provee la capacidad necesaria de alta disponibilidad para comenzar a considerar a Windows NT como un ambiente empresarial y así comenzar a usarlo como una plataforma para aplicaciones criticas de negocios.

Como es sabido, Windows NT es una solución de arquitectura abierta. Se entiende como arquitectura abierta a la extendida disponibilidad de plataformas de hardware con diferentes tipos de procesadores (Alpha, Intel, etc.), opciones de almacenamiento y configuraciones de nodos, y aplicaciones de software de una gran gama de empresas.

La *figura 1.3* muestra las diferencias entre soluciones de alta disponibilidad, *fault-resilient y fault-tolerance.*

Figura 1.3 - Soluciones de alta disponibilidad, fault-resilient y fault-tolerance

Windows NT bajo una plataforma Pentium, Pentium Pro, Pentium II, Pentium III, Xeon ó Alpha es una solución altamente rentable para el cómputo empresarial, en comparación de otras opciones de alta disponibilidad.

Hoy en día, entendemos como disponibilidad convencional al típico servidor, sin muchas de las opciones de alta disponibilidad de hardware. Agregando a este servidor componentes de RAID, ECC, hot swap y redundancia, el nivel de disponibilidad crece a un rango del 99.5%, o tener el servidor fuera de servicio 40 horas al año.

A continuación se muestra una tabla que indica los niveles en que puede considerarse la disponibilidad.

Niveles de Disponibilidad

% Tiempo de recuperación	*Tiempo fuera max*	*Disponibilidad*
99%	3.5 días/año	Convencional
99.9%	8.5 horas/año	Alta-Disponibilidad
99.99%	1 houra/año	Fault Resilient
99.999%	5 minutos/año	Fault Tolerant

- Los productos de Windows NT Clustering proveen por lo general ***alta-disponibilidad*** lo cual, en el mejor de los casos con el hardware apropiado y redundancia de fuentes, puede conseguir niveles de ***fault-resilience***.

Un *cluster* se encuentra dentro de la categoría de alta disponibilidad la cual, en el mejor de los casos, puede proveer niveles de seguridad de un sistema *fault-resilient (Características de redundancia que evitan que existan una interrupción en el servicio.* Estas características ofrecen una excelente disponibilidad usando hardware redundante). Con lo que se reduce el *downtime* a 12 horas al año, y en algunos casos, a menos de una hora al año.

1.4 REQUERIMIENTOS DE USUARIO

Existen varios aspectos (requerimientos de usuario) que intervinieron para el diseño del *Cluster*. La meta del proyecto de *Cluster* era crear un producto el cual al presentarse una falla de hardware o software mantenga siempre accesible al servicio que está corriendo o es requerido. Por lo que el Cluster detecta y reconfigura dinámicamente al *host*, dispositivos de almacenamiento y fallas de red. Esto genera que el ambiente de *Cluster* sea aún más disponible al eliminar puntos de falla en el hardware.

Uno de los requerimientos más importantes era eliminar la posibilidad de corrupción de datos. En los sistemas simples se asumía que aplicaciones existentes en nodos no permitían que una instancia corriera en otros nodos que pudieran acceder una misma información. Si el acceso concurrente sucedía, la información podría corromperse. Debido a esto el primer objetivo es asegurar que la aplicación corra en un solo nodo a la vez.

Otro punto era el uso de estándares tanto de almacenamiento como de interconexiones para mejorar su función. Por tal motivo se implemento el uso de componentes de almacenamiento SCSI (el cual se definirá, más adelante).

Existen otros aspectos menos importantes que también influyeron al momento de estructurar las soluciones de *Cluster*. Una decisión que se tomo por razones de mantenimiento fue hacer del *Cluster* un producto con un mínimo impacto en la base de sistema operativo. Esto no quiere decir que no se requieran cambios en el sistema operativo para soportar el *Cluster*, pero sí que estos cambios pueden ser mínimos y sólo cuando sean necesarios.

Otra necesidad era que el *Cluster* soportara múltiples tipos de servicio (aplicaciones). Una de las más solicitadas por los usuarios era la de bases de datos, por lo que el diseño debía presentar mayor disponibilidad en el soporte de aplicaciones.

Cabe mencionar también, la necesidad de poder correr múltiples tipos de servicio de manera concurrente en los nodos del *cluster*. El *hot-standby*, el cual es otra solución de alta disponibilidad, requiere que los clientes adquieran sistemas adicionales que puedan permanecer inactivos durante la operación normal. Por lo que se pensó que el *cluster* debía ser capaz de permitir a todos los miembros correr aplicaciones de alta disponibilidad, así como también de poder usar la configuración tradicional (*hot-standby*).

1.5 BENEFICIOS DE UN *CLUSTER*

Algunos de los beneficios que brinda un *Cluster* son los siguientes:

1.5.1 INCREMENTA LA DISPONIBILIDAD

Los clusters incrementan la disponibilidad en varios aspectos, no sólo haciendo una réplica del medio de almacenamiento, sino también replicando el funcionamiento de la computadora (un CPU puede fallar, pero la cantidad de trabajo continuará operando en los CPUs restantes en el *cluster*). La característica de mayor atractivo en la tecnología de *cluster* es la alta disponibilidad de información y aplicaciones.

1.5.2 ESCALABILIDAD CON CRECIMIENTO SENCILLO

Al agregar componentes a un cluster, habrá un aumento con gran eficiencia en el desempeño y uso de los componentes. En un diseño sólido de *cluster*, debe ser posible agregar componentes sin pérdida de tiempo (*downtime*).

1.5.3 MANTENIMIENTO SENCILLO VS MÚLTIPLES SISTEMAS INDIVIDUALES

El *cluster* provee medios para administrar una serie de sistemas distribuidos como si fueran uno solo, además de que asegura la integridad de los datos a través de una extensa serie de sistemas y subsistemas.

1.5.4 COMPATIBILIDAD CON SISTEMAS NO-CLUSTER

Los clientes que tienen configuración de *no-cluster* pueden actualizar su máquina a configuración *cluster* sin la necesidad de adquirir aplicaciones nuevas o modificar sus aplicaciones.

1.6 REQUERIMIENTOS PARA EVALUAR UN *CLUSTER*

En el apartado anterior se mencionaron las necesidades de usuarios que se consideraron para el desarrollo del cluster. Además existen tres factores importantes para evaluar un sistema de computación que, debido a la alta dependencia de operar con IT (Tecnología de Información), muchos negocios los reconocen. Estos son los siguientes:

1.6.1 DISPONIBILIDAD Y ALTA DISPONIBILIDAD

"La disponibilidad se puede describir como la proporción de tiempo que un sistema puede ser usado para ser productivo en un trabajo"[3]. El no poder obtener aplicaciones importantes cuando son necesarias y/o la pérdida de datos impiden un mejor desarrollo y progreso de organizaciones.

Un término que está muy ligado al *cluster* es la alta disponibilidad, para comprender de manera sencilla lo que significa, veremos a continuación un ejemplo. Si dos sistemas miembro pueden acceder la misma información y uno de ellos falla, el otro sistema miembro debe estar en condiciones de acceder esa información, esto hace que la información tenga mayor disponibilidad para aquellas aplicaciones que la utilizan. A esta característica de mantener disponible la información después de una pérdida de comunicación entre los sistemas sobre un *bus* SCSI compartido se le llama alta disponibilidad.

Si aplicamos la alta disponibilidad en una máquina significaría mantenerla operando el mayor tiempo posible, adicional a la disponibilidad propia de la máquina.

Hablando en porcentajes, lo ideal sería que un sistema, en un período de un año, ofreciera el 100% de disponibilidad, es decir, que durante ese tiempo siempre estuviera activo. Si en el transcurso de un año un sistema sufre caídas o paros inesperados, la disponibilidad se reducirá proporcionalmente a los días no operables. Por ejemplo, si un sistema durante un año estuviera parado 2 semanas por fallas o causas imprevistas, entonces la disponibilidad sería de 351/365 = 96%. Los sistemas *stand-alone* (servidores independientes) ofrecen un 99% de disponibilidad. Esto puede parecer atractivo pero deja de serlo al saber que el 1% representa 3 días y medio al año. Esto sería suficiente para organizaciones con aplicaciones de uso casual.

El hecho de que existan máquinas o sistemas que se desea mantener en alta disponibilidad, se debe por lo general a que en ellas se corren aplicaciones muy sensitivas a caídas o fallos no deseados, y que en caso de que esto ocurra en los momentos más críticos, ya sea por segundos, minutos u horas, pueda representar pérdidas a la empresa. Si un sistema que corre varias aplicaciones falla y provoca corrupción de datos, el daño pudiese ser de consecuencias graves.

---

[3] Digital Product Training: Cluster fundamentals. Pag. 1-40

Las aplicaciones críticas, son aquellas de las que dependen los negocios, y deben estar disponibles cuando el usuario las necesite. Ejemplo de estas aplicaciones son los centros de llamadas de emergencia, control de tráfico aéreo, equipo médico, etc., que deben estar funcionando las 24 horas y todos los días del año. En tales casos, cualquier caída del sistema puede causar serios problemas ya sea de vidas, dinero y/o reputación. Para estas aplicaciones el ideal son sistemas de "*Fault-tolerance*" o "procesamiento continuo" los cuales utilizan redundancia extensiva y construcciones especializadas, para prevenir las interrupciones o fallas de servicio. Estos sistemas pueden proveer un 99.99% de disponibilidad, que sería aproximadamente 5 minutos al un año.

En general varios métodos diseñados para garantizar la disponibilidad continua utilizan alguna forma de redundancia: duplicando tarjetas de red, CPU, memoria, arreglos de discos (RAID) o utilizando servidores de bases de datos paralelos. El *cluster* es una forma de lograr alta disponibilidad, conectando dos o más máquinas (redundancia en los componentes como: tarjetas de red, procesador, memoria, etc.) que comparten un *bus* SCSI porque se conectan a dispositivos de almacenamiento. El *cluster* es una de las mejores opciones para mantener aplicaciones críticas en alta disponibilidad utilizando al máximo el hardware y aplicaciones de software.

Los ambientes de alta disponibilidad son ideales para clientes que pueden tolerar una caída del sistema de corto tiempo mientras se restablecen los servicios. Por ejemplo, para el control de vuelos se requiere de un *fault-tolerance*, ya que una caída del sistema pone vidas en peligro, mientras que para la reservación de boletos un sistema de alta disponibilidad sería lo más adecuado para mantener a los agentes vendiendo y contar con la satisfacción del cliente.

1.6.2 DESEMPEÑO / ESCALABILIDAD

Día a día podemos ver que la tecnología avanza a gran velocidad, lo que provoca que los sistemas de cómputo sean obsoletos en un periodo de corto tiempo. A medida que un negocio crece y requiere de un mayor número de aplicaciones en línea, la escalabilidad y el costo-beneficio se convierten en un factor importante para el éxito. Se puede decir que la escalabilidad, es la habilidad que tiene un sistema de mantener un alto desempeño conforme aumentan los requerimientos de éste. Esto implica que diversos componentes adicionales, como son dispositivos o

almacenamiento, puedan ser agregados al sistema y que tanto su desempeño y disponibilidad se incrementen proporcionalmente al crecimiento de dicho negocio.

En un sistema es importante considerar el factor de desempeño-escalabilidad debido a los siguientes puntos:

- Generar y mantener un alto desempeño es una de las principales características que cualquier usuario desea de un sistema.
- Después de hacer una inversión inicial en un sistema, el usuario se preguntará ¿qué tan sencillo y/o costoso será incrementar la capacidad de dicho sistema? ¿Por cuanto tiempo le brindará un alto desempeño?
- Como se mencionó en el párrafo anterior, la escalabilidad indica que es posible agregar componentes a un sistema para mejorar tanto el desempeño como la disponibilidad.
- Por último el costo-eficiencia, es decir; qué tan elevado será el costo y si bastará para un buen desempeño, pues es un factor importante para el crecimiento con éxito de una empresa.

Los usuarios suelen depender de determinado software, programas y aplicaciones para mantener en marcha sus negocios. Conforme el sistema crece, también se incrementa el interés del usuario en tener cierta protección de su inversión tanto de software como de hardware, lo que significa que al agregar componentes estos sean compatibles con el sistema instalado. Por tales motivos la escalabilidad se convierte en un punto clave en el criterio para cualquier evaluación de un sistema.

1.6.3 COSTEABLE O RENTABLE

Esto significa el costo total del sistema, se refiere especialmente a la protección de la inversión existente, es decir, poder agregar hardware y software compatible a un precio razonable. El costo de un sistema aumenta si este requiere del uso de: programas especiales, componentes específicos, alto costo en el mantenimiento y administración del sistema.

Los siguientes factores afectan directamente en el costo-inversión de un sistema:

- ✓ *Hardware, software y/o programación especializada o estándar.* Si un sistema tiene un grado elevado de especialización su costo se incrementará. De manera que los sistemas que utilizan componentes estándar y requieren de menor cantidad de software o programas especiales su costo será mucho más accesible.
- ✓ *Complejidad en la administración del sistema.* Sucede lo mismo que en el punto anterior, mientras más complejas sean las tareas de administración más elevado es el costo del sistema.
- ✓ *Costos de mantenimiento.* Si estos costos son bajos (mantenimiento de la tecnología embebida, capacidades de mantenimiento en línea, por ejemplo) el sistema generará una retribución con mayor rapidez.

Algo que parecería ser simple, como es agregar un software a un sistema, puede causar una gran variación de costo, tiempo y dinero en su implementación (por ejemplo una recopilación). Es por ello que estos tres criterios son críticos dentro del mercado de la tecnología de cluster.

CAPÍTULO II.- EL CLUSTER, CONCEPTOS GENERALES.

A partir de que la alta disponibilidad se convirtió en una característica fundamental en aplicaciones de misión crítica como lo son bases de datos y servidores de archivos, se comenzó una búsqueda para encontrar soluciones de alta disponibilidad en ciertos sistemas operativos. La interconexión de sistemas de almacenamiento compartido así como la interfaz SCSI dentro de un sistema operativo (UNIX, VMS o Windows NT) dan la oportunidad de hacer más disponibles los servicios basados en disco. Esta integración de sistemas de computadoras y discos externos a través de uno o más buses SCSI da origen al "*Cluster*".

2.1. ¿QUE ES UN CLUSTER?

Para comprender la forma en que opera el Servicio de Cluster junto con los demás sistemas que lo forman, el manejo de varios recursos y el Cluster en sí, existen ciertas características básicas que se verán a continuación.

Una definición muy general de un Cluster es la siguiente:

- Los *Clusters* son servidores:
  - "Un cluster es un grupo de servidores independientes que trabajan en conjunto y se presentan como un solo sistema dentro de una red." El manejo de los servidores que se encuentran en cluster se realiza como si fuera un solo sistema.
- Un *Cluster* está diseñado para realizar las siguientes tareas, a un precio relativamente bajo para el usuario:
  - Proveer cooperación entre sistemas.
  - Brindar servicios en forma rápida e ininterrumpida.
  - Maximizar el desempeño (eficiencia)
  - Minimizar el tiempo de caída del sistema (*downtime*)

Se puede decir que un *cluster* es una interconexión de sistemas, dispositivos de almacenamiento y otros periféricos, soportados por un software que integra todos estos componentes, formando un ambiente con las siguientes características:

- Cada máquina física en un *cluster* es considerada un sistema.
- Un *cluster* consiste en dos o más sistemas.
- Los CPU´s se encuentran separados así como los dispositivos de almacenamiento para compartir datos, archivos, recursos y aplicaciones.
- Alta disponibilidad para aplicaciones y datos
- Facilidad de crecimiento ya que es rentable y no rompe con las operaciones existentes de negocio.
- Protección de la inversión, asegura que los sistemas actuales y periféricos trabajarán con sistemas futuros.
- Incremento de productividad en cuanto a que sistemas múltiples trabajan como uno solo, dando así más poder de cómputo sin la necesidad de más administradores del sistema.
- Soporte de *failover* en caso de falla de alguno de los componentes del Cluster.

La mayoría de las máquinas que forman parte de los modelos de *cluster* se encuentran agrupadas bajo un nombre común, de tal forma que los clientes conectados a la red pueden hacer uso de los servicios disponibles del *cluster* de manera transparente.

Cada computadora que es miembro de un *cluster* es llamada nodo. Los nodos de un *cluster* deben de tener una red privada entre ellos, misma que tendrá como función el permitir la comunicación entre los nodos sin afectar el tráfico normal de red pública. Además los nodos deben de estar conectados a un dispositivo de almacenamiento compartido, en donde se guardará la información que va a compartir el *cluster*.

Los *clusters* son manejados como un sistema sencillo, desde el punto de vista de un administrador de sistema.

Un nodo de *cluster* puede ser un sistema uniprocesador o multiprocesador. Además cada nodo ejecuta una copia separada del sistema operativo y es aislado de las fallas tanto del software como del hardware de otros nodos del *cluster*.

2.2. FUNCIONAMIENTO DEL *CLUSTER*

Para conocer el *cluster* y tener alta disponibilidad es necesario contar con hardware y software especializado. Mediante el hardware se diseñan las conexiones del *bus* de datos que puede ser compartido por varias máquinas (hasta 16 en *clusters* de Digital) permitiendo que los dispositivos unidos a este *bus* estén disponibles para todas las máquinas.

El software es el que crea el ambiente de *cluster*, permitiendo que los clientes vean a éste como un sólo sistema. Los dispositivos de una máquina que fallan son desmontados por un proceso que se lleva a cabo gracias al software de control del *cluster* cuando se ejecutan acciones preprogramadas. También monta los dispositivos ordenadamente en otra máquina capaz de realizar el trabajo de la máquina que falló, siguiendo ciertas políticas, en donde el servicio continúa disponible al usuario final, el cual mantiene la misma forma de acceso que tenía antes de que hubiese fallado la máquina.

Dicho proceso de cambiar servicios de una máquina a otra en un *cluster* demora aproximadamente de 15 segundos a un minuto, si es que no existe una base de datos. En caso de que haya alguna base de datos en un servidor de alta disponibilidad, el tiempo de transferencia de servicios entre las máquinas dependerá del tamaño de la base de datos, si es de 10 a 15 gigabytes, dicho procesos se realizará en menos de 5 minutos aproximadamente.

El *cluster* está formado de varios demonios o servicios de sistema que corren en las máquinas que lo forman, y se encuentran monitoreando operaciones y enviando señales de alerta en caso de detectar fallas como las siguientes:

- ★ Caídas del sistema operativo
- ★ Falla de la fuente de poder (de una máquina con la aplicación crítica)
- ★ Fallas de las tarjetas de red
- ★ Fallas de dispositivos del bus de almacenamiento compartido.

2.2.1 COMPONENTES DE UN *CLUSTER*

Como se mencionó, un *cluster* es una configuración de dos o más nodos y se presenta al usuario como un solo servidor. Los nodos en un *cluster* están

conectados usando uno o más buses de almacenamiento compartido y una o más redes físicas independientes (en ocasiones son llamadas interconexiones). Se llama red privada cuando una red conecta únicamente a los miembros del *cluster*, sin incluir a los clientes. Dichos servidores sólo proveen un medio redundante para una comunicación *intracluster*. Una red pública es aquella que soporta clientes y miembros del *cluster*.

Cada nodo puede tener uno o más discos locales y cada medio de almacenamiento compartido puede contener uno o más discos. En estos medios de almacenamiento se encuentra información necesaria tanto para correr las aplicaciones de servidor en el *cluster* como por aplicaciones para administrarlo y por lo general utilizan la tecnología de bus SCSI. Cada disco del almacenamiento compartido le pertenece a sólo un nodo del *cluster*, dicha pertenencia puede moverse de un nodo a otro ya sea por que falle el nodo al que pertenecía por el administrador del sistema.

2.2.2 MODELO BÁSICO

La *figura 2.1* muestra la configuración básica de un *cluster* de dos nodos. Éste consiste en dos servidores, cada uno con su disco local y sistema independiente, conectados físicamente a una red y a un medio de almacenamiento SCSI compartido al que ambos sistemas tienen acceso. A dicho cluster tienen acceso múltiples clientes.

conectados usando uno o más buses de almacenamiento compartido y una o más redes físicas independientes (en ocasiones son llamadas interconexiones). Se llama red privada cuando una red conecta únicamente a los miembros del *cluster*, sin incluir a los clientes. Dichos servidores sólo proveen un medio redundante para una comunicación *intracluster*. Una red pública es aquella que soporta clientes y miembros del *cluster*.

Cada nodo puede tener uno o más discos locales y cada medio de almacenamiento compartido puede contener uno o más discos. En estos medios de almacenamiento se encuentra información necesaria tanto para correr las aplicaciones de servidor en el *cluster* como por aplicaciones para administrarlo y por lo general utilizan la tecnología de bus SCSI. Cada disco del almacenamiento compartido le pertenece a sólo un nodo del *cluster*, dicha pertenencia puede moverse de un nodo a otro ya sea por que falle el nodo al que pertenecía por el administrador del sistema.

2.2.2 MODELO BÁSICO

La *figura 2.1* muestra la configuración básica de un *cluster* de dos nodos. Éste consiste en dos servidores, cada uno con su disco local y sistema independiente, conectados físicamente a una red y a un medio de almacenamiento SCSI compartido al que ambos sistemas tienen acceso. A dicho cluster tienen acceso múltiples clientes.

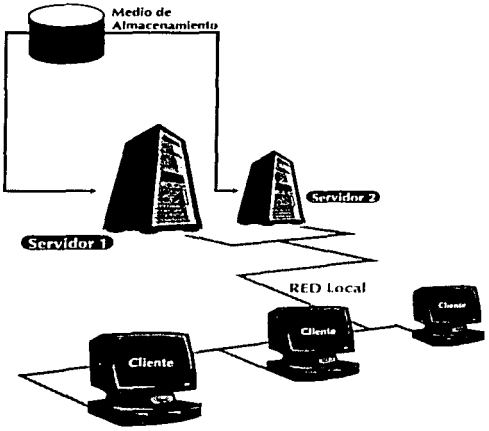


Figura 2.1 - Configuración básica de cluster, con dos nodos y un medio de almacenamiento compartido SCSI.

2.3 CONCEPTOS GENERALES DE LA TECNOLOGIA DE *CLUSTER*

2.3.1 DOMINIO DE ALMACENAMIENTO DISPONIBLE (*STORAGE AVAILABILITY DOMAIN*, SAD)

Un SAD o dominio de almacenamiento disponible, es un conjunto de nodos que pueden acceder dispositivos de almacenamiento comunes o compartidos en un Cluster. El dibujo anterior (figura 1) es un ejemplo de SAD. En él se incluyen las conexiones de hardware entre los nodos así como los dispositivos de almacenamiento y de red. En un *cluster* se pueden incluir varias redes pero sólo una es utilizada para la comunicación de su protocolo; en tanto que otras las pueden utilizar los clientes para el acceso a servicios del cluster. La interfaz de red puede ser cualquier estándar que soporte *broadcast* (por lo general Ethernet ó FDDI). Las

interconexiones del medio de almacenamiento pueden ser *single-ended o wide-differential*, éstos son controladores SCSI.

2.3.2 *CLUSTER* SIMÉTRICO VS ASIMÉTRICO

Existen dos medios por los que un *cluster* puede ser configurado respecto a los nodos y al medio de almacenamiento y es de manera simétrica o asimétrica. En una configuración simétrica (*figura 2.1*) todos los nodos están conectados a todo el medio de almacenamiento. Esta configuración simplifica la detección de partición de redes y previene la corrupción de información.

En una configuración asimétrica no todos los nodos se conectan a todos los dispositivos del medio de almacenamiento (*figura 2.2*). Esta configuración mejora el desempeño ya que si sólo algunos nodos están en un mismo bus, éste se saturará con menor facilidad. Otra ventaja es que se incrementa la escalabilidad de almacenamiento, pues permite una mayor capacidad de éste a través del bus SCSI.

2.3.3 ACCESO PARTICIONADO Y ACCESO COMPARTIDO

Un *cluster* se puede clasificar con base en la manera en que comparten dispositivos, sobre todo CPU y almacenamiento. Una es un cluster particionado o no compartido, el cual restringe el acceso de otros sistemas a los datos o al medio de almacenamiento, el disco es accedido por un solo servidor, los nodos requieren del servicio de petición de I/O, no es necesario "*lock manager*" y es indispensable la partición. El *cluster* de acceso compartido (multi-tailed) permite el acceso a todos los recursos localizados en el cluster, el disco es accedido por múltiples servidores, el controlador de discos requiere el servicio de petición de I/O, así como "*lock manager*" y no es necesaria la partición.

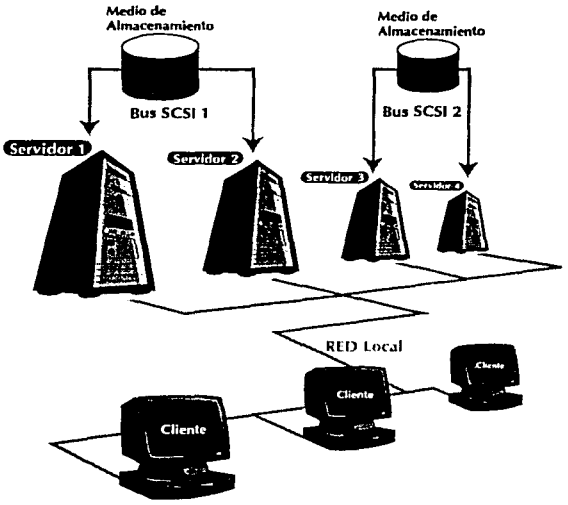


Figura 2.2 - Configuración de cluster asimétrica.

2.3.4 INVISIBILIDAD

Un elemento del *cluster* es invisible, es decir, que dé la apariencia y se trabaje en él como si fuera un sólo sistema. Hay tres niveles que están en contacto directo con el *cluster*, con base en los cuales es medida la invisibilidad, y son los siguientes:

Invisibilidad al usuario.

- El usuario ve al sistema como uno solo.
- La mayoría de las implementaciones se integran en este nivel

- El usuario siempre trabaja con una disponibilidad continua de servicio.
- Los usuarios no requieren entrenamiento para ser capaces de hacer uso de las ventajas del cluster.

Invisibilidad al administrador del sistema.

- Es más difícil la necesidad de ejecutar y coordinar la misma operación en una serie de diversos sistemas que actúan como servidores que en un grupo de usuario/cliente.
- La mayoría de las soluciones deben apuntar a mantener un cierto límite en la complejidad de la administración (que sea lo más parecido posible a la operación de un solo sistema)
- Un *cluster* puede discernir de otro en el número y calidad de herramientas disponibles para manejar las tareas de administración del sistema.

Invisibilidad en aplicaciones.

- Nivel más difícil de ejecutar debido a la compleja tarea de difundir una aplicación a través de diferentes y numerosos sistemas y coordinar su simultánea operación
- Por lo general requiere de una primitiva y compleja sincronización, como es un *Distributed Lock Manager.*
- La mayoría de las implementaciones en el mercado tratan de ignorar este nivel de invisibilidad, debido a que tienen que trabajar nuevamente con las aplicaciones para que éstas sean invisibles.

Todas las implementaciones en el *cluster* deben cubrir esta particularidad en los distintos niveles.

2.3.5 SERVICIO

Un servicio se puede describir como los componentes necesarios para el funcionamiento de una aplicación o una parte de ella. Por lo general está formado por archivos del sistema en donde residen datos, aplicaciones y "scripts" de control que indican las acciones que se ejecutarán al inicio, durante y al finalizar un servicio. Esta serie de programas o procesos necesitan ser ejecutados secuencialmente para iniciar o detener un servicio. Si alguno de éstos no es

ejecutado de manera apropiada, el servicio no podrá iniciar ni detenerse. Un servicio define uno ó varios programas que se hacen altamente disponibles. Ejemplos de servicios en Cluster son NFS (Network File System) y SQL (System Qualified Library).

2.3.6 EVENTOS Y MODOS DE FALLAS

El *cluster* monitorea el software y hardware para determinar el estado del ambiente. Un cambio en el estado es reportado como una notificación de evento por el software del cluster. Ejemplos de estos eventos son falla y recuperación de *host*, fallas en la red o en dispositivos de disco o en algún comando de las herramientas de administración del *cluster*.

2.3.7 SERVICIO DE *FAILOVER*

El software de *Cluster* responde a los eventos relocalizando servicios de un nodo a otro. Una reubicación de un servicio debido a una falla de hardware se conoce como *servicio de failover*. Existen además otros motivos para hacer relocalizaciones, que se verán más adelante.

2.3.8 POLÍTICA DE RELOCALIZACIÓN DE SERVICIOS

Siempre que un servicio tenga que ser relocalizado, el *cluster* usa políticas configurables para determinar qué nodo es el adecuado para correr ese servicio. Las políticas se definen en función a eventos y preferencias del administrador del sistema instaladas para cada servicio. Por ejemplo, si un servicio corre en un nodo y éste falla, el servicio tiene que ser relocalizado en otro nodo que debe ser especificado por el administrador del sistema. También se puede dar preferencias para que un nodo proceda a hacer una recuperación del servicio, por ejemplo, el administrador del sistema puede definir que un servicio siempre regrese al nodo donde corría originalmente, una vez que está disponible, después de una falla, a esto se le llama "*fail back*". Por lo general, en servicios que tardan demasiado en iniciarse, el administrador del sistema define que la relocalización de ese servicio sólo se haga en caso de que el nodo falle.

2.3.9 CONTROL CENTRALIZADO VS. DISTRIBUIDO

El software de *cluster* es una colección de servicios de sistema (procesos independientes que corren en background al nivel de usuario) y *scripts* que corren en todos los nodos en un SAD. En el diseño distribuido el software en cada nodo participa en determinar dónde se va a colocar un servicio. En el diseño centralizado sólo uno de los nodos es responsable de determinar la política.

2.3.10 PARTICIÓN DE RED DETECTABLE VS PARTICIÓN TOTAL NO DETECTABLE

Una partición de red detectable se presenta cuando dos o más nodos no se pueden comunicar a través de su red pero pueden acceder al almacenamiento compartido. Esta condición puede causar corrupción de datos si cada nodo reporta que los demás nodos están fuera del sistema. Otro problema es que cada nodo puede intentar adquirir el servicio, el cual puede correr de manera concurrente en múltiples nodos y esto posiblemente corrompa el almacenamiento compartido.

En un *cluster* existen distintos mecanismos para prevenir la corrupción de datos al haber una partición de red. Uno de ellos es el estado de comunicación sobre el bus SCSI. Al detectar una partición de red, el cluster verifica que exista comunicación a través del bus SCSI previniendo así instancias múltiples del servicio. Por otra parte, cuando no hay comunicación a través del bus SCSI, el *cluster* lo transmite como una desconexión eléctrica propia del bus. En este caso, tanto para el servidor 1, como para el servidor 2 será imposible acceder al almacenamiento compartido en ese *bus*.

Como medida de seguridad, el *cluster* aplica reservaciones de dispositivos (*hard locks*) en el disco. Los *hard locks* son mecanismos extra de prevención a fallas, que rara vez son necesitados. Por ejemplo, si una aplicación puede acceder al medio de almacenamiento que necesita para correr, está permitida a hacerlo. Cuando existen más de dos servidores en un *cluster* se requerirá que un porcentaje (por lo general más de la mitad) de los servidores esté disponible para su operación correcta. Para una configuración de dos servidores si un nodo falla el otro debe continuar con la operación.

Existe un caso muy inusual en el cual los datos pueden corromperse, éste sucede cuando hay una partición total durante una copia espejo del medio de

almacenamiento. El espejo de discos replica de manera transparente la información en uno o más dispositivos de almacenamiento. En una partición total, dos servidores no pueden comunicarse a través de la red y tampoco existe comunicación entre ellos mediante el bus SCSI, de manera que podría suceder que un servidor pueda acceder a una serie de discos y el segundo servidor a otra serie (como se ve en la *figura 2.3*). A pesar de que esta situación no permite un acceso común a discos, es posible que el medio de almacenamiento que estaba siendo replicado se haya corrompido. Cada servidor tomará al otro como fuera del sistema por que no existe un medio de transferencia para comunicación, por lo que, si un servidor tiene acceso a la mitad de los discos replicados y el otro servidor tiene acceso a la otra mitad, el servicio podrá correr en ambos servidores. De tal forma que cuando se quieran volver a unir las dos mitades el conjunto de discos estará fuera de sincronización, causando la corrupción de datos. Debido a esto se implementó en el *cluster* una política opcional para que no pueda correr un servicio si no existe una copia de espejo completa o réplica total disponible.

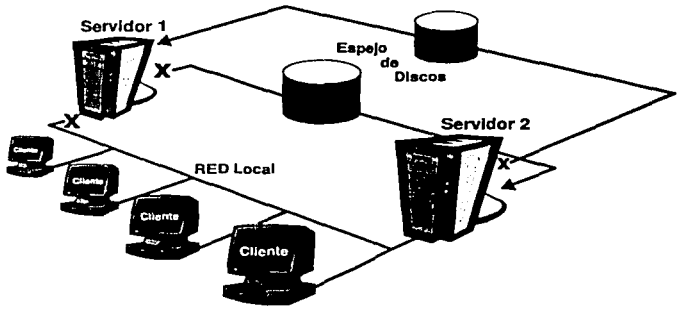


Figura 2.3 - Partición total no detectable.

2.3.11 ADMINISTRACIÓN DE SERVICIOS

El administrador de servicios del *cluster* realiza las funciones siguientes: Inicialización del servicio, monitoreo del SAD y relocalización de servicios. El administrador de servicios pide información como son: el tipo de servicio, archivos de sistema y discos que requiere. Con la cual genera una secuencia de comandos que iniciaran un servicio, integrando subsistemas complejos en un servicio sencillo.

Hay tres tipos de servicio importantes dentro del *cluster*:

* ***Servicio de usuario***: provee alta disponibilidad a programas de usuarios que no dependen de almacenamiento en discos, simplemente requieren que los *scripts* de usuario sean ejecutados en alguno de los nodos. Por ejemplo, el programa "*login*" se puede definir en alta disponibilidad, si queremos permitir el acceso al sistema a un solo usuario, por lo tanto se habilitará en un nodo y sé deshabilitará en los demás. Si el nodo donde corre "*login*" llega a fallar, el servicio se migrará a otro nodo, programando los "*scripts*" necesarios, para seguir ofreciendo el servicio.

* ***Servicio de disco***: es un servicio de usuario que provee alta disponibilidad a servicios que dependen del almacenamiento de datos en discos, es decir, información de archivos de sistema y disco. Una aplicación de este servicio son los manejadores de bases de datos, que necesitan tener sus bases siempre accesibles.

* ***Servicio NFS (Network File System)***: provee alta disponibilidad de acceso a sistemas de archivo NFS, es una versión especial del servicio de disco que pide información específica que es relacionada al NFS, como sería la exportación de información. Este servicio por lo general se relaciona a una dirección IP, en donde los requerimientos que llegan a dicha dirección IP los atiende cierto nodo. Si el nodo falla, el usuario seguirá buscando la misma dirección, solo que lo atenderá otro nodo.

El monitoreo indica tanto el estado de un nodo como de un servicio, es decir, si está corriendo o no y en dónde. La relocalización permite al administrador del sistema mover los servicios manualmente con sólo especificar la nueva localización.

2.3.12 ESPEJO DE SOFTWARE

El hacer un espejo de software es un mecanismo en donde se hace una réplica de la información a través de dos o más discos, de tal forma que si un disco falla, la información puede ser accedida en otro disco.

2.4 CONCEPTOS DE SOFTWARE

Los términos a continuación descritos, son los de mayor importancia en la tecnología de *cluster* ya que lo diferencian de otras implementaciones cliente/servidor:

- *Failover*
- *Fault Tolerance*
- *Lock Manager*
- Paralelización

2.4.1 *FAILOVER*

El *cluster* utiliza un proceso llamado "*failover*" (continuar trabajando) para proveer alta disponibilidad de servicios a los clientes sobre la red en caso de alguna falla. Se entiende como "*failover*" al proceso de transferir el control de uno o más servicios de cliente (aplicaciones, discos, impresiones, etc.) de un nodo a otro.

El tiempo total y los eventos específicos asociados con el proceso de *failover*, dependen del tipo de *failover* que se presenta. Éstos pueden ser:

- Migración de recursos o *failover* voluntario. Éste lo realiza el administrador del sistema, dando de baja intencionalmente algún miembro del *cluster*.
- *Failover* involuntario o no planeado. Éste ocurre al presentarse una falla de hardware o software en el *cluster*.

Un failover involuntario es iniciado por el cluster cuando detecta una falla en alguno de sus nodos, cuando esto ocurre, ciertas aplicaciones que estaban siendo utilizadas en el nodo que falló, hacen un *failover* a otro nodo disponible del *cluster*. Al presentarse este proceso, el cliente no ve ningún cambio en la actividad, o

probablemente necesite reconectarse al sistema dependiendo del tipo de servicio que está utilizando. En la *figura 2.4* se muestra el proceso de *failover*, donde el servidor 2 falló, provocando la migración de los servicios de archivos e impresión (*file/print*) del grupo 2 al servidor 1.

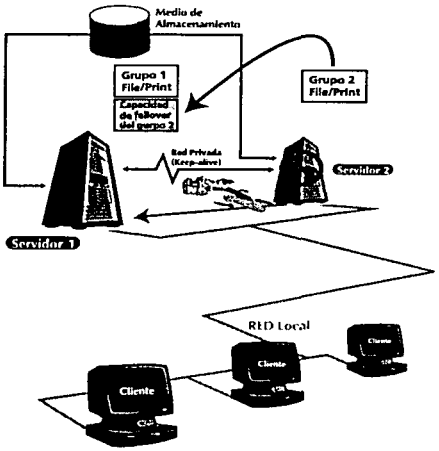


Figura 2.4 Proceso de failover de una configuración básica de cluster.

Durante el proceso de *failover* se presentan una serie de actividades:

2.4.1.1 TIEMPO DE DETECCIÓN

Un proceso de *failover* voluntario, no tardará más de unos cuantos segundos. El administrador de *failover* manda un mensaje a los miembros del *cluster* notificando dicho evento. El tiempo de detección en un *failover* involuntario comienza cuando los dos miembros del *cluster* (la explicación se hará con una configuración de dos nodos) pierden comunicación entre ellos o cuando hay demasiados errores generados por un disco. Durante el funcionamiento normal de un *cluster*, existe una conexión de red con el protocolo de "*Keep-alive*" (verificación

de conexión) que es establecido entre los administradores de *failover*. Ésta es una red privada con la que se verifica si hay comunicación entre los miembros del *cluster*. Al perderse la comunicación, el tiempo inicial de detección de *failover*, es el tiempo que tarda el servidor en línea en restablecer una conexión de red con el servidor que falló, antes de asumir que realmente está fuera de línea e iniciar un failover. Este intervalo de tiempo es de 30 segundos a 1:30 minutos aproximadamente, el cual podría ser reducido a menos de 5 segundos, sin embargo incrementa el factor de riesgo ya que puede provocar un *failover* falso o prematuro.

2.4.1.2 PERIODO DE ESTABILIZACIÓN

Este es sólo para el *failover* involuntario. Es un periodo de espera opcional que se define en el software de *cluster* para detectar y prevenir rápidamente un *failover*. El protocolo de red "*Keep-alive*" puede presentar una desconexión temporal. En un *cluster* de dos redes este periodo es requerido para ejecutar un failover dependiendo del estado de una red con respecto a la otra. En un *cluster* con una sola red este periodo es de ayuda para mejorar la estabilidad en el *cluster* y evitar falsas fallas en la red.

2.4.1.3 ARBITRAJE DEL *BUS* SCSI

El *cluster*, al determinar que no hay comunicación a través de la red, ejecuta un "*bus reset*" para eliminar las reservaciones de *failover* en el disco y espera cierto tiempo a que el otro administrador de *failover* muestre interés en el disco reafirmando la reservación sobre el *bus*. Esto es para asegurar que ninguno de los miembros del *cluster* tienen control sobre el disco al mismo tiempo. Este arbitraje del *bus* es requerido cuando se presenta una desconexión de red, para confirmar si realmente falló uno de los miembros o fue sólo una falla de red.

2.4.1.4 *FAILOVER* DE DISCO

Es requerido para *failover* voluntario e involuntario. El tiempo que tarda el *cluster* en hacer un *failover* de disco, es decir, pasar el control de un nodo a otro,

se determina a través del software y es un proceso demasiado rápido que es medido en milisegundos.

2.4.1.5 CONFIRMACIÓN DE LA INTEGRIDAD DE ARCHIVOS DEL SISTEMA

Una vez que se hace el cambio de control, se hace un chequeo para asegurar que los archivos del sistema están montados correctamente en el disco.

2.4.1.6 RECUPERACIÓN DE APLICACIONES

Esta acción es ejecutada después de que el disco es puesto nuevamente en línea. El tiempo total para recuperar las aplicaciones depende del número de grupos y del contenido de cada uno.

El servicio de *failover* define ciertas características en un cluster como:

- Habilidad de resistir una falla y mantener al cluster en operación.
- Debe incluir todos los componentes principales en la operación del cluster, como son CPU, interconexiones de red y dispositivos SCSI.
- Debe ser transparente al usuario y asegurar un acceso no interrumpible a los recursos.

El administrador del sistema generalmente tiene las siguientes opciones sobre el proceso de *failover*:

- ✓ Ejecutar todos los *clusters* de forma concurrente; esto maximiza la utilización del procesador.
- ✓ Mantener un servidor redundante en "*hot standby*". Esto significa, que dicho servidor estará operando y procesado datos pero no estará dando servicio al cliente de manera activa. Es decir, cuando se presente una falla el nodo "*hot-standby*" comenzará a dar servicios a clientes sin que baje el desempeño. Sin embargo, este modo de operación incrementa el costo general de las operaciones.

- ✓ Un buen *cluster* permite al administrador del sistema hacer *failover* de manera manual o automática. El *failover* manual es útil cuando se hacen actualizaciones en el *cluster*, por ejemplo.

Existe otro proceso muy relacionado con *failover* llamado "*failback*". Este ocurre cuando el nodo que ocasionó un *failover* es reemplazado o está nuevamente en estado operacional. Si el *failback* es activado, los servicios o grupos que hicieron un *failover* son regresados al nodo primario, es decir son transferidos al nodo original para ser controlados por dicho nodo. Si el proceso de *failback* no es activado, los servicios que hacen el failover permanecerán en el nodo secundario. Este proceso es determinado por el administrador del sistema y es importante para restablecer el *cluster* y balancear las cargas de trabajo.

2.4.2. Sistema tolerante a Fallas (Fault Tolerant FT)

Un sistema *fault tolerant* está diseñado para proveer "procesamiento continuo", donde los sistemas ejecutan un 99.99% de disponibilidad, esto es cerca de 5 minutos de baja promedio por año. En esta arquitectura se corren aplicaciones críticas, como centros de llamadas de emergencia, en telecomunicaciones, control de tráfico aéreo, etc., donde una baja del sistema provoca riesgos de vida, dinero y/o reputación.

En un sistema de *fault tolerance*, la falla de un componente no dará como resultando una caída ó baja del sistema (*downtime*) ni provocará la corrupción de datos en ningún sistema. Es un sistema que consta de una o cinco computadoras donde se corren aplicaciones idénticas de manera paralela. Por medio de mecanismos de comparación y selección, todos los datos son supervisados, y cualquier dato erróneo es reconocido y eliminado. En caso de presentarse una falla el control de operación del sistema sigue trabajando sin interrupción. La recuperación del sistema o de procesos se hace de manera automática y continua, incluso la computadora dañada puede ser reparada durante la operación normal del sistema.

Por ser un diseño extenso y utilizar componentes redundantes, su nivel de inversión es mayor.

2.4.3. LOCK MANAGER

El *lock manager* como su nombre lo dice es una serie de candados de administración, en un ambiente típico de *cluster*, su función es asegurar la integridad de los datos y aplicaciones ya que varios nodos del *cluster* pueden competir por un mismo servicio o recurso. Es decir, es un control de acceso que es necesario en sistemas que comparten recursos comunes.
Los candados de administración distribuidos (Distributed Lock Manager, DLM) sincronizan el acceso a los recursos que son compartidos dentro del cluster. Provee servicios para reforzar las políticas del recurso compartido basado en permisos y restricciones.

Los candados pueden ser utilizados para diversas necesidades, por ejemplo:

* Las bases de datos y archivos del sistema. Los pueden utilizar para controlar el acceso a copias distribuidas o limitar el acceso concurrente a dispositivos de disco.
* Son utilizados para controlar el inicio de aplicaciones y detectar fallas.
* Las aplicaciones pueden hacer uso de los candados para necesidades de sincronización.
* Juega un papel importante, asegurando alta disponibilidad y óptimo desempeño dentro del *cluster*, conforme las aplicaciones y bases de datos requieren de mayor paralelización y distribución.

En resumen, las características de los candados de administración son las siguientes:

* Actúa como controlador de tráfico para proveer sincronización de servicios dentro del ambiente cluster.
* Provee una interfaz de programación de aplicaciones (API *Aplication Programming Interface*) para peticiones, entregas y alteraciones de candados.
* Un candado es un método generado por un proceso el cual básicamente bloquea el uso de un recurso por algún otro proceso del sistema.
* El *lock manager* debe ser tan sofisticado como sea posible sin incurrir demasiado en el sistema.

2.4.4. PARALELIZACIÓN

Una característica con la que cuenta el software de cluster es la habilidad de correr aplicaciones en forma paralela. Al igual que uniprocesadores, multiprocesadores, *clusters* y cliente/ servidor utilizan paralelización en alguna forma, también los programas de aplicación pueden hacer uso de la paralelización que cada sistema ofrece. El utilizar esquemas de paralelización da al sistema alta disponibilidad siendo ideal para aplicaciones de misión crítica.

Los siguientes puntos son característicos de la paralelización:

- En cuanto a hardware, el configurar múltiples procesadores para que trabajen en forma paralela es una forma de contar con alto desempeño.

- El principal beneficio está en el uso de paralelización que pueden hacer los programas de aplicación, que como se mencionó anteriormente, son ideales para misión crítica donde el acceso a datos es de gran importancia y no debe ser interrumpido. Por lo general, estas aplicaciones encajan dentro de una de las siguientes categorías:

  - Información particionada. Por ejemplo, el sistema de sucursales bancaria en donde cada sucursal normalmente accede a sus propias cuentas y ocasionalmente accede a una cuenta de otra sucursal.
  - Acceso aleatorio a grandes bases de datos. Por ejemplo, en sistemas donde los registros accedidos por un nodo no pueden ser accedidos por otro nodo en un mismo período de tiempo, como sería en el registro de un vehículo.
  - Acceso de tablas departamentales. Un ejemplo es un sistema donde un nodo es dedicado a inventarios, otro es dedicado a personal y otro a ventas. Este tipo de aplicación soporta cierto límite para compartir datos entre los nodos, además de que sólo uno es el administrador de la base de datos, no tres.
  - Soporte de decisión. Por ejemplo, una base de datos de transacciones financieras es accedida continuamente, además de que se agrega constantemente nueva información.

- Conforme un cluster se hace más sofisticado, las aplicaciones en paralelo se convierten en un mayor beneficio para los usuarios.

Las ventajas de utilizar aplicaciones paralelas son las siguientes:

- ✓ Alta disponibilidad: una base de datos opera continuamente; si un nodo del cluster falla, en los demás nodos continuará la actividad de la base de datos.
- ✓ Alto desempeño: La base de datos ya no existe en un solo servidor gracias a las aplicaciones en paralelo, ejecutando un desempeño de *near-line*.
- ✓ Consolidación de base de datos: los administradores de bases de datos pueden combinar varias bases de datos previamente accedidas en diferentes nodos.
- ✓ Escalabilidad: los usuarios pueden incrementar su hardware, sin hacer modificaciones en el ambiente existente de la base de datos.

2.5 COMPONENTES DE HARDWARE PARA CLUSTER

2.5.1 SCSI. (*SMALL COMPUTER SYSTEM INTERFACE*)

2.5.1.1. INTRODUCCIÓN

Es importante conocer la tecnología SCSI, ya que parte de la estructura principal (o *backbone*) de un *Cluster* es el *bus* SCSI junto con los dispositivos y adaptadores unidos a éste.

El SCSI fue diseñado por Adaptec para enlazar subsistemas y periféricos inteligentes. En un principio se desarrolló para los sistemas de mini-computadoras y más adelante se convirtió en estándar oficial de ANSI en 1982.[4]

Desde la aparición del SCSI se habla de éste como un estándar aunque en realidad no está suficientemente definido para cubrir todas las necesidades, ya que los dispositivos SCSI diseñados por los principales fabricantes como son: CD-Technology, Chinon, COMPAQ, Digital Equipment Corporation, Hitachi, NEC, Panasonic, Sony, Texel, Toshiba y Adaptec, crean este estándar de acuerdo a sus requerimientos, por lo que los productos no son totalmente compatibles. Como resultado cada dispositivo SCSI tiene su propio adaptador y el software para este dispositivo no podrá ser usado con un adaptador hecho por otro fabricante. Al no

[4] Esta fecha fue tomada del manual de Microsoft Wolfpack versión Beta.

haber un estándar en el adaptador, el software y los dispositivos SCSI, se convierten en un problema para el usuario final.

La interfaz SCSI es totalmente diferente y más popular que IDE. Esta popularidad continúa creciendo debido, principalmente, a la velocidad con que trabaja, además de cantidad de tipos de hardware que pueden ser utilizados con éste bus. SCSI es una interfaz expandible, a diferencia de IDE que esta limitado a controlar discos duros y CD-ROMs.

Hoy en día, se escucha continuamente el término de "*Plug & Play*" (conectar y usar) como una herramienta que garantiza que los puertos y dispositivos tengan acceso a todas las características de un sistema, además de establecer un orden en el problema de asignación de recursos en las máquinas, sin embargo siguen existiendo dificultades en el sistema. Conectar un dispositivo SCSI en una máquina puede ser un proceso simple, pero al ir aumentando varios elementos en la misma conexión, aumentará el consumo de recursos. Por tal motivo los productores del equipo SCSI (Adaptec, DEC, Future Domain, Maxtor y NCR Corporation, etc.) se unieron junto con Microsoft para extender el uso de elementos SCSI en el sistema "*Plug & Play*".

Muchos sistemas ("*high-end*"[5]) han sido fabricados para soportar dispositivos SCSI. Los fabricantes siguieron los pasos de IBM el cual contenía el soporte nativo de SCSI. Esto dio como resultado el progreso en la tecnología SCSI para hacer de ésta una interfaz más fácil de usar.

2.5.1.2. INTERFAZ SCSI

El SCSI es un *bus* de I/O (entrada / salida) aceptado como estándar, que permite unir una serie de dispositivos en un sistema. Esta interfaz de alto nivel es válida para conectar periféricos como son unidades de disco, CD-ROM, escáners, unidad de cinta, etc. El adaptador SCSI también se conoce como "host adapter" y es el encargado de controlar de manera directa a los dispositivos, proporcionando así una menor carga de trabajo al CPU, pues aísla a éste de las funciones internas de los periféricos conectados. Es por ello que se puede decir que es una interfaz "inteligente". Por ejemplo, SCSI permite que las unidades de discos duros monitoreen sus propias pistas de forma independiente a la computadora, de tal

[5] Se entiende por sistema High End un servidor de alto performance.

forma que se ajustan para detectar en forma automática las pistas defectuosas o que tengan problemas, con el fin asignar los datos que estén ahí a cualquier otro sector que no tenga falla, e informa a la computadora como si se tratara de un disco sin problemas.

2.5.1.3. DESEMPEÑO DEL BUS SCSI

Es importante comprender algunos puntos que pueden afectar la funcionalidad de un bus así como la forma en que operan los dispositivos conectados. Básicamente, el desempeño de un bus está influenciado por los siguientes factores:

- Método de Transmisión.
- Ruta de Datos
- La velocidad del *Bus*.

MÉTODO DE TRANSMISIÓN

Existen dos métodos de transmisión que pueden ser usados en un *bus*:

- **Single-ended:** En un bus SCSI *single-ended*, un punto de dato y un punto de tierra son usados para entablar una conexión, es decir, los dispositivos sobre el bus se comunican a través de una línea (*single line*). Este método de transmisión hace que el costo del SCSI *single-ended* sea económico y su velocidad sobre distancias cortas sea mayor, pero es más susceptible al ruido que el método de transmisión diferencial, requiere de cables cortos y usualmente tiene una ruta de 8-bits de datos. La distancia total del cable para *single-ended* está limitada a 6 metros (incluso menos para el Fast SCSI)

- **Diferencial.-** Este método de transmisión no requiere de ningún punto de tierra, los dispositivos se comunican a través de un par de cables. El SCSI diferencial (*differential SCSI*) es menos susceptible al ruido que en el método anterior, además permite la comunicación a través de grandes distancias. Debido a sus características, tiene componentes más costosos, cables de gran longitud, el conector que se utiliza es de 68 pines de alta densidad, y generalmente su ruta

de datos es de 16-bit. La distancia del cable para diferencial soporta hasta 25 metros para conexiones.

Estos métodos de transmisión son independientes uno del otro y no pueden ser usados ambos en un mismo bus físico. Un dispositivo *single-ended* sólo puede ser conectado a un bus *single-ended* y un dispositivo diferencial solo puede ser conectado a otro dispositivo diferencial. Si se desea conectar dispositivos que usan diferentes métodos de transmisión, se requiere de un convertidor de señal SCSI entre los dispositivos. La distancia mínima entre dispositivos es de 10 cm, para evitar que la señal se degrade, tanto para el *bus single-ended* como para el *bus diferencial*.

La longitud total de un bus físico es calculada de donde se encuentra un terminador hasta donde se encuentra el otro. En la medición debe de incluirse la cantidad de cable que se encuentra dentro de cada sistema y unidad de almacenamiento de disco. La longitud interna varía, dependiendo del dispositivo.

La siguiente tabla describe la máxima longitud que un bus físico puede tener:

SCSI Bus	Velocidad del Bus	Longitud Máxima del Cable
Single-ended	5 MB/segundo (modo lento)	6 metros
Single-ended	10 MB/segundo (modo rápido)	3 metros
Diferencial	20 MB/segundo (modo lento o rápido)	25 metros

RUTA DE DATOS

Existen tres posibles rutas de datos para los dispositivos SCSI:

- **Narrow** - Implica una ruta de datos de 8-bit. La eficiencia de este método es limitada.
- **Wide** - Implica una ruta de datos de 16-bit. Este modo incrementa la cantidad de datos que son transferidos en paralelo sobre el bus.

- Ultra Wide- Los dispositivos *ultra wide* al igual que los dispositivos *wide* implica una ruta de datos de 16-bit, pero la velocidad con la que transmite los datos es cuatro veces más rápida que en los dispositivos *wide*.

Las rutas de datos *narrow* y *wide* no pueden ser usadas sobre un mismo bus físico. Un dispositivo *wide* debe de ser conectado a otro dispositivo *wide*, y un dispositivo *narrow* debe de ser conectado a otro dispositivo *narrow*.

VELOCIDAD DEL BUS

Los controladores SCSI operan en dos diferentes tipos de velocidad de transmisión de datos: Modo estándar o *slow* (lento) y modo *fast* (rápido). En modo *slow*, el bus SCSI puede manejar hasta 5 millones de bytes por segundo. Para fijar la velocidad del bus sobre un controlador SCSI, se utilizan ya sea comandos de consola o la utilería de configuración de la controladora.

2.5.1.4. CARACTERÍSTICAS DEL DISPOSITIVO SCSI

INICIADORES Y OBJETIVOS (TARGETS)

Existen ciertas características de los dispositivos SCSI que se mencionan a continuación. Una de ellas es que pueden ser clasificados como iniciadores ó *targets* (objetivos). El dispositivo iniciador se conoce también como "*host*" y es quien establece la comunicación de un dispositivo a otro. Por lo general los adaptadores suelen ser los dispositivos iniciadores. Los dispositivos *target* o blancos, reciben la información del dispositivo iniciador y responden a la petición para terminar el proceso de I/O. Cualquier SCSI puede hacer la función de iniciador o *target* dependiendo si reciben o envían señales, un ejemplo de una interacción entre iniciadores y *targets* es cuando un adaptador SCSI (iniciador) hace una petición de datos a un disco duro SCSI (*target*) el cual responde enviándole datos.

ARBITRAJE EN SCSI

SCSI cuenta con un tipo de arbitraje entre todos los dispositivos conectados al bus. Los dispositivos SCSI pueden iniciar un arbitraje por sí mismos de forma independiente del CPU, por ejemplo, iniciar la transferencia de información entre dos dispositivos sin requerir de la intervención ni desempeño del CPU, como lo hace un disco duro SCSI cuando se respalda a sí mismo en una unidad de cinta. Cuando un dispositivo no necesita en cierto momento tener acceso al bus, se libera de éste y se hace una nueva selección, mientras el dispositivo lleva a cabo otra operación y retoma más adelante el control.

SCSI ID

Para que la información y comandos puedan ser ruteados de manera correcta a través del Bus SCSI todos los dispositivos que formen parte de éste deben tener una identificación única, es decir, un SCSI ID. La selección de ID para cada dispositivo SCSI puede ser elegido indistintamente por lo que hay que cuidar que no se utilice el mismo para más de un dispositivo. Al elegir el ID también se debe tomar en cuenta que existe un Arbitraje SCSI, en otras palabras, existen prioridades en los ID SCSI. Cuando múltiples dispositivos tratan de acceder al bus simultáneamente el SCSI que tenga el ID más alto tomará el control del bus y los demás dispositivos tendrán que esperar a que el bus se encuentre libre para poderlo acceder. El SCSI-1 y SCSI-2 tienen 8 ID´s para acceder el bus, con un rango de 0 a 7 en donde el 7 es canal con mayor prioridad y el 0 el de menor prioridad. AGREGAR SCSI III, ETC.

Nota: Para los dos nodos del *cluster* de Windows NT, el ID que llevan las controladoras utilizadas para compartir el bus deben ser diferentes. Por default se asigna el ID 7 a una de ellas y la otra controladora puede llevar el ID 6, antes de ser conectadas al mismo bus.

2.5.1.5. TERMINADORES DEL BUS SCSI

Es indispensable que el SCSI bus esté terminado en sus dos extremos de manera correcta, para que los comandos y la información puedan ser trasmitidos hacia y desde todos los dispositivos conectados al bus. Si en medio del bus se encuentra un dispositivo terminado, entonces la información, así como los comandos, no podrán ser enviados de un extremo a otro del bus SCSI.

Los terminadores en cada extremo del bus SCSI previenen la interrupción de la señal para que ésta llegue de lado a lado. Las señales que viajan sobre un bus SCSI que no está terminado encuentran alta impedancia al final del bus. Lo cual actúa como una barrera para el flujo de la señal, provocando que la señal se refleje sobre el bus en sentido opuesto y se corrompa la información. Para prevenir la corrupción de la señal, los extremos del bus SCSI deben estar terminados. El terminar un bus significa agregar un circuito apropiado llamado terminador, el cual provee una impedancia que se incorpora al cable del bus y que impide la reflexión en sentido opuesto sobre el cable.

MÉTODOS DE TERMINADORES

Para terminar un bus SCSI existen tres métodos: Pasivo, activo y *forced perfect termination* (FPT).

Terminación pasiva.

El método más simple y antiguo para terminar un bus SCSI es el pasivo el cual mantiene una impedancia constante similar a la impedancia del cable del bus. La terminación pasiva se activa al encenderse la fuente de poder por medio del adaptador manteniendo un cierto nivel de impedancia. Esta impedancia fluctúa con los cambios de voltaje de la fuente, por lo que la terminación pasiva no es recomendable.

Terminación activa.

En este método el voltaje que recibe el terminador activo del adaptador es controlado por un regulador de voltaje. Debido a que la impedancia en el terminador es mantenida en forma constante, no hay fluctuaciones en el voltaje suministrado, por lo que este método es más recomendable.

FPT.

Es el método más complejo, además de mantener constante la impedancia en el terminador, altera la impedancia para compensar fluctuaciones de impedancia a través del cable, dispositivos y terminadores. FPT se usa normalmente en sistemas SCSI de alta velocidad, donde diferentes dispositivos y cables pueden causar discordancias en la impedancia.

TERMINADORES EN CLUSTER

Como ya se dijo, los terminadores son dispositivos formados por resistores eléctricos especiales, que se encuentran al final del Bus SCSI y no en otro lugar. Existen terminadores que son removidos o insertados manualmente, otros pueden ser habilitados o deshabilitados vía *switches* o comandos de software.

Los extremos del bus SCSI que forman el *Cluster* de Windows NT deben de estar terminados, aún si la máquina está encendida o apagada. El adaptador de SCSI también tiene terminadores, en éste caso es un jumper, el cual debe estar instalado en el adaptador para asegurar que esté terminado.

Existe otro dispositivo llamado cable "Y" (o "*rat tail*") que además de servir como terminadores SCSI, se usan para conectar cables SCSI. El cable "Y" está formado por dos cables unidos en el centro de un conector de bus SCSI. Uno de los extremos del cable Y se conecta a un terminador, mientras que el otro conecta al disco SCSI. Una ventaja de usar este tipo de cables es que al ir conectados al adaptador, si éste último falla, seguirá existiendo un terminador en el Bus, y por lo tanto el adaptador podrá ser removido de la máquina para reparación sin que el bus sea interrumpido.

En todos los dispositivos SCSI que se encuentren en medio o dentro de la cadena, es decir, que no estén en los extremos, se debe verificar que no tengan terminadores. Si se rompe la cadena en alguna parte central, se afectará a todo el Bus. Existen diferentes medios para cambiar un terminador:

- ✓ Remover físicamente los resistores en donde se encuentren o insertarlos (para dispositivos SCSI internos).

- ✓ Cambiar el estado de un *switch* en el bloque de *switch* del dispositivo.
- ✓ Remover o insertar un terminador para dispositivos SCSI externos.

2.5.2 RAID (*REDUNDANT ARRAY OF INDEPENDENT[6] DISKS*)

A la habilidad de una computadora o sistema operativo de responder a una catástrofe, como sería una falla en la fuente de poder o de hardware, sin que exista pérdida de datos y sin que el trabajo en proceso se corrompa, se le llama "*fault tolerance*". RAID es un sistema que provee *fault tolerance* de discos, protegiendo la información para que sea recuperada y restaurada.

RAID es una forma de coordinar un conjunto de múltiples discos, con la ayuda de una controladora especial que se encarga de administrar la distribución de datos a través de los discos. La información de un archivo es dividida en segmentos (grupo de bloques), los cuales pueden ser guardados a través de múltiples discos.

Mediante el uso de más de un disco, un arreglo de RAID provee los siguientes beneficios:

- × Mejora la disponibilidad de datos mediante configuraciones redundantes (niveles de RAID 1, 0+1, y 5).
- × Mejora el desempeño de I/O, hay mayor velocidad en la transferencia de datos comparada con la de un solo disco.
- × Incrementa la escalabilidad.

Dicha tecnología puede combinarse con otras tecnologías para sistemas de alta disponibilidad como son:

- × Sistemas de control de fuentes de poder ininterrumpibles.
- × Ventiladores y fuentes de poder redundantes.
- × Controladoras inteligentes que puedan respaldarse una a la otra.
- × Ambientes que puedan detectar y responder a sistemas de almacenamiento en acciones de recuperación.

[6] En algunas traducciones se encuentra como Independent y en otras como Inexpensive.

Un arreglo de discos independientes es controlado por un software que coordina todas las actividades. Al medio que contiene los discos, una o más controladoras inteligentes y en ocasiones otros dispositivos de almacenamiento, se le llama subsistema de almacenamiento (*Storage subsystem*).

Si un arreglo es administrado por una controladora inteligente y los discos residen en un subsistema de almacenamiento, es llamado *subsystem-base* RAID. Si el software encargado de administrar el arreglo se ejecuta en un servidor dedicado, lo más probable es que los discos que forman el RAID estarán en diferentes subsistemas de almacenamiento, a esta configuración se le llama sistema de *host-base* RAID.

RAID es una tecnología de *fault tolerance* gracias a la redundancia que se lleva a cabo para verificar datos en el arreglo de discos. El verificar datos es utilizado para regenerar bloques individuales de datos, conforme son solicitados por aplicaciones, de algún disco que se dañó. También se utiliza para reconstruir el contenido total de un disco que falló como protección de los datos después de la falla.

2.5.2.1 IMPLEMENTACIÓN DE RAID

Existen dos tipos de implementación para soluciones de RAID que son mediante hardware o software. Al decidir el tipo de implementación es recomendable considerar los siguientes puntos:

- El *fault tolerance* de software es más barato que el de hardware.
- El rendimiento del sistema es mejor con *fault tolerance* de hardware.
- Una solución de *fault tolerance* de hardware puede limitar las opciones del equipo a un solo vendedor.
- En una implementación *fault tolerance* de hardware varios proveedores permiten remplazar algún disco dañado sin la necesidad de dar de baja el sistema.

SOLUCIÓN DE RAID HARDWARE.

En un RAID basado en hardware los algoritmos corren en la tarjeta controladora conectada al bus de I/O, la cual se encarga de la creación y regeneración de la información. Debido a que las especificaciones de *fault tolerance* son llevadas a cabo directamente en el hardware, fuera del sistema operativo, se incrementa el desempeño sobre un amplio rango de aplicaciones instaladas.

SOLUCIÓN DE RAID SOFTWARE.

A diferencia de RAID hardware una implementación de software utiliza recursos del sistema operativo para desarrollar operaciones lógicas ya que los algoritmos de RAID corren en el CPU. Esta solución se acerca a su límite conforme se incrementa el software instalado en el sistema (esto sucede especialmente después de una falla de un disco, cuando la información debe ser reconstruida).

Con una solución de RAID software, la reconstrucción tomará algunas horas más de lo que tardaría una basada en hardware. El uso de RAID basado en software es menos costoso y es recomendable si la carga de software en el sistema es bajo, con pocos periodos picos de utilización.

2.5.2.2 NIVELES DE RAID

La tecnología RAID es catalogada en niveles[7], cada uno de los cuales ofrece distintas funciones para la configuración de discos y dependiendo del desempeño, confiabilidad y costo deseado. Los niveles más importantes son los siguientes:

- RAID 0
- RAID 1
- RAID 0+1
- RAID 5
- JBOD (Just a Bunch of Disks)

[7] Windows NT Server permite la implementación de RAID, soportando los niveles 1 y 5.

RAID 0

En éste arreglo, la información es guardada por segmentos, un segmento a la vez a través de los discos en arreglo, como se muestra en la *figura 2.5*, a lo que se conoce como "*striping*". En RAID 0 no hay redundancia por lo que, si uno de los discos falla, toda la información contenida en ese disco se pierde. A pesar de que no ofrece *fault tolerance* se considera una configuración válida de RAID, siendo su única ventaja la velocidad ya que ofrece una elevada trasferencia de I/O de cada uno de los discos.

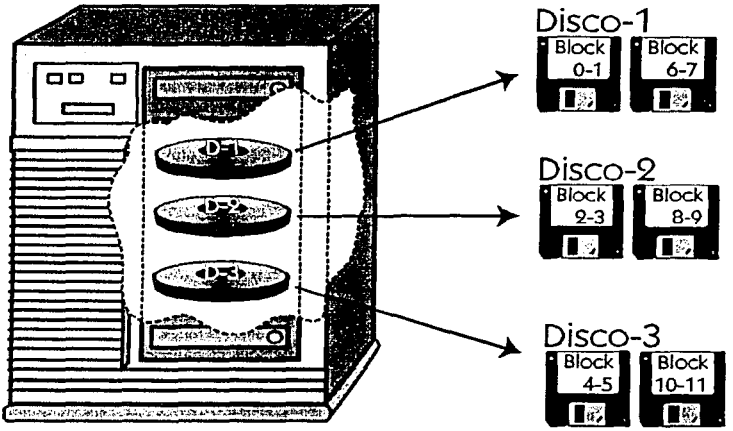


Figura 2.5- Los segmentos están definidos como dos bloques de 512 bytes cada uno. Por lo que los bloques 0 y 1 son almacenados en el disco 1, los bloques 2 y 3 en el disco 2, y así sucesivamente.

RAID 1

En RAID 1, mejor conocido como " *disk mirroring*" o "espejo de discos", se almacenan segmentos de datos idénticos en dos discos simultáneamente, de modo

que uno es espejo del otro, como se ve en la *figura 2.6*. Este es el medio más simple de almacenar información redundante, ya que para cada operación de disco el sistema escribe la misma información en ambos discos. A estas particiones también se les llama "partición original" y "partición de sombra". En este tipo de arreglo, si un disco falla el controlador puede recuperar la información del otro disco en ese conjunto por lo se recomienda a usuarios donde la confiabilidad es lo más importante. Sin embargo el costo de almacenamiento de la información es mayor ya que se duplica el requerimiento de espacio en disco, utilizando la mitad del disco como almacenamiento y la otra mitad como espejo.

Cabe mencionar que a cualquier partición, incluyendo la del sistema, se le puede crear un espejo. También incrementa el desempeño en cuanto al acceso a la información pero por el contrario la velocidad de escritura decrece.

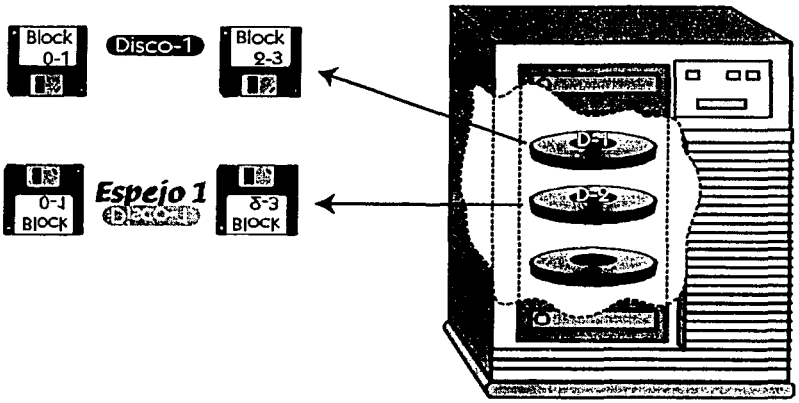


Figura 2.6 - Cada segmento es definido como dos bloques de 512 bytes cada uno. La información es almacenada tanto en el disco original como en el espejo.

A este nivel se puede implementar lo que se llama "*disk duplexing*" en donde se agrega una controladora más, por lo que cada disco contará con su propia

controladora, si una de ellas falla la otra continuará con el proceso. Con este arreglo se protege tanto una falla de disco como una falla en la controladora.

RAID 0+1

Es una combinación de "*striping*" y "*mirroring*", distribución y espejo, como se ve en la *figura 2.7*. La información es guardada a través de discos de la misma manera que lo hace RAID 0 y además hace una redundancia de espejo similar a RAID 1. En RAID 0+1 el espejo de la información se hace rotándola de disco en disco (al disco que se encuentra enseguida), por lo que se puede elegir un número impar de discos en dicha configuración mientras que en RAID 1 se requieren de una cantidad par de discos. La información puede seguir siendo accedida cuando se presenta una falla en uno de los discos. Sin embargo, cuando hay fallas en múltiples discos no es posible recuperar la información.

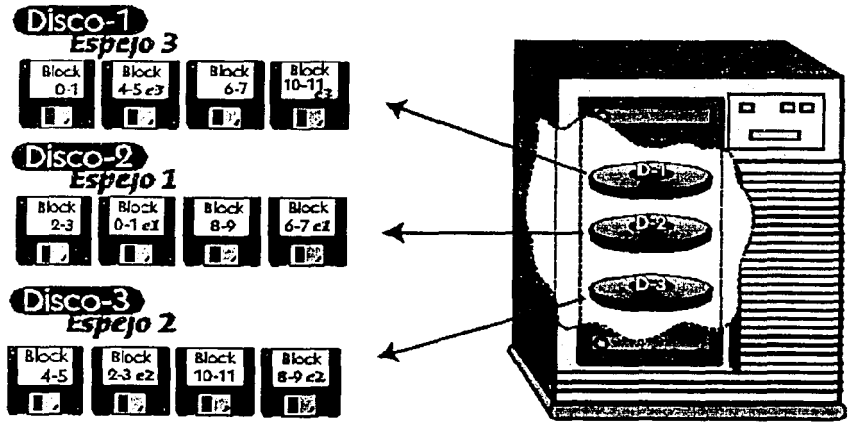


Figura 2.7- RAID 0+1. Cada segmento es definido como dos bloques de 512 bytes cada uno. Como se muestra, el *espejo 1* (en el disco 2) hace referencia a los datos en el disco 1, *espejo 2* (en el disco 3) hace referencia al disco 2, el espejo 3 (en el disco 1) hace referencia al disco 3, y así sucesivamente.

RAID 5

Esta configuración combina "*striping*" (descomposición de la información en discos) y redundancia. La redundancia se genera con información de paridad, esto es, que tanto la información de datos como la de paridad es escrita en diferentes discos. La paridad es un método matemático para verificar la integridad de los datos. Si llegase a fallar uno de los discos del arreglo los datos pueden ser recuperados a partir de los bloques de datos y de paridad de los discos en funcionamiento, de manera que la información permanece accesible. Sin embargo, al igual que el RAID 0+1, no se puede recuperar información cuando se presentan fallas en múltiples discos.

Como se muestra en la *figura 2.8*, este tipo de configuración dedica el equivalente al espacio de un disco para distribuir la información de paridad a través de todos los discos en el grupo. Por tal motivo tiene una ventaja sobre RAID 1 en cuanto a costo ya que el uso del disco es optimizado. Por ejemplo: si se tienen 4 discos en RAID 5 el espacio que se ocupara será del 25% comparado con un 50% que se utiliza en el espejo de discos. RAID 5 tiene un mejor desempeño en cuanto a la lectura de la información por la distribución en los múltiples discos, pero las operaciones de escritura son más lentas debido a los cálculos de paridad.

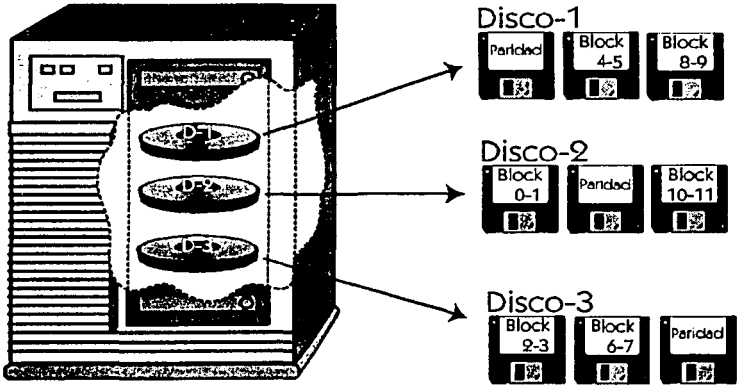


Figura 2.8 - La información es distribuida en los discos que componen el arreglo, Además cada disco cuenta con un archivo de paridad.

Al implementar RAID 5 se deben configurar al menos tres discos en el arreglo. No es necesario que estos discos sean idénticos, sin embargo se deben configurar bloques de un mismo tamaño de espacio no particionado en cada uno de los discos. Al haber paridad, en una configuración ya creada, no se pueden agregar más discos al arreglo para incrementar el volumen.

JBOD (JUST A BUNCH OF DRIVES)

Este tipo de configuración permite instalar y acceder un disco en subsistemas de Storage Works RAID 200 como un disco convencional. Un ejemplo está en la *figura 2.9*. Al igual que cualquier configuración no-RAID, JBOD no permite redundancia de datos.

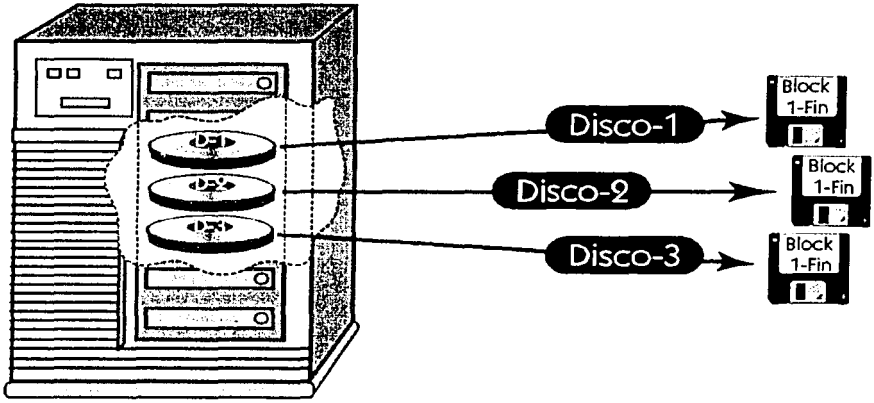


Figura 2.9 - Cada disco almacena información de forma independiente.

CAPÍTULO III.- INSTALACIÓN Y CONFIGURACIÓN DE UN CLUSTER

3.1 CONFIGURACIÓN DE UN CLUSTER

En el capítulo anterior se mencionaron conceptos generales de lo que es un cluster, con el fin de dar al lector una visión de lo que abarca dicha tecnología, su funcionamiento, así como sus componentes necesarios que lo forman. En este capítulo se verá cómo se lleva a cabo la instalación tanto de hardware como de software de un cluster, así como una definición más detallada de dichos componentes y algunos modelos de tarjetas, servidores, discos, etc., que han sido probados y utilizados en configuraciones de cluster.

3.1.1 CONFIGURACIÓN DE HARDWARE

Antes de instalar el software de cluster y configurar los nodos y servicios, primero se debe de configurar el hardware. Esta configuración consiste en un número de componentes de hardware con ciertas especificaciones. Si se falla o se omite cualquiera de estos requerimientos, la operación del cluster puede llegar a no ser del todo correcta.

Los componentes de hardware de un cluster son:

- Nodos del cluster.- Los nodos son los recursos básicos de cómputo dentro de un cluster, los cuales pueden ser desde dos hasta dieciséis nodos. Los nodos del cluster corren aplicaciones y proveen acceso a los datos, y deben de estar conectados al menos a un bus *SCSI* compartido y a una red común. Los nodos se comunican el uno con el otro y a su vez monitorean los dispositivos compartidos y la red por medio del bus. Si una falla de hardware o software ocasionan que un nodo no pueda seguir ejecutando aplicaciones o proveyendo datos, el mecanismo de *failover* del cluster relocaliza los recursos a un nodo disponible, siendo así de alta disponibilidad.
- Controladores *SCSI*.- Cada nodo debe de tener cuando menos una controladora *SCSI* instalada dentro de un *slot* de *I/O* para poder conectar el sistema al bus compartido.

- Configuración de la unidad de almacenamiento (Storage).- Los discos usados en el cluster deben de estar localizados dentro de una unidad de expansión de almacenamiento externo conectado al bus SCSI compartido. Con esto todos los nodos del cluster podrán acceder a los datos que se encuentran en los discos. Debido a que estas unidades externas tienen sus propias fuentes de alimentación, no dependen de la alimentación de otro sistema.
- Bus *SCSI* Compartido.- Los nodos del cluster deben de estar conectados al menos a un bus *SCSI* compartido, y a una unidad de almacenamiento conectada al mismo bus. Es necesario que los buses *SCSI* compartidos tengan el mismo número lógico en cada nodo, que estén terminados, y que se encuentren dentro de los límites de longitud de cableado. *SCSI* cuenta con ocho números lógicos (ID) disponibles sobre cada bus. Los dispositivos sobre el bus compartido deben estar conectados de tal manera que cualquiera de estos dispositivos se pueda desconectar sin afectar la operación del bus.
- Interconexión de Red.- Los nodos deben de estar conectados al menos a una misma red, es decir al mismo segmento de red. Los cluster nos permiten configurar redes redundantes.

La configuración mínima de hardware de un cluster es la siguiente:

- ✓ Dos nodos.
- ✓ Una unidad de almacenamiento.
- ✓ Un bus *SCSI* compartido.
- ✓ Una red común.

Para incrementar la disponibilidad o desempeño del cluster hay que adicionar más nodos, buses *SCSI*, o conexiones de red. Si se requiere de más discos se puede agregar un subsistema *RAID* o más buses *SCSI*. Por ejemplo, se pueden usar dos buses *SCSI* compartidos y hacer un espejo de los discos a través de los buses con el fin de obtener más confiabilidad en los datos.

La *figura 3.1* muestra un ejemplo de una configuración de hardware de un cluster que incluye dos nodos, con un bus *SCSI* compartido y una unidad de almacenamiento. Como se puede observar la terminación de este bus se encuentra en los extremos de los cables, los cuales nos permiten desconectar los sistemas sin afectar la terminación del bus. La configuración de red no se muestra.

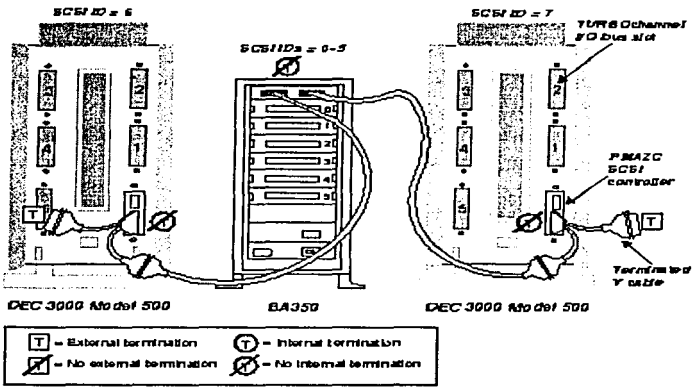


Figura 3.1 Configuración de hardware de un servidor de alta disponibilidad.

3.1.2 SOFTWARE DEL CLUSTER

Una vez verificada la configuración de hardware, se procede a instalar el software del cluster en todos los nodos que van a formar parte de éste. Después de haber instalado el software, se debe ejecutar la utilería que permite verificar que el software se encuentra correctamente instalado y el hardware está propiamente configurado. Posteriormente se corre la utilería de administración del cluster la cual nos permite agregar más nodos a un cluster (sólo si este puede aceptar más nodos) y además nos permite crear los servicios que harán que nuestras aplicaciones y datos se encuentren altamente disponibles. A continuación se describe cómo funciona el software de un cluster.

3.1.2.1 Verificación De La Instalación De Un Cluster

El procedimiento de verificación de instalación de un cluster detecta errores de configuración, como ejemplo; Verifica que se cumplan las siguientes condiciones:

* Las subredes requeridas por un cluster se encuentren instaladas.
* Los parámetros de instalación sean los correctos.
* Los nodos y el cluster puedan ser vistos en la red utilizando la utilería *ping*.
* La licencia introducida sea válida.

El procedimiento de verificación no hace ningún tipo de modificación al sistema, de tal manera que este programa se puede ejecutar siempre que se desee diagnosticar errores. Esta utilería se debe de ejecutar siempre en el nodo que se desea validar.

Cuando se encuentra un error, la utilería de verificación sugiere algunas posibles acciones correctivas que pueden ayudar a solucionar el problema. En algunos casos; el error reportado puede ser síntoma o consecuencia de otro problema. Siempre se deben de leer todos los mensajes enviados por el programa antes de intentar corregir algún problema. Cuando la acción correctiva sugerida no soluciona el problema, el siguiente paso es analizar el "archivo de log" del sistema para obtener más información.

3.1.2.2 La Utilería de Administración

La utilería de administración provee una interfaz amigable que nos permite instalar y administrar el cluster y sus nodos desde cualquier sistema. Dentro de esta utilería se pueden ejecutar las siguientes tareas:

* Agregar o borrar nodos.
* Administrar la configuración de red del cluster.
* Verificar los eventos registrados del cluster en el archivo log.
* Instalar y administrar los servicios.
* Verificar el estado de los nodos y los servicios ofrecidos por el cluster.

Algunos cluster cuentan también con una utilería de administración, así como con una interfaz de comandos en línea que permite administrar el cluster desde un script. Con esto el cluster se puede coordinar con las operaciones de mantenimiento del centro de cómputo.

3.1.2.3 CLUSTER MONITOR

Es una interfaz gráfica que nos permite monitorear los nodos, la unidad de almacenamiento compartida y los recursos del cluster.

3.1.2.4 OPERACIÓN DE LOS NODOS DEL CLUSTER

Como se mencionó anteriormente los nodos de un cluster se encuentran conectados al menos a un bus *SCSI* compartido, a una red común y tienen instalado el software de cluster. Los nodos monitorean los discos, la red y a todo el sistema del cluster. En ellos se ejecutan los servicios de cluster y hacen que las aplicaciones y datos se encuentren siempre disponibles a los clientes. Un nodo puede ejecutar más de un servicio, pero solo puede ejecutar uno a la vez.

Si una falla de hardware o software ocasiona que un nodo no pueda tener disponible un servicio, el mecanismo de *failover* del cluster automáticamente relocaliza el servicio en otro nodo capaz de ejecutarlo. De esta manera el servicio que era ejecutado en el nodo que falló se mantiene disponible. Cuando el nodo original que falló vuelve a encontrarse en funcionamiento el servicio puede regresar al él (*failback*), o permanecer en el nodo donde se encuentra siendo ejecutado, esto depende de la configuración del servicio.

Además, la utilería de administración permite mover las aplicaciones de un nodo a otro para balancear la carga de trabajo y planear los servicios de mantenimiento y bajas del sistema.

Los nodos del cluster ejecutan los siguientes "drivers" y "demonios".

- Demonio Director.- Se ejecuta únicamente en un nodo y controla el cluster entero.
- Demonio Agente.- Se ejecuta en todos los nodos y controla las operaciones del cluster en ese nodo.
- Driver de administración de disponibilidad.- Se encarga de monitorear al cluster y reporta cualquier falla de disco a los demonios director y agente. Se ejecuta en cada nodo.
- Demonio de monitoreo de estado.-. También monitorea al cluster y reporta cualquier falla sobre los nodos o la red a los demonios director y agente. Se ejecuta en cada miembro.

- Demonio de Log.- Se encarga de recopilar todos lo mensajes que son generados por cualquier nodo.

3.1.2.5 SERVICIOS DE UN CLUSTER

Un ambiente cluster hace que una aplicación o disco de datos sea altamente disponible. Por ejemplo, se puede crear un servicio para exportar archivos de sistema o ejecutar una aplicación de base de datos. Crear un servicio de cluster involucra tareas similares a las necesarias para instalar una aplicación o disco de datos fuera de un cluster. Antes de crear un servicio de cluster, se debe instalar la aplicación en todos los nodos. Además, hay que hacer uso de las utilerías de administración de disco para preparar los servicios de configuración de la unidad de almacenamiento.

Cuando se crea un servicio en un cluster la utilería de administración nos pregunta, como mínimo, la siguiente información:

- Un nombre único
- La configuración ya sea de la aplicación o de la unidad de almacenamiento que se desea hacer altamente disponible.
- El nombre del nodo en el que se desea ejecutar el servicio y cómo se desea que este servicio se comporte cuando ocurra una falla.

Solo cierto tipo de aplicaciones pueden configurarse en un ambiente de cluster para que sean altamente disponibles dentro de un servicio de cluster. La aplicación debe de tener las siguientes características:

- La aplicación debe ejecutarse sólo en un nodo a la vez.
- La aplicación debe de ser capaz de empezar y terminar gracias a un conjunto de comandos que son ejecutados en un orden específico. Cuando se crea un servicio de cluster, estos comandos se incluyen en un conjunto de programas llamado "*action scripts*". El software de cluster usa estos *scripts* para realizar un *failover* de un servicio de cluster.

Un cluster provee soporte para tres tipos de servicio:

- Servicios de Sistema de Archivos de Red (Network File System; NFS).- Permite ofrecer alta disponibilidad de acceso a un disco de datos (por ejemplo, un disco montado o un sistema de correo). Cuando se crea un servicio de NFS, se debe

especificar un nombre de red único que usará el servicio. El nodo que ejecuta el servicio, responde a la dirección de IP que le es asignada al nombre del servicio de NFS y exporta los servicios de datos. Si el servicio es relocalizado en otro nodo, este responderá a la dirección de IP. Los clientes nunca notaran el cambio de nodo que exporta los datos, sino que solo experimentan una desconexión temporal del servicio de NFS.

- Servicio de Disco.- Mantiene alta disponibilidad de acceso a un disco o a una aplicación basada en disco, tal como a un programa de base de datos. Un servicio de disco es similar a un servicio de NFS excepto que ningún dato es exportado. Al crear un servicio de disco se debe especificar el sistema de archivo que se desea hacer altamente disponible.

- Servicios definidos por los usuarios.- Este tipo de servicio permite tener siempre acceso a una aplicación que no está basada en un disco (por ejemplo un servicio de login). En este tipo de servicio se deben usar los "*action scripts*" desarrollados para poder realizar un *failover* de la aplicación.

La *figura 3.2* muestra cómo los servicios de un cluster aparecen a los clientes.

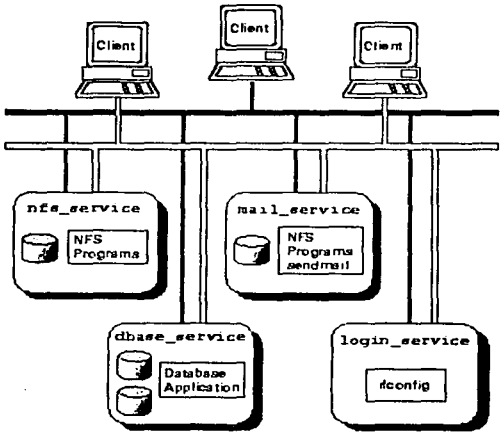


Figura 3.2 - los clientes usan los servicios de nombres *nfs_service, dbase_service, mail_service,* y *login_service* para acceder a los servicios del cluster a través de la red.

Un servicio de NFS puede usar solo datos de disco, o puede usar aplicaciones y datos de disco. Por ejemplo, el servicio *nfs_service* consiste sólo de un servicio de NFS, pero el servicio *mail_service* consiste de un servicio de NFS y del comando *sendmail*.

Un servicio de disco puede también utilizar sólo datos de disco, o utilizar aplicaciones y datos de disco. Por ejemplo, el servicio *dbase_service* consiste de ambos, es decir, tiene tanto datos de disco como una aplicaciones de base de datos. Un servicio definido por un usuario puede usar únicamente aplicaciones.

3.1.2.6 ACTION SCRIPTS

Un cluster usa *action scripts* para mantener la disponibilidad de un servicio de una aplicación o datos. Los *action scripts* siguen un procedimiento (por ejemplo, levantan un servicio en un nodo del cluster) realizado por una serie de pasos con el fin de ejecutar una tarea. El cluster siempre se asegura que cada paso del procedimiento sea exitosamente ejecutado. El orden en que van siendo ejecutados los pasos del procedimiento confirma que cualquier dependencia sea cumplida antes de continuar al siguiente.

Existen cinco tipos de *actions scripts*: Agregar (*add*), eliminar (*delete*), iniciar (*strart*), detener (*stop*), y verificar (*check*). Estos *scripts* son ejecutados únicamente en ciertas ocasiones y sólo en nodos específicos de la siguiente manera:

- Los *action scripts* "agregar" y "eliminar" son ejecutados en todos los nodos para configurar o remover los servicios en un cluster, respectivamente.
- Los *action scripts* "iniciar" y "detener" se ejecutan únicamente sobre un nodo con el fin de activar o suspender un servicio.
- El *action script* "verificar" es ejecutado sobre todos lo nodos con el fin de determinar si el servicio está siendo ejecutado.

Para cada uno de los cinco tipos de *action script* existen dos versiones: internos y definidos por el usuario. Los internos son provistos por el software del cluster y son utilizados únicamente en los servicios de disco y NFS para exportar o hacer un *failover* de un disco de datos, y utiliza los definidos por el usuario para hacer un *failover* de una aplicación de base de datos.

3.1.3 Operación de un Cluster

Con el fin de entender completamente la operación de un cluster, hay que conocer como un cluster reacciona a fallas, así como también su reacción a cambios que ocurren cuando el administrador del sistema usa la utilería de administración para modificar o relocalizar servicios. A continuación se verán dichos temas.

3.1.3.1 Detección y Respuesta a Fallas en un Cluster

Un cluster detecta y responde a los siguientes tipos de fallas:

- Una falla de un nodo.- Cuando un nodo deja de responder.
- Una falla crítica de un controlador *SCSI*.- Cuando un nodo detecta un error de I/O en un disco compartido que está disponible. Este error también puede ser causado por una falla al convertir una señal *SCSI*.
- Una falla de Dispositivo.- Un disco compartido no responde al I/O.
- Una falla de red.- Cuando dos nodos no pueden comunicarse a través de la red. Por ejemplo, una instancia de red falla o un cable de red es desconectado.
- Una falla en una interfaz de red que se está monitoreando.

Los encargados de detectar fallas en un cluster son demonios de monitoreo de estado, que detectan fallas en los nodos y en la red, y el driver de administración de disponibilidad, que detecta fallas de acceso a los dispositivos que forman el cluster.

Cada demonio de monitoreo de estado local verifica si existen fallas en los nodos o en la red, preguntando, en intervalos de aproximadamente tres segundos, si los demás nodos se encuentran activos. El demonio de monitoreo de estado usa el driver de administración para interrogar a los nodos a través de la red y del bus *SCSI* compartido. El demonio de monitoreo de estado notifica al demonio agente local y al demonio director si existe una falla de nodo o de red. El driver de administración de disponibilidad notifica al demonio agente local si se presenta una falla en un dispositivo que forma el cluster.

3.1.3.2 Falla de un Nodo

Si un nodo no responde a las interrogaciones de red y *SCSI*, el cluster realizará las siguientes acciones:

- El software del cluster manda una señal de alerta al administrador del cluster para notificar que un nodo ha fallado.
- Si el demonio director está siendo ejecutado en el nodo que falló, el cluster reinicia al demonio director en otro nodo que se encuentre disponible.
- El cluster reinicia los servicios ejecutados en el nodo que falló, de acuerdo a la política de relocalización de servicios.

NOTA: Cuando el nodo que falló vuelve a estar disponible, el demonio agente local ejecuta el *script* de detener (*stop action script*) para cada servicio en la base de datos del cluster.

3.1.3.3 Falla Crítica de un Controlador SCSI

Cuando un cluster no puede acceder a un dispositivo de almacenamiento compartido ocurre lo siguiente:

- El software de cluster manda una señal de alerta al administrador del cluster para notificar que un error de I/O ha ocurrido.
- Si el disco es parte de un arreglo de discos y puede continuar funcionando sin que éste afecte a todo el dispositivo, el cluster retiene el servicio ejecutado.
- Si el servicio no se puede ejecutar sin el dispositivo afectado, el cluster detiene el servicio y trata de reiniciarlo en otro nodo, de acuerdo a la política de relocalización de servicios. Si el servicio no puede ser reiniciado, éste se mantiene sin asignar. Si el servicio no puede ser suspendido, el cluster reinicializa al nodo.

3.1.3.4 Falla de un dispositivo de Disco

Cuando un cluster detecta una falla de un dispositivo de disco, ocurre lo siguiente:

- El software de cluster manda una señal de alerta al administrador del cluster para notificar que un error de I/O ha ocurrido.
- Si el disco es parte de un arreglo de disco y puede continuar funcionando sin que éste afecte a todo el dispositivo, el cluster retiene el servicio ejecutado.
- Si el servicio no se puede ejecutar sin el dispositivo afectado, el cluster detiene el servicio y lo deja sin asignar. Si el servicio no puede ser suspendido, el cluster reinicializa al nodo.

3.1.3.5 Falla en la Comunicación por Red

Si una línea de red sobre un nodo primario o de respaldo falla (por ejemplo, por la tarjeta de red o por un cable de red desconectado), y otra línea de red se encuentra disponible, ambos nodos se comunican entre si a través de esa línea y la operación del cluster no se ve afectada. Ver la sección 1.4.1.5 para obtener información acerca de cómo se comporta un cluster cuando falla la interfaz de red que se esta monitoreando.

Si el nodo no puede acceder a otros nodos sobre cualquiera de sus interfaces de red, pero si puede acceder a ellos a través del bus *SCSI* compartido, entonces se determina que ocurrió una partición total de la red; el cluster responde de la siguiente manera:

- El software de cluster manda una señal al administrador del cluster para notificar que una partición total de la red ha ocurrido.
- Los servicios del cluster continúan ejecutándose y un *failover* automático puede ocurrir. No se puede hacer uso de la utilería de administración para cambiar al cluster o para relocalizar a los servicios manualmente.

3.1.3.6 Falla en el Dispositivo de Red que se está Monitoreando

La respuesta de un cluster a una falla de un dispositivo de red sobre un nodo depende de si se está monitoreando este dispositivo de red. Algunos clusters permiten monitorear el estado en que se encuentra cualquiera de las diferentes tarjetas de red de un nodo, para así poder especificar qué acciones se deben realizar si se presenta una falla en alguna de éstas.

En la sección 1.4.1.4 se mencionó el comportamiento del cluster cuando una interfaz de red falla, en caso de que no se estén monitoreando. Si por el contrario, se está monitoreando una tarjeta de red y ésta falla, ocurre lo siguiente:

- El software de cluster manda una señal de error al administrador del cluster para notificar que una tarjeta de red ha fallado.
- En algunos clusters, la señal de error se envía a un *script* con el fin de realizar pruebas. Este *script* realiza las siguientes acciones si todas las interfaces sobre un nodo fallan:
  - El cluster detiene todos los servicios que se están ejecutando en el nodo, iniciándolos en otro, de acuerdo a las políticas de relocalización de servicios. Si el servicio no se puede detener, el software del cluster reinicializa al nodo.
  - Si el nodo está ejecutando al demonio director, el cluster inicia a este demonio en otro nodo.

3.1.3.7 Respondiendo a la Modificación y Relocalización de Servicios

Si estamos usando la utilería de administración para relocalizar un servicio de un nodo a otro, el cluster reacciona de la siguiente manera:

- La utilería de administración indica al demonio director que detenga el servicio sobre un nodo y lo reinicie en otro.
- El demonio director indica al demonio agente, del nodo en donde se está ejecutando el servicio, que lo detenga.
- El demonio director indica al demonio agente, del nodo que se eligió para ejecutar el servicio, que lo inicie.

Si se está usando la utilería de administración para modificar un servicio existente, el cluster reacciona como sigue:

- La utilería de administración indica al demonio director que suspenda el servicio, borre el servicio, agregue el servicio modificado, reinicie el nuevo servicio, y almacene la nueva configuración en la base de datos que contiene las modificaciones.
- El demonio director indica al demonio agente del nodo que está corriendo el servicio que detenga dicho servicio.
- El demonio director indica a todos los demonio agentes que borren el servicio.
- El demonio director indica a todos los demonio agentes que agreguen el servicio modificado.

- El demonio director selecciona un nodo sobre el cual ejecutar el servicio y le dice al demonio agente de ese nodo que inicie el servicio.
- El demonio director indica a todos los demonio agentes que almacenen la nueva configuración en su base de datos.

3.1.4 Pasos para Instalar un Cluster

Un cluster está formado tanto por hardware como por software. Las tareas que se deben realizar para instalar un cluster por primera vez son las siguientes:

1. Instalar el sistema operativo en cada nodo que vaya a formar parte del cluster, si éste no esta ya instalado.

2. Planear la configuración de hardware y software del cluster. La configuración de hardware varía dependiendo del número y tipo de nodos, de la configuración del almacenamiento compartido, y del número de buses *SCSI* que van a formar el cluster.

3. Preparar cada nodo de la siguiente manera:
   a) Configurar la red del nodo primario y, opcionalmente, configurar una o dos redes más como respaldo.
   b) Instalar un controlador *SCSI* para el bus compartido, removiendo los terminadores de la tarjeta si es necesario.
   c) Actualizar el *firmware* de las controladoras *SCSI* si es necesario.
   d) Asegurar que todos los ID´s *SCSI* sean únicos para cada bus compartido.
   e) Fijar la velocidad del bus.
   f) Conectar un cable Y, conector *trilink*, o un convertidor de señal *SCSI* a la controladora para así conectar el bus compartido.

4. Preparar la configuración de la unidad de almacenamiento:
   a) Instalar los discos en la unidad de almacenamiento, asegurando que cada ID *SCSI* sea único sobre el bus compartido.
   b) Poner un terminador en la unidad de almacenamiento, si es necesario.
   c) Agregar un conector trilink o convertidor de señal *SCSI* a la unidad de almacenamiento para la conexión compartida.

5. Conectar los nodos y la unidad de almacenamiento al bus o buses *SCSI* compartidos, tomando en cuenta lo siguiente:
   a) Conectar sólo dispositivos que tengan el mismo método de transmisión, o usar un convertidor de señal *SCSI* entre los dispositivos.
   b) Conectar dispositivos al bus *SCSI* compartido de tal manera que puedan ser desconectados sin que afecten la terminación del bus.
   c) Terminar físicamente cada bus en los extremos.
   d) Mantener las especificaciones de longitud de cada bus compartido.
   e) Los números lógicos (ID) que se le asignaron a los controladores para los buses compartidos, deben ser mayores que los números asignados a los buses locales.

6. Instalar el software de cluster en cada nodo, de la siguiente manera:
   a) Registrar la licencia del producto, si es necesario.
   b) Instalar el software de cluster y todos sus "parches".
   c) Reiniciar el sistema.

7. Instalar los nodos del cluster y crear los servicios, como sigue:
   a) Agregar los nodos, usando la utilería de administración.
   b) Preparar todos los discos para que sean usados en los servicios del cluster, crear o configurar arreglos de discos.
   c) Instalar cualquier aplicación que se desee hacer altamente disponible y crear todos los "*action scripts*" para los servicios.
   d) Crear los servicios del cluster, usando la utilería de administración.

3.2 HARDWARE SOPORTADO

En este capítulo se describe el hardware soportado por un cluster y cualquier requerimiento para cada componente de hardware. Las configuraciones pueden variar, pero solo el hardware descrito en este capítulo puede ser usado en un cluster.

3.2.1 NODOS

Un cluster puede consistir de dos o más nodos (hasta 128 en algunos casos). La siguiente tabla lista las controladoras *SCSI* que se deben de usar para cada uno de ellos. También es necesario verificar la revisión de *firmware* requerido para configurar el cluster.

Sistema	Controladora *SCSI*
Alpha Server 800	KZPSA
Alpha Server 1000	KZPSA
Alpha Server 1000A	KZPSA
Alpha Server 2000 y 2100	KZPSA
Alpha Server 4000 y 4100	KZPSA
Alpha Server 8200 y 8400	KZPSA
Digital Server 1200	Adaptec 2944UW
Digital Server 3100 y 3200	Adaptec 2944UW
Digital Server 3300	KZPSA
Digital Server 5100 y 5200	Adaptec 2944UW
Digital Server 5300	KZPSA
Digital Server 7100	Adaptec 2944UW
Digital Server 7300	KZPSA

3.2.2 ADAPTADORES DE RED Y OPCIONES

Un Cluster soporta las siguientes tarjetas de red:

- ✓ DE500 (PCI/Fast Ethernet)
- ✓ DEFPA (PCI/FDDI)
- ✓ DE435 (PCI/Ethernet)
- ✓ DEFEA (EISA/FDDI)
- ✓ DE422 (EISA/Lance Ethernet)
- ✓ DE425 (EISA/Ethernet)
- ✓ PMAD (TURBOchannel/Ethernet)
- ✓ DEFTA (TURBOchannel/FDDI)
- ✓ DEFZA (TURBOchannel/FDDI)
- ✓ DEMNA (XMI/Ethernet)
- ✓ DEMFA (XMI/FDDI)

3.2.3 CONTROLADORAS SCSI

Cada nodo utilizado en un cluster debe de tener una controladora *SCSI* soportada para que se pueda unir al bus *SCSI* compartido. La siguiente tabla lista las controladoras *SCSI* que son soportadas y el método de transmisión.

Controladora	Método de Transmisión	Revisión de *Firmware*
KZPSA	Diferencial	A10
Adaptec 2944UW	Diferencial	1.35

Nota.- Se entiende por revisión de firmware como la versión mínima de software que debe de tener la memoria ROM (Read Only Memory) de las controladoras.

Los adaptadores KZPSA deben de contar con los siguientes requerimientos:

- ✓ Todo adaptador KZPSA debe de tener como mínimo la revisión A10 como *firmware*; en caso de no tener esta revisión se deberá de utilizar la herramienta de actualización para la revisión requerida.
- ✓ Se deberá de deshabilitar la opción de *bus reset* en todas las tarjetas que formen parte del cluster.
- ✓ Se les deberá de asignar un ID único por bus *SCSI*.
- ✓ No tendrán que tener el BIOS habilitado.

Los adaptadores ADAPTEC 2944UW deben contar con los siguientes requerimientos:

- ✓ Deberán tener como mínimo la versión 1.35 de BIOS.
- ✓ Deberán de tener el BIOS deshabilitado.
- ✓ No deberán tener la opción de *Reset Bus SCSI* habilitada.
- ✓ Deberán tener un número de ID único por bus *SCSI*.

3.2.4 UNIDADES DE ALMACENAMIENTO

En la siguiente tabla se lista las unidades de almacenamiento soportadas.

Unidad de Almacenamiento	Modo de transmisión	Data Path	Raid
BA356	Single-Ended	Ultra Wide, Wide, Narrow	N
RA310	Diferencial	Wide, Narrow	Y
RA450	Diferencial	Wide, Narrow	Y
RA3000	Diferencial	Ultra Wide, Wide, Narrow	Y
RA7000	Diferencial	Ultra Wide, Wide, Narrow	Y

UNIDAD DE ALMACENAMIENTO BA356

La opción BA356, no soporta configuraciones de RAID, y tiene una capacidad de almacenamiento de hasta 7 discos o 63 GB de capacidad utilizando discos de 9 GB. La unidad de almacenamiento BA356 soporta conexiones ultra-wide, *SCSI* diferencial y discos ultra-wide (40 MB/sec.). Los clientes pueden utilizar una configuración en cadena de hasta dos gabinetes de almacenamiento sobre un mismo bus *SCSI* para proveer una capacidad total de almacenamiento externo de 14 discos (126 GB de almacenamiento cuando se usa discos de 9 GB).
Esta unidad de almacenamiento está soportada tanto por los servidores Intel, como por los servidores Alpha.

UNIDAD DE ALMACENAMIENTO RA310

Esta unidad puede almacenar internamente hasta 7 discos o 63 GB de capacidad de almacenamiento cuando se usan discos de 9 GB. La unidad de almacenamiento RA310 cuenta con una controladora de RAID de 2 canales fast narrow, en donde cada canal es capaz de manejar hasta 7 discos. Para poder manejar 7 discos más se debe de utilizar la unidad de expansión (FR-SWXRA-Z2 opcional) para así tener un total de 126GB (JBOD) de capacidad de almacenamiento.

La unidad de Almacenamiento RA310, cuenta con una controladora de RAID HSZ20; la cual le permite crear cualquiera de las siguientes opciones de RAID:

* Raid 0
* Raid 0 + 1
* Raid 3
* Raid 5
* Raid 7 (JBOD)

Esta unidad de almacenamiento soporta discos *ultra-wide, wide SCSI* o *narrow SCSI* de cualquier capacidad. Está soportada tanto por los servidores Intel, como por los servidores Alpha.

UNIDAD DE ALMACENAMIENTO RA450

Esta unidad puede almacenar internamente hasta 24 dispositivos *hot-swap*, los cuales pueden ser discos y baterías para el cache. También soporta controladoras de RAID redundantes, eliminando así a las controladoras de RAID como un punto posible de falla. La unidad de Almacenamiento RA450, cuenta con dos o una controladora de RAID HSZ50; la cual es una controladora de 6 canales fast-*narrow*. Cada canal es capaz de soportar hasta 7 dispositivos *SCSI*, aunque no existe un gabinete de expansión esta unidad de almacenamiento puede soportar hasta 24 discos. No existe una segunda unidad de expansión para el RA450. La unidad de almacenamiento RA450 soporta las siguientes opciones de RAID:

* Raid 0
* Raid 0 + 1
* Raid 3
* Raid 5
* Raid 7 (JBOD)

Esta unidad de almacenamiento soporta discos ultra-wide *SCSI*, wide *SCSI* o narrow *SCSI* de cualquier capacidad. Está soportada tanto por los servidores Intel, como por los servidores Alpha.

UNIDAD DE ALMACENAMIENTO RA3000

Esta unidad puede almacenar internamente hasta 7 discos y cuenta con una unidad de expansión (opcional) que le permite controlar hasta 14 discos.

Soporta discos *ultra-wide SCSI*, wide *SCSI* o *narrow SCSI* de cualquier capacidad.

Cuenta con una controladora de RAID HSZ50; la cual le permite crear cualquiera de las siguientes opciones de RAID:

* Raid 0
* Raid 0 + 1
* Raid 3
* Raid 5
* Raid 7 (JBOD)

La controladora HSZ50 tiene dos canales *SCSI* diferentes, lo cual le permite manejar quince dispositivos por canal y tener un mejor desempeño al manejar más discos *SCSI* pues utiliza diferentes LUNs (*Logical Unit Number*).

UNIDAD DE ALMACENAMIENTO RA7000

Esta unidad puede almacenar internamente hasta 24 dispositivos *hot-swap*, los cuales pueden ser discos y baterías para el cache. Soporta controladoras de *RAID* redundantes, eliminando a las controladoras de *RAID* como un punto posible de falla. La unidad de Almacenamiento RA7000, cuenta con dos o una controladora de *RAID HSZ70*, la cual es una controladora de 6 canales *ultra-wide*. Cada canal es capaz de soportar hasta 14 dispositivos *SCSI*, existe un gabinete de expansión para esta unidad de almacenamiento con lo cual puede soportar hasta 72 discos. La unidad de almacenamiento RA7000 soporta las siguientes opciones de *RAID*:

* Raid 0
* Raid 0 + 1
* Raid 3
* Raid 5
* Raid 7 (JBOD)

Esta unidad de almacenamiento soporta discos *ultra-wide SCSI*, *wide SCSI* o *narrow SCSI* de cualquier capacidad. Está soportada tanto por los servidores Intel, como por los servidores Alpha.

Nota: Cabe mencionar que en las unidades RA310, RA450, RA3000, RA7000, no es valido crear arreglos combinando los diferentes tipos de discos *SCSI* que existen, es decir, no se puede crear un arreglo usando discos narrow y wide; pero si se puede tener diferentes tipos de discos almacenados dentro de la unidad de almacenamiento. En algunas ocasiones se pueden crear estructuras de arreglos utilizando diferente tipos de discos *SCSI*, con la desventaja de que este arreglo trabajará utilizando la tecnología *SCSI* de menor desempeño, es decir *narrow SCSI*.

3.2.5 Discos SCSI Soportados

La familia RZ de discos *SCSI* consiste en discos de 3.5 pulgadas, los cuales pueden transmitir datos en 8-bit (*Narrow*) o 16-bit (*Wide*); estos discos pueden girar a una velocidad de 5400 RPM (Revoluciones Por Minutos), 7200 RPM, 10,000 RPM o 15,000 RPM con capacidades que van desde 4 *Gigabytes* hasta 144 *Gigabytes*. Al ofrecer una variedad en capacidad y características de desempeño, es posible fácilmente encajar en cualquier ambiente o aplicación. Estos discos pueden ser montados en sistemas Alpha o Intel, y pueden usarse en unidades de almacenamiento externo.

Sus características principales son las siguientes:

- Los discos de 9.1 *Gigabytes* de 15,000 RPM incrementan el desempeño de la aplicación en un 35% sobre los discos de 10,000 RPM.
- Los discos de 15,000 RPM cuentan con muy alto desempeño, mientras que los discos de 7200 RPM tienen solamente alto desempeño.
- Cuentan con un código de detección de errores de 96 bits lo cual asegura la integridad de los datos.
- Tienen un tiempo de encendido mínimo, realizan pruebas de diagnósticos de manera periódica, las cuales pueden detectar posibles fallas en los discos que se pueden presentar a futuro.
- Soportan temperaturas por arriba de los estándares de los demás discos, lo cual provee mayor confiabilidad en ambientes de desktop o de oficina.

A continuación se listan los discos *SCSI* soportados. Para determinar si se tiene un disco con la versión correcta de *firmware*, se hace uso de la utilería scu

(*System Configuration Utility*) o se examinan los mensajes que son desplegados cuando el sistema es encendido.

Disco	Firmware	Tecnología
RZ26 (1 Gigabyte)	T392 o 392A	Narrow
RZ26L (1 Gigabyte)	442D	Narrow
RZ26L (1 Gigabyte)	442E	Wide
RZ26N (1 Gigabyte)	0466 o mayor	Narrow
RZ26N (1 Gigabyte)	0568 o mayor	Wide
RZ28 (2 Gigabytes)	442C	Narrow
RZ28 (2 Gigabytes)	442E	Wide
RZ28B (2 Gigabytes)	006 o mayor	Narrow
RZ28D (2 Gigabytes)	006 o mayor	Narrow y wide
RZ28M (2 Gigabytes)	0568 o mayor	Narrow y wide
RZ29B (4 Gigabytes)	0011 o mayor	Narrow y wide
RZ1BB (2.1 Gigabytes)	----------	Ultra-wide
RZ1CB (4.3 Gigabytes)	----------	Ultra-wide
RZ1DB (9.1 Gigabytes)	----------	Ultra-wide
RZ1EB (18 Gigabytes)	----------	Ultra-wide

Si se desea seleccionar un disco que se encuentra en un bus *SCSI* compartido, se debe verificar que ese disco no esté siendo utilizado por ningún servicio, a menos que el disco sea parte de un arreglo de discos.

Discos de 5400 RPM

Estos discos son de alta eficiencia, y están disponibles en capacidades de 2.1 Gigabytes y 1.05 Gigabyte. Para las necesidades de almacenamiento de disco adicional, los discos de 2.1 Gigabytes provee una sólida solución para la mayoría de las aplicaciones transaccionales, al igual que en situaciones en donde la necesidad de crecimiento moderado en capacidad de almacenamiento es requerida. Los discos de 5400 RPM de 1.05 Gigabytes entregan una eficiencia aceptable a un costo muy bajo, estos discos son una optima solución para aquellos usuarios que requieren de poca capacidad de almacenamiento.

DISCOS DE 7200 RPM

Estos discos fueron de los primeros en utilizar lo último en tecnología magneto-resistiva para así poder ofrecer a la industria mayor capacidad de almacenamiento en un disco de 3.5 pulgadas. El disco de7200 RPM de 4.3 *Gigabytes* es 30% más rápido que un disco de 5400 RPM en ambientes en donde el procesamiento de transacciones en muy grande. Las mejoras en su cache aumenta la eficiencia en los subsistemas de I/O, mientras que un servomecanismo dedicado permite un acceso rápido a los datos. Los discos de 4.3 *Gigabytes* son ideales para aplicaciones que requieren hacer una gran cantidad de transacciones de IO. Los discos de 2.1 *Gigabytes* se utilizan para aquellos ambientes en donde no se requieren grandes capacidades de almacenamiento. Estos discos son soportados en unidades de almacenamiento externo de tecnología narrow (8 bits).

DISCOS DE 10,000 RPM

Este tipo de discos es el estándar actual en la industria para cualquier tipo de aplicaciones, existen en capacidades de 9, 18, 36, 72 y 144 *Gigabytes*. Estos discos ofrecen un 25% más de desempeño que un disco de 7200 RPM. Son de tecnología Ultra-wide *SCSI* de 16 bits.

DISCOS DE 15,000 RPM

Este tipo de discos es lo último en la industria para aplicaciones altamente demandantes, incluyendo servidores de red y archivos. Existen en capacidades de 9, 18, 36 *Gigabytes*. Estos discos ofrecen un 35% más de desempeño que un disco de 10,000 RPM. Son de tecnología Ultra 3 *SCSI* de 32 bits

3.2.6 CONVERTIDORES SCSI DE SEÑAL

En algunas ocasiones se requiere convertir una señal *SCSI single-ended* a una señal *SCSI* diferencial; y para realizar tal función, se utilizan los convertidores *SCSI* de señal. Los convertidores *SCSI* de señal pueden estar en unidades *standalone* o en unidades tipo SBBs (*Storage Building Blocks*) que pueden ser instalados en un slot de una unidad de expansión.

Un convertidor *SCSI* de señal tiene los siguientes requerimientos:

- Si se observan mensajes de "*BUS Hung*", puede que el convertidor de señal *SCSI* tenga el hardware incorrecto, o bien puede que tenga una revisión de hardware baja y requiera actualizarse.
- Si se desea desconectar un convertidor de señal *SCSI* de un bus *SCSI* compartido, lo primero que debemos de hacer es apagar al convertidor de señal antes de desconectar los cables. Para reconectar al convertidor de señal al bus compartido, simplemente hay que reconectar los cables antes de encender al convertidor de señales.

La siguiente tabla lista a los convertidores *SCSI* de señal que son soportados en un ambiente de cluster.

Dispositivo	Descripción
DWZZA-AA	Unidad *standalone* Convierte de una señal *SCSI single-ended narrow* a una señal *SCSI diferencial wide*
DWZZA-VA	SBB Convierte de una señal *SCSI single-ended narrow* a una señal *SCSI diferencial wide*
DWZZB-AA	Unidad standalone Convierte de una señal *SCSI single-ended wide* a una señal *SCSI diferencial wide*
DWZZB-VW	SBB Convierte de una señal *SCSI single-ended wide* a una señal *SCSI diferencial wide*

3.2.7 CABLES SCSI

El tipo de cables necesarios en un ambiente de cluster, depende de la configuración de hardware. Hay que determinar si se requieren cables con conectores que son de alta densidad (conector largo) o de baja densidad (conector pequeño), 50-pin ó 60-pin. Además, cada cable soportado viene en diferentes longitudes. Es recomendable usar siempre los cables menos largos en un ambiente de cluster.

El requerimiento para usar un cable *SCSI* es el siguiente:

* Verificar que el cable no tenga pines doblados o rotos, así como asegurar el no doblar o romper ningún pin al momento de insertar el cable dentro de un conector.

A continuación se describe cada cable *SCSI* soportado y dónde usarlo.

Cable	# de Conectores Tipo de Densidad	Pins	Uso
BN21W-0B	Tres de alta	68-pines	Este cable tipo Y se puede conectar a KZPSA, HSZ20, HSZ50, o HZS70 y pueden ser terminados si es necesario.
BN21K o BN21L	Dos de Alta	68-pines	Se pueden conectar cables de tipo BN21W o dispositivos *wide*. Por ejemplo, se puede conectar KZPSAs, HSZ20, HSZ50, HSZ70, Los lados diferenciales de dos convertidores de señal *SCSI*, o de un DWZZB-AA a un BA356.

3.2.8 CONECTORES Y TERMINADORES

Dependiendo del tipo de configuración de hardware, puede que se requiera de terminadores o conectores para el bus *SCSI* compartido. En las siguientes tablas se describen los terminadores y conectores soportados y la manera en que pueden ser usados.

Terminador	# de Conectores Tipo de Densidad	Pines	Uso
H879-AA	Alta	68-pines	Puede terminar un conector trilink o un cable BN21W-0B

Terminador	# de Conectores Tipo de Densidad	Pines	Uso
H885 trilink	Tres de Alta	68-pines	Se puede conectar a dispositivos de alta densidad. Cables de 68-pin o dispositivos, tales como una KZPSA, HSZ20, HSZ50, HSZ70, o al lado diferencial de un convertidor de señal *SCSI*, y puede ser terminado si es necesario.

Los conectores *trilink* tienen los siguientes requerimientos:

* Si se conecta un cable a un conector *trilink*, no hay que bloquear el acceso a los tornillos que sujetan al *trilink*, de lo contrario no será posible desconectar el trilink del dispositivo.

3.3 REQUERIMIENTOS Y CONFIGURACIÓN DEL BUS SCSI

En este capítulo se verán algunos requerimientos necesarios para la instalación y configuración del bus *SCSI*.

3.3.1 REQUERIMIENTOS PARA LA INSTALACIÓN DE UN BUS SCSI

Para configurar un bus *SCSI* compartido se deben cumplir los siguientes requerimientos:

* Un cluster requiere que todos los buses compartidos tengan el mismo número lógico de bus en cada nodo. Además los números lógicos asignados a los adaptadores de los buses compartidos deben ser mayores que los números asignados a los buses locales. Ver la sección 3.2 para más información.
* Únicamente los buses externos pueden ser compartidos en un cluster.
* Se pueden usar hasta 30 buses compartidos en un cluster.
* La longitud física para cada bus compartido esta estrictamente limitada. Esta longitud depende de la velocidad que se está utilizando, *SCSI fast* o *slow*, *SCSI single-ended*, o *SCSI* diferencial. Ver la sección 3.3.5 para más información.
* Cada bus debe de ser terminado sólo al final. El exceso o la carencia de terminadores pueden causar que un bus *SCSI* se comporte de manera impropia. Ver la sección 3.3.6 para más información.

- Para conectar los dispositivos a un bus compartido se utilizan los conectores *trilink* o cables Y. Esto permite unir un dispositivo al cluster sin afectar la terminación del bus. Ver la sección 3.3.6 para más información.
- Si se desea conectar dispositivos con diferentes métodos de transmisión y rutas de datos se requiere de un convertidor de señal *SCSI*.
- Se necesita de mucho cuidado al realizar algún mantenimiento a cualquier dispositivo que se encuentre en un bus *SCSI* compartido ya que existe constante actividad sobre el bus. Usualmente para realizar algún mantenimiento sobre un dispositivo sin dar de baja el cluster, debe existir la capacidad de aislar el dispositivo del bus compartido sin afectar la terminación del bus. Ver la sección 3.2.8 para más información.
- Al desconectar una unidad de almacenamiento de un bus compartido (sin afectar la terminación del bus) o al remover un disco de un *slot*, el cluster detiene cualquier servicio que use al disco, a menos que el disco sea parte de un arreglo en espejo o RAID 5.

3.3.2 Numerando Buses SCSI

Todos los nodos de un cluster deben reconocer los discos que se encuentran en el bus compartido en el mismo número de dispositivo. El número de dispositivo es obtenido del número lógico de bus que es definido en la configuración. Si conectamos un bus compartido a un controlador *SCSI* que tiene el mismo número lógico de bus en cada nodo, los discos compartidos tendrán el mismo número de dispositivo en cada sistema.

Los números de bus son asignados a las controladoras *SCSI* durante el proceso de configuración del kernel, siendo especificados en el archivo de configuración. En el programa de configuración del kernel de un servidor se ejecuta un algoritmo que es usado para probar los controladores *SCSI* instalados en el sistema. Conforme el algoritmo va detectando controladoras, les va asignado números lógico de bus en secuencia, comenzando con el 0.

Antes de instalar las controladoras *SCSI*, se debe planear la configuración de bus. Es recomendable que las controladoras *SCSI* de los buses locales se instalen en los *slots* inferiores del sistema, dejando si es posible algunos vacíos, posteriormente instalar las controladoras *SCSI* de los buses compartidos. Este método permite instalar controladoras *SCSI* adicionales tanto para los buses locales como buses compartidos sin afectar el esquema de numeración de buses

compartido. Por ejemplo, si un máximo de ocho controladoras locales van a ser instaladas en un sistema, la primera controladora de bus compartido se instalará en el *slot* 8, la siguiente controladora de bus compartido se instalará en el *slot* continuo que sea superior, y así sucesivamente.

Algunos controladores *SCSI* cuentan con dos puertos (o canales), de tal manera que es posible conectar dos buses compartidos en cada controladora. Por ejemplo, un módulo PMAZC tiene los puertos A y B, un adaptador KZMSA tiene los canales 0 y 1. Ambos puertos no tienen que ser utilizados necesariamente, pero todo puerto no utlizado debe de ser terminado.

Si se utilizan controladores *SCSI* de dos puertos, un bus compartido debe estar conectado al mismo puerto en cada sistema. Por ejemplo, si un bus compartido está conectado al puerto A (o canal 0) sobre un controlador *SCSI*, entonces debe estar conectado al puerto A (o canal 0) de todas las demás controladoras *SCSI*.

3.3.3 Desempeño del Bus SCSI

3.3.3.1 Método de Transmisión

Como se mencionó en el Capítulo II, si se desea conectar un dispositivo diferencial y un dispositivo *single-ended* se debe de usar un convertidor de señal *SCSI* entre los dispositivos ya que no pueden ser usados en un mismo bus físico.

- Los dispositivos single-ended deben de incluir lo siguiente:
  - Un módulo PMAZC
  - Un módulo KZMSA
  - Un extremo *single-ended* de un convertidor de señal *SCSI*.
  - Unidades de almacenamiento BA350, BA353, y BA356.

- Actualmente Digital incluye los siguientes dispositivos diferenciales:
  - Adaptador KZPSA
  - El lado diferencial de un convertidor de señal *SCSI*
  - Los controladores HSZ20 y HSZ40

3.3.3.2 Ruta de Datos

Cabe mencionar que en la ruta de datos, usualmente pero no siempre, los dispositivos *single-ended* son *narrow*, y los dispositivos diferenciales son *wide*. La unidad de almacenamiento BA356 que tiene diferentes rutas de datos, esto lo hace mediante el uso de un convertidor de señal *SCSI* entre los dispositivos.

3.3.3.3 Velocidad del BUS

Como ya se dijo, el medio para establecer la velocidad de un bus *SCSI* es a través de comandos de consola o mediante la configuración que viene en el *firmware* de la controladora; el método a usar depende del tipo de controladora *SCSI*.

Aunque el bus fast *SCSI* dobla la velocidad de transmisión a 10 millones de bytes por segundo, tiene como desventaja que reduce la longitud máxima del cable para cada bus *single-ended* de 6 metros a 3 metros. Ver la sección 3.3.5 para mayor información.

3.3.4 Número de Identificación de Dispositivos

Sobre un bus *SCSI*-2, la especificación *SCSI* limita el número de dispositivos a 16, donde cada dispositivo (controladora *SCSI* o disco) debe de tener un *SCSI ID* único (de 0 a 15). Por ejemplo, si contamos con dos sistemas miembros en un cluster, únicamente podremos contar con catorce discos sobre el bus compartido, al menos que se esté utilizando un subsistema DEC *RAID*.

Cada puerto controlador *SCSI* usualmente tiene como número de identificación el 7. Dependiendo de la controladora *SCSI*, podemos usar ya sea comandos de consola o la utilería *LFU* para fijar el *SCSI ID* del puerto. Si se están utilizando ambos puertos de una controladora *SCSI* con dos puertos, los puertos pueden tener el mismo *SCSI* ID por que cada puerto va a ser conectado a un bus SCSI compartido diferente.

Se recomienda usar el siguiente orden para asignar un *SCSI ID* a una controladora SCSI:

7 - 6 - 5 - 4 -3 - 2 - 1 - 0

El orden anterior especifican que el número 7 es el número de mayor prioridad, y 0 es el de menor prioridad. Cuando se le asigna un *SCSI ID* a un sistema miembro, Digital recomienda que se use el número de *ID* mas alto, comenzando en 7. Después, se debe de usar los ID menores para los discos.

El *SCSI* ID que se le asigna a un disco en una unidad de almacenamiento BA356 corresponde a la colocación del disco en la unidad.

En las controladoras HSZ20 y HSZ40 el *SCSI* ID se tiene que fijar para cada uno de sus puertos. Para asignar un *SCSI* ID a cualquiera de estas controladoras debemos de recurrir a la consola de comandos (por lo general se utiliza el siguiente comando: HSZ> set this id=n, donde n represente el *SCSI* ID que se le desea asignar a la controladora).

3.3.5 Longitud del Bus SCSI

Como se mencionó anteriormente, la longitud de cada bus físico es limitada, la cual depende de la velocidad del bus y del método de transmisión utilizado, ya sea *SCSI single-ended* o *SCSI* diferencial.

Si se utilizan dispositivos que tienen el mismo método de transmisión y ruta de datos, el bus compartido consistirá en un solo bus físico. Si por el contrario, se utilizan convertidores de señal *SCSI* para conectar dispositivos con diferentes métodos de transmisión o rutas de datos, el bus compartido consistirá en un solo bus *single-ended* o diferencial, en donde el bus deberá estar terminado y además seguir las reglas de longitud de bus.

Debido al límite de longitud para un bus *SCSI*, se debe planear cuidadosamente la configuración de hardware, por lo que es recomendable colocar los sistemas y las unidades de almacenamiento tan cerca como sea posible y elegir los cables mas cortos para el bus *SCSI* compartido.

3.3.6 CONFIGURACIÓN DEL BUS *SCSI*

3.3.6.1 TIPOS DE CONEXION Y UNIDADES DE ALMACENAMIENTO

Es básico que los dispositivos se encuentren propiamente conectados al bus *SCSI* compartido. Como ya se mencionó, únicamente el principio y el final de cada bus físico debe estar terminado. Ver capítulo 2 para más información.

Los dispositivos que forman parte del bus *SCSI* compartido deben estar conectados de tal forma que puedan ser removidos del mismo. Esto nos permite desconectar dispositivos para propósitos de mantenimiento, sin afectar la terminación del bus y sin dar de baja al cluster. Además, si los dispositivos se conectan de cierta forma, es posible adicionar nuevos dispositivos al bus sin afectar la terminación.

La mayoría de los dispositivos cuentan con terminación interna o algún otro método de terminación. Por ejemplo, los adaptadores KZPSA, las unidades de almacenamiento BA356, y los convertidores de señal *SCSI* cuentan con terminación interna, pero existen otros controladores que cuentan con terminación automática.
Dependiendo de la instalación del bus compartido, se debe de habilitar o deshabilitar esta terminación.

Aunque una posibilidad es la terminación interna de los dispositivos en un bus compartido, si desconectamos el cable que conecta a un dispositivo, el bus no se encontrará terminado. Por tal motivo es recomendada la terminación externa, que permite sacar cualquier dispositivo del bus compartido sin afectar la terminación del mismo.

Para una terminación externa, que sea capaz de desconectar y conectar dispositivos sin afectar la terminación del bus, se recomienda usar uno de los siguientes cables o conectores:
Un cable Y puede conectarse a un módulo PMAZC, un adaptador KZTSA, o un adaptador KZPSA al que ha sido removida su terminación interna.

Un conector *trilink* puede conectarse a un adaptador KZTSA, o un adaptador KZPSA al que ha sido removida su terminación interna, un controlador HSZ20, HSZ70, o del lado diferencial (no terminado) de un convertidor de señal *SCSI*.

Si se utiliza un cable Y o un conector *trilink* a un dispositivo sin terminación, es posible colocar este dispositivo en medio o al final del bus. Si el dispositivo se encuentra al final, el terminador debe de conectarse en al cable Y o al conector *trilink* para así terminar el bus. De tal manera, que si se desconecta al cable Y o al conector *trilink* del dispositivo, el bus compartido se encontrará aún terminado, y el cluster seguirá funcionando correctamente. Si se conecta un cable Y o un conector *trilink* a un bus compartido sin conectarlo a ningún dispositivo, es posible conectar al cable Y o al conector *trilink* a un dispositivo y así expandir nuestra configuración.

En la *figura 3.3* se muestra un cable BN21V-0B que conecta a un modelo PMAZC al que le ha sido removida la terminación interna.

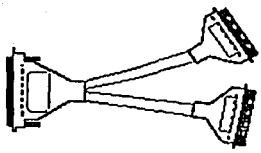


Figura 3.3 - Cable BN21V-0B.

En la *figura 3.4* se muestra un cable Y BN21W-0B, el cual une un adaptador KZTSA o un adaptador KZPSA al que le ha sido removida la terminación interna.

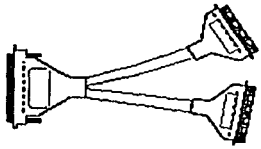


Figura 3.4- Cable "Y" BN21W-0B.

La *figura 3.5* muestra un conector *trilink*, el cual une un adaptador KZTSA o un adaptador KZPSA al que le ha sido removida la terminación interna, un controlador HSZ20 o HSZ50, o del lado diferencial (no terminado) de un convertidor de señal *SCSI*.

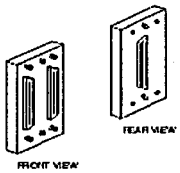


Figura 3.5- Conector trilink.

La *figura 3.6* muestra la configuración de hardware de un cluster con dos nodos DEC 3000 Modelo 500 con módulos PMAZC instalados, una unidad de almacenamiento BA350, y un bus *single-ended*. Un cable BN21V-0B es conectado al puerto no terminado A en cada módulo PMAZC. (Al puerto B que no es utilizado se le coloca un terminador). La unidad de almacenamiento BA350 se encuentra en la mitad del bus, de tal manera que su terminación interna es removida. El bus compartido es terminado por un terminador H8574-A o H8860-AA, el cual es conectado a uno de los extremos del cable BN21V-0B.

Si se utiliza esta configuración y un cable Y es desconectado de un módulo PMAZC, el servidor que lo contiene no estará disponible. Sin embrago, el cluster sigue operando ya que la terminación del bus se mantiene, como se muestra en la *figura 3.7*.

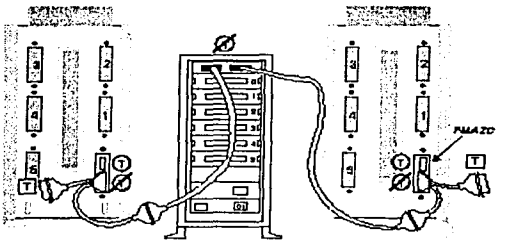


Figura 3.6- Bus single-ended terminado con cables "Y".

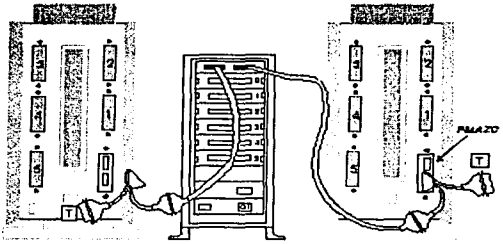


Figura 3.7 - Desconectando un cable "Y".

Al hardware descrito en la *figura 3.8*, se le puede crear una configuración alterna colocando uno de los sistemas en medio del bus compartido y la unidad de almacenamiento al final. En este caso, la terminación interna de la unidad de almacenamiento BA350 es usada para terminar al bus, como se ve en la *figura 3.8*.

Sin embargo, en la configuración de la *figura 3.8*, al desconectar el cable de la unidad de almacenamiento BA350, el bus *single-ended* no se encontrará terminado y la operación del cluster será interrumpida. Para que sea posible desconectar una unidad de almacenamiento *single-ended* de un bus *SCSI* compartido, se debe conectar la unidad de almacenamiento del lado *single-ended* de

un convertidor de señal *SCSI* y, por el lado diferencial, al conector *trilink*. Con dichas modificaciones se podrá desconectar tanto la unidad de almacenamiento como el convertidor de señal del bus compartido sin afectar la operación del cluster.

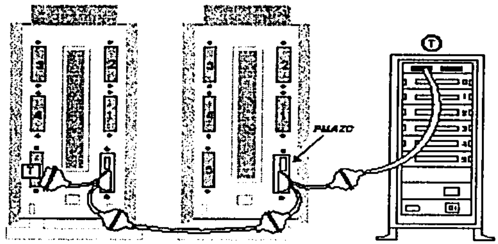
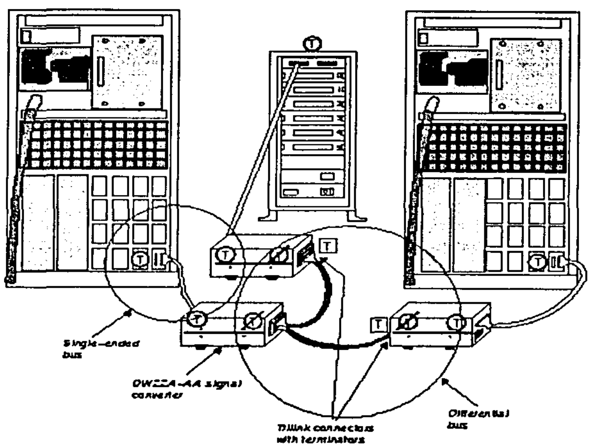


Figura 3.8 - Bus single-ended terminado con una unidad de almacenamiento.

La *figura 3.9* nos muestra una configuración de hardware que usa convertidores de señal. Todo el bus compartido consiste en tres buses *single-ended* y un bus diferencial.

Figura 3.9 - Configuración de un bus mediante el uso de convertidores de señal *SCSI*.



En la *figura 3.9*, se tiene un bus *single-ended* entre cada dispositivo *single-ended* (adaptador KZMSA o unidad de almacenamiento BA350) y el lado *single-ended* de un convertidor de señal DWZZA-AA. El bus *single-ended* es terminado de forma interna en el dispositivo y en el convertidor de señal, como se muestra en la *figura 3.10*. Si un cable de un dispositivo *single-ended* es desconectado, el bus *single-ended* no esta terminado, es decir se encontrará abierto; sin embargo, el bus diferencial no es afectado incluso si apagáramos el convertidor DWZZA.

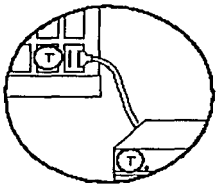


Figura 3.10 - Terminación de un bus *single-ended*.

En la *figura 3.9*, el bus diferencial es conectado a un *trilink* el cual está unido al lado diferencial (no terminado) de cada convertidor de señal DWZZA-AA. El bus diferencial es terminado colocando terminadores en los conectores *trilink* al final de bus, como se muestra en la Figura 3.11. Si un conector *trilink* es desconectado de un DWZZA-AA, el bus diferencial sigue estando terminado.

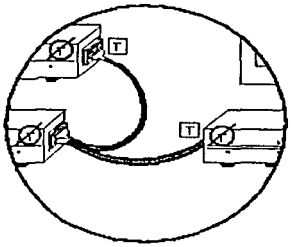


Figura 3.11 - Terminación de un bus diferencial.

CAPÍTULO IV.- CLUSTER EN WINDOWS NT

4.1 ¿QUÉ ES UN CLUSTER EN WINDOWS NT?

En los capítulos anteriores hemos estado definiendo lo que es un cluster, pero para Microsoft Cluster Server, un cluster se define como un grupo de computadoras independientes que trabajan en conjunto como si fueran una sola. Esto nos permite que las computadoras sean accedidas y administradas como un único sistema, en vez de que sean accedidos como sistemas distintos. Para completar esto, todas las computadoras en el cluster son agrupadas bajo un nombre en común, el nombre del cluster, el cual es usado para acceder y administrar el cluster. Actualmente Microsoft Cluster Server tiene la capacidad de formar un cluster de dos nodos únicamente.

Cada servidor que es miembro de un cluster es llamado nodo. Los nodos de un cluster deben de tener una red privada entre ellos, esta red va a tener como función el permitir la comunicación entre los nodos sin afectar el tráfico normal de red pública. Además ambos nodos también deben de estar conectados a un dispositivo de almacenamiento compartido (ya sea *SCSI* o de Fibra). Es en este dispositivo de almacenamiento compartido en donde la información que va a compartir el cluster va a ser almacenada.

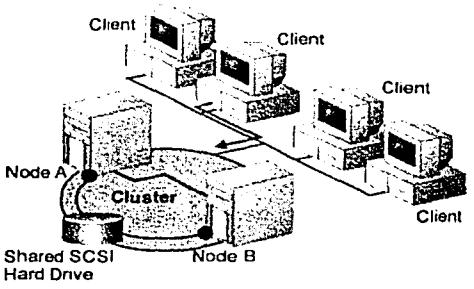


Figura 4.1 - Descripción de Cluster.

Ahora antes de profundizar más en lo que a Microsoft Cluster Server (MSCS) se refiere, se deben conocer los siguientes términos que estaremos usando en este capítulo:

- Recursos

  Un recurso es una entidad lógica o física, como por ejemplo un archivo compartido, administrado por el servicio de Cluster. Un recurso provee un servicio a los clientes. Estos son la unidad básica que puede ser administrada por el servicio de Cluster. Un recurso únicamente puede correr sobre un solo nodo a la vez dentro de un esquema de cluster.

- Dependencias

  Una dependencia es una relación o regla entre dos recursos que se necesita cumplir para que ambos recursos puedan correr sobre un mismo nodo. Por ejemplo, un recurso de archivo compartido (file share resource) depende de que el recurso de disco (disk resource) con un subdirectorio que pueda ser compartido esté en línea.

- Grupos

  Los grupos son una colección de recursos con propósitos de configuración y administración. Si un recurso depende de otro recurso, cada uno de estos recursos deben de pertenecer al mismo grupo. En el ejemplo del recurso de archivo compartido, el recurso de archivo compartido debe de pertenecer al mismo grupo que contiene al recurso de disco. Todos los recursos que se encuentran dentro de un grupo deben de estar en línea en el mismo nodo del cluster.

- *Failover*

  En este esquema vamos a entender como *failover* al proceso de mover un recurso o un grupo de recursos de un nodo a otro en caso de que se presente una falla. Por ejemplo, en un cluster donde se esté ejecutando Microsoft Internet Information Server (IIS) sobre el nodo A y éste llegase a fallar, entonces el recurso de IIS se va a mover al nodo B, o también se dice que acaba de ocurrir un *failover* sobre el recurso de IIS.

- *Failback*

  Se entiende como *failback*, al proceso de regresar un recurso o grupos de recursos al nodo en el cual estaban corriendo antes de que ocurriera un *failover*.

Extendiendo un poco el ejemplo anterior, cuando el nodo A vuelva a estar en línea otra vez, entonces el recurso de IIS a regresar al nodo A, o también se dice que el recurso de IIS hizo un *failback* al nodo A.

- Recurso *Quórum* (*Quórum Resource*)
  El *quórum resource* es el recurso que almacena la información de administración del cluster, tal como los archivos de log de recuperación para realizar los cambios hechos al cluster. Éste es accesible a todos los nodos del cluster. MSCS requiere de un almacenamiento de disco compartido entre los nodos del cluster. Un disco de este almacenamiento compartido es elegido como *quórum resource* por default en MSCS.

4.2 BENEFICIOS DE UN CLUSTER BAJO WINDOWS NT

Los beneficios que podemos obtener de implantar un esquema de cluster bajo Windows NT, son los siguientes:

✓ **Alta Disponibilidad.-** Un cluster está diseñado para proveer alta disponibilidad de los recursos a los clientes, en la forma de que si un nodo falla, los recursos que se estaban ejecutando sobre este nodo van a ser automáticamente movidos a otro nodo. Aunque los recursos no siempre van a estar disponibles a los clientes (No es disponibilidad constante).

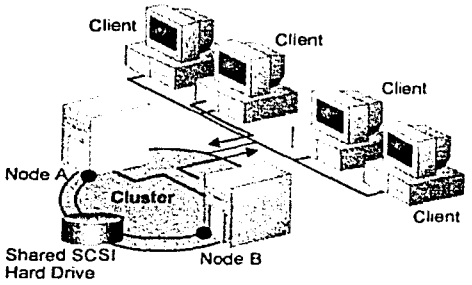


Figura 4.2 - Beneficios de un Cluster.

- ✓ **Manejabilidad.-** MSCS permite al administrador administrar ambos nodos del cluster como si fuera una sola entidad. Además, la utilería de Cluster Administrator permite al administrador poner casi cualquier recurso, incluyendo aplicaciones de red y servicios, fuera de línea con propósitos de mantenimiento.

- ✓ **Escalabilidad.-** La escalabilidad se puede llevar a cabo combinando una existente, sobre utilizada computadora con una nueva computadora. Una vez que el cluster ha sido creado, los recursos de la computadora sobre saturada pueden repartirse entre ambos nodos del cluster. Esto ayuda a balancear la carga de trabajo y disminuir la carga de trabajo en la computadora original.

4.3 MODELOS DE IMPLEMENTACIÓN DE CLUSTER EN WINDOWS NT

Actualmente existen dos modelos de implementar un cluster que se están usando actualmente en la industria, el modelo de *dispositivos compartidos* (shared device) y el modelo de *nada compartido* (shared nothing).

Es posible para un cluster el soportar ambos modelos a la vez. Típicamente, las aplicaciones que requieren de acceso limitado a los recursos compartidos el mejor modelo a aplicar es el de dispositivos compartidos. Aquellas aplicaciones que requieren de máxima escalabilidad se verán enormemente beneficiadas con un modelo de nada compartido.

Microsoft Cluster Server soporta de manera natural al modelo de Nada Compartido. Sin embargo, también puede soportar el modelo de dispositivos compartidos siempre y cuando la aplicación cuente con un DLM (*Distributed Lock Manager*).

4.3.1 MODELO DE DISPOSITIVOS COMPARTIDOS

En el modelo de dispositivos compartidos (*shared device model*), el software que se está ejecutando en cualquier computadora en el cluster puede tener acceso a cualquier recurso de hardware conectado a cualquier computadora en el cluster (por ejemplo, un disco duro). Sólo dos aplicaciones requieren acceder a un mismo dato,

algo similar a una computadora de multiprocesamiento simétrico (symmetric multiprocessor SMP), el acceso a los datos debe de ser sincronizado. En la mayoría de los cluster de este modelo, un componente llamado Administrador de Candados Distribuidos (*Distributed Lock Manager* DLM) es usado para manejar estas situaciones.

El DLM es un servicio que es provisto a las aplicaciones que están corriendo en el cluster que deja pistas de referencias a los recursos de hardware dentro del cluster. Si múltiples aplicaciones intentan acceder a un mismo recurso de hardware, el DLM detecta y resuelve el conflicto. Sin embargo, el utilizar al DLM crea cierta cantidad de trabajo extra dentro del sistema, este trabajo extra se genera por la creación de mensajes adicionales entre los nodos del cluster, también el DLM genera una pérdida en el desempeño debido a que se genera una cola para poder acceder al recurso de hardware.

4.3.2 Modelo de Nada Compartido

El modelo de nada compartido está diseñado para evitar la sobrecarga de trabajo que genera el DLM en modelo de dispositivos compartidos. En este modelo, cada nodo del cluster es dueño de un conjunto de recursos de hardware que conforman al cluster. Como resultado, sólo un nodo puede ser dueño y accesar a un recurso de hardware a la vez. Cuando existe una falla, otro nodo se apropia del recurso de hardware de tal manera que el recurso de hardware siempre puede ser accedido.

En este modelo, las peticiones de las aplicaciones son automáticamente enrutadas al sistema que es dueño del recurso. Por ejemplo, si una aplicación necesita acceder a una base datos que se encuentra en un disco duro que pertenece a un nodo que no es en donde se está ejecutando la aplicación, el nodo pasa la petición de acceso al otro nodo. Esto permite la creación de aplicaciones que pueden estar distribuidas a través de los nodos del cluster.

4.4 ARQUITECTURA DE MICROSOFT CLUSTER SERVER

El servicio de cluster (Clussvc.exe) y los monitores de recurso (Resrcmon.exe) son las llaves para crear un cluster en MSCS. Cada nodo en un cluster contiene todos los componentes que aparecen en la figura 4.3.

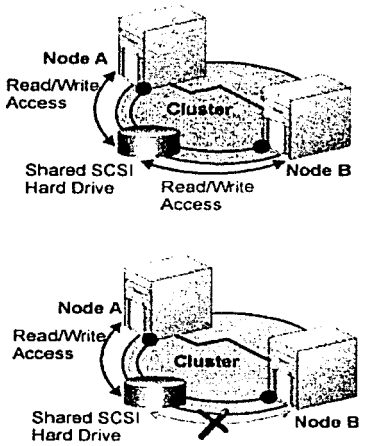


Figura 4.3 - Arquitectura de Microsoft Cluster Server

4.4.1 ADMINISTRADORES DEL SERVICIO DE CLUSTER

El servicio de cluster es un servicio de Windows NT que corre en cada nodo del cluster. Este servicio consiste de los siguientes servicios de administración:

- Administrador de Base de Datos (*Database Manager*).- Administra y mantiene la configuración de la base de datos del cluster.

- Administrador de Nodos (*Node Manager*).- Este administrador mantiene la calidad de miembro del cluster, en otras palabras se encarga de administrar qué nodos son miembros de cluster.
- Procesador de Eventos (Event Processor).- Inicializa el servicio de cluster y reporta los eventos entre los otros componentes.

- Administrador de Comunicaciones (Communication Manager).- Maneja la comunicación entre los nodos.

- Administrador Global de Actualizaciones (Global Update Manager).- Realiza las actualizaciones que se lleven a cabo en el cluster.

- Administrador de Recursos y *Failovers* (Resource/Failover Manager).- Administra los recursos de cluster.

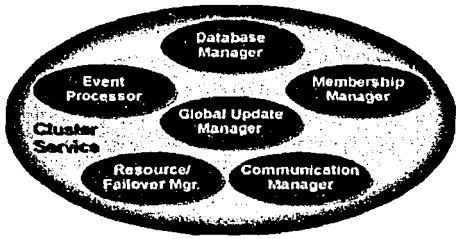


Figura 4.4 - Administradores del Servicio de Cluster

4.4.1.1 ADMINISTRADOR DE BASE DE DATOS

El administrador de Base de Datos es el componente del Servicio de Cluster que implementa la base de datos de cluster. La base de datos contiene la información de todas las entidades en el cluster, tales como el cluster mismo, tipos de recurso, grupos, y recursos. La base de datos del cluster es almacenada en el "*registry*" (es la base de datos en donde se guarda toda la configuración de Windows NT) de cada nodo del cluster.

El administrador de Base de Datos, se encuentra en cada nodo del cluster, y cooperan entre sí para mantener consistente la información de configuración del cluster.

Además, el Administrador de Bases de Datos también provee una interface para configurar la base de datos, la cual es usada por los otros componentes del Servicio de Cluster. Los Servicios de Cluster coordinan las actualizaciones en el "*registry*" para mantenerlo constante y a su vez actualiza a los otros nodos del cluster.

4.4.1.2 Administrador de Nodos

El administrador de nodos sobre un nodo en el cluster se comunica con el Administrador de Nodos sobre el otro nodo con el fin de detectar fallas en los nodos del cluster. Esto se lleva a cabo por medio de mensajes "latido de corazón (heartbeat)" entre los nodos.

Si el Administrador de Nodos sobre un nodo no responde, los recursos activos deben de realizar un *failover* aún cuando los recursos se encuentren en línea y funcionando. El administrador de recursos y *failover* se encargan de ejecutar el *failover*.

Si el Servicio de Cluster llegase a fallar en un nodo, todos los recursos sobre ese nodo realizarían un *failover* aún si los recursos se encontrasen en línea y funcionando. Esto pasa como resultado de la inicialización de un *failover* por el Administrador de Nodos cuando éste no se puede comunicar con el Administrador de Nodos del otro nodo.

¿Que pasaría si la comunicación entre los nodos llegase a fallar, pero ambos nodos se encuentran funcionando correctamente?, o en otras palabras ¿Qué ocurriría si se presenta una partición de red?, lo que pasa es que el Administrador de Nodos en cada nodo van a tratar de tomar los recursos al nodo en donde se encuentra corriendo este Administrador de Nodos, que para este caso son los dos. Esto nos da como resultado que cada nodo piensa que es el único nodo sobreviviente del cluster y va tratar de traer todos los recursos del cluster en línea. Para prever esta situación, el Servicio de Cluster confía en que únicamente un nodo del cluster tiene el recurso de quórum en línea. Si la comunicación entre nodo falla, el nodo que

tiene el control del recurso de *quórum* va a traer a todos los recursos en línea. El nodo que no tiene acceso al recurso de quórum y no puede comunicarse va a poner todos los recursos que tienen fuera de línea.

4.4.1.3 Procesador de Eventos

El Procesador de Eventos es el centro de comunicaciones del Servicio de Cluster. Es el responsable de conectar a los eventos con las aplicaciones y los componentes del Servicio de Cluster. En concreto el Procesador de Eventos tiene como tareas:

- Mantener los objetos del cluster.
- Atiende las peticiones de las aplicaciones para abrir, cerrar, o enumerar los objetos del cluster.
- Dirige todos los eventos a las aplicaciones *cluster-aware* (aplicaciones desarrolladas para funcionar en cluster de manera natural) y a los otros componentes del Servicio de Cluster.

El procesador de Eventos también es responsable de inicializar el Servicio de Cluster y de poner al nodo en el estado de "fuera de línea (offline)". El procesador de eventos es el encargado de decirle al Administrador de Nodos que comience el proceso de crear o unirse a un cluster.

4.4.1.4 Administrador de Comunicaciones

Todos los componentes del Servicio de Cluster se comunican con el Servicio de Cluster de otro nodo a través del Administrador de comunicaciones. El administrador de comunicaciones es el responsable de:

- Mantener el protocolo "*keepalive*".- Checa si existe alguna falla del Servicio de Cluster sobre los nodos del cluster.
- Mover los grupos.- Inicia el proceso de *failover* de los recursos de un nodo a otro.

- Negocia el próximo dueño de un recurso.- Determina quién será el nuevo dueño de los recursos de un nodo que falló.

- Comunica el estado de los recursos.- Notifica al resto de los nodos del cluster cuando un recurso se pone fuera de línea o se pone en línea. Éste se utiliza para saber quién es el dueño de un recurso.

- Enlista los nodos del cluster.- Inicia el primer contacto con el cluster.

- Une el nodo al cluster.- Sincroniza al nodo del cluster que se encontraba fuera de línea con el nodo que se encuentra en línea cuando el primero se pone en línea.

- Actualiza la Base de Datos.- Actualiza la base de datos del cluster a través de un proceso de "*Two Phase Commit*".

4.4.1.5 Administrador Global de Actualizaciones

El Administrador Global de Actualizaciones provee una interfaz para los otros componentes del Servicio de Cluster para iniciar y administrar las actualizaciones. El Administrador Global de Actualizaciones permite que los cambios de estado de los recursos (pasar de en línea a fuera de línea y viceversa) sean fácilmente propagados a través de los nodos del cluster. Además, las notificaciones de cambio de estado del cluster también son notificadas a todos los nodos activos en el cluster.

4.4.1.6 Administrador de Recursos y Failovers

El Administrador de Recursos y Failovers es el responsable de:

- Administrar las dependencias entre recursos.
- Pone en línea o fuera de línea a los recursos.
- Inicia el proceso de *failover* o *failback*.

Con el fin de ejecutar las tareas arriba listadas, el Administrador de Recursos y *Failovers* recibe la información del estado de los recursos y el cluster

de los recursos de monitoreo y el Administrador de Nodos. Si por alguna causa un recurso se vuelve inaccesible, el Administrador de Recursos y Failovers van a intentar a volver a poner al recurso en línea sobre el mismo nodo o va a iniciar el proceso de *failover* del recurso.

Cuando el Administrador de Recursos y *Failovers* es notificado de que uno de los recursos que se encontraba en línea se ha vuelto inaccesible, dependiendo de los parámetros de *failover*, va a poder escoger no intentar reiniciar el recurso y en su lugar poner al recurso fuera de línea junto con todos los recursos dependientes a éste. Una vez que el recurso es puesto fuera de línea, se inicia un proceso de *failover*, y después todo el grupo va a ser empujado al otro nodo en el cluster. A esta situación se le conoce como *empujar a un grupo (pushing a group) a otro nodo.*

¿Cómo trabaja esto?, bueno el proceso de empujar un grupo trabaja de la siguiente manera:

1. Todos los recursos que dependen de este recurso son enumerados según el árbol de dependencias, esto incluye a todos los recursos que dependen del recurso que falló y a los recursos que se suponen dependen de este recurso.

2. El Administrador de Recursos y *Failovers* pone a todos los recursos que se encuentran el árbol de dependencias fuera de línea. Estos recursos son puestos fuera de línea de acuerdo a sus dependencias, secuencialmente y sincronizadamente.

3. Se inicia el proceso de *failover*. El Administrador de Recursos y *Failovers* sobre el nodo que previamente era el dueño del recurso notifica al Administrador de Recursos y *Failovers* sobre el nodo destino que se está ejecutando un *failover*.

4. El Administrador de Recursos y *Failovers* del nodo destino comienza el proceso de poner en línea a los recursos, en orden contrario a como éstos fueron puestos fuera de línea.

Cuando un nodo en el cluster falla, los recursos que se encontraban en este nodo deben de ser movidos del nodo que falló al nodo que sobrevivió. A este proceso lo conocemos como *jalar un grupo (pulling a group).* Este proceso es muy similar al proceso de empujar un grupo con la excepción de que no tiene que poner ningún recurso fuera de línea porque lo que falló fue todo el nodo.

Cuando un nodo ha fallado y después éste vuelve a estar en línea, el Administrador de Recursos y *Failovers* en el nodo que acaba de regresar en línea inicia los procesos de *failback* configurados. Simplemente tiene que conectarse con el Administrador de Recursos y *Failovers* del nodo que actualmente tiene los recurso en línea y le dice que le regrese los recursos.

Existe una opción de *failback* que puede ser configurada con el fin de candelarizar la hora del día en que debe de iniciarse el proceso de *failback*. Si la venta de *failback* ha sido configurada y habilitada, el Administrador de Recursos y *Failovers* va a esperar hasta la hora designada para iniciar el proceso de *failback*.

4.4.2 COMPONENTES ADICIONALES DE LA ARQUITECTURA DE MICROSOFT CLUSTER SERVER

Los Monitores de Recursos y las librerías dinámicas de los recursos (DLLs) manejan las comunicaciones entre el Servicio de Cluster y los Administradores de Recursos a través del hardware del cluster y sus aplicaciones. Por ejemplo, el Servicio de Cluster usa a los monitores de recurso y a las librerías dinámicas para determinar cuándo falló una aplicación o cuándo se puso en línea.

Microsoft Cluster Server utiliza otros recurso aparte del Servicio de Cluster y de los monitores de recurso, las cuales son:

- Las aplicaciones de Administración del cluster.- Estas aplicaciones realizan llamadas al Servicio de Cluster usando la aplicación de administración del cluster. La aplicación de Administración de Cluster incluida en MSCS es un ejemplo de tales aplicaciones.

- Las aplicaciones *Cluster Aware*.- Ésta es una aplicación que corre sobre un nodo en el cluster y puede inherentemente tomar ventaja de las características del cluster. Típicamente las aplicaciones cluster *aware* se comunican con el Servicio de Cluster usando la interfaz de administrador del cluster o utiliza una librería dinámica en específico.

- Las aplicaciones que no son *Cluster Aware*. Éstas son aplicaciones que corren sobre un nodo en el cluster pero no está consciente de las características del

cluster. Debido a que estas aplicaciones no son cluster *aware*, la relación de éstas con el Servicio de Cluster es únicamente a través de las librerías dinámicas.

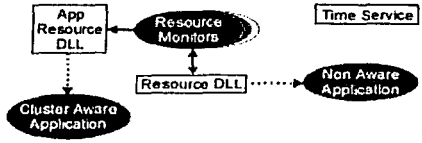


Figura 4.5 - Componentes Adicionales de la Arquitectura de MSCS

4.4.2.1 Monitores de Recursos

Los monitores de recurso dejan pistas de los estados (en línea o fuera de línea) en los que se encuentran los recursos en un nodo, notificando al Servicio de Cluster sobre cualquier cambio. Es posible que en cada nodo existan múltiples monitores de recursos, donde cada uno de los cuales monitorea uno o varios recursos. Por default el Servicio de Cluster cuenta con un solo monitor de recursos para todos los recursos en el nodo. Es posible con MSCS el configurar un monitor de recurso para cada recurso.

Con el fin de prevenir que los monitores de recursos causen problemas al Servicio de Cluster, como ocasionar que el servicio falle, cada monitor de recurso es ejecutado dentro de su propio proceso. Puede presentarse el caso en donde una librería dinámica llegase a fallar y como consecuencia un monitor de recurso no responda más. Es por eso que cuando no estemos del todo seguros de que una librería dinámica no es 100% estable se debe de configurar que esta librería cuente con su propio monitor de recurso, esto con el fin de que no afecte el monitor de recursos de los otros recursos.

Los monitores de recursos simplemente llevan a cabo los comandos del Servicio de Cluster, estos no pueden tomar ninguna decisión por sí mismos. Para determinar si un recurso ha fallado, los monitores de recursos realizan llamados a las librerías dinámicas.

Cada monitor de recursos contiene un "sondeador (poller)" de tareas. Los sondeadores de tareas son el medio por el cual los monitores de recursos determinan el estado (en línea o fuera de línea) de los recursos. Existe un sondeador de tareas por cada 16 recursos.

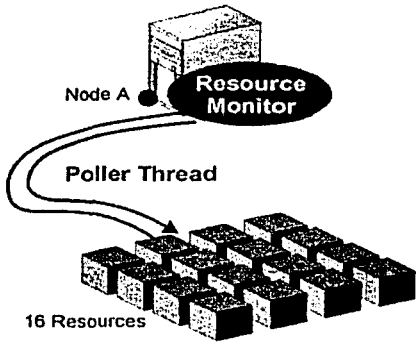


Figura 4.6 - Cómo se determina una falla de un recurso.

Cada recurso tiene dos intervalos de sondeo, los cuales son:

- LooksAlive (Parece vivo).- En este intervalo es usado por el monitor de recursos para determinar si el recurso sigue activo. Las librerías dinámicas realizan un chequeo rápido y superficial para ver si el recurso sigue corriendo.

- IsAlive (Está vivo).- Este intervalo es menos usado por el monitor de recursos y cuando lo llega a usar lo utiliza con el fin de determinar si el recurso se está comportando como se espera.

Si el sondeador de tareas llegase a detectar una falla, el Servicio de Cluster es notificado. Sin embargo el servicio detecta que el sondeo de IsAlive se sigue ejecutando sobre el recurso, por lo que continua monitoreando al recurso y le permite que éste se recobre de la falla, de ser así el Servicio de Cluster es notificado de que el recurso volvió a trabajar otra vez.

El monitor de recursos no intenta reiniciar al recurso que falló al menos que se lo diga el Servicio de Cluster. Ésta es tarea del Administrador de Recursos y Failovers el intentar reiniciar el recurso o realizar un *failover*.

4.4.2.2 Librerías Dinámicas

Cada tipo de recurso en un cluster cuenta con una librería dinámica que implementa las operaciones de administración y monitoreo del recurso. MSCS incluye las siguientes librerías dinámicas para los siguientes tipos de recursos:

- Aplicaciones Genéricas (Generic application)- Clusres.dll
- Servicios Genéricos (Generic service) - Clusres.dll
- IIS virtual root - Iisclus3.dll
- Disco Físico (Physical disk) - Clusres.dll
- Archivo compartido (File share) - Clusres.dll
- Nombre de Red (Network name) - Clusres.dll
- Cola de Impresión (Print spooler) - Clusred.dll
- Dirección de TCP/IP (TCP/IP address) - Clusres.dll
- Servicio de Tiempo (Time service) - Clusres.dll
- Coordinador de Transacciones Distribuidas (Distributed Transaction Coordinator) - Clusres.dll
- Servidor de Colas de Microsoft Message (Microsoft Message Queue Server) - Clusres.dll

4.4.2.3 El Servicio de Tiempo

El servicio de Tiempo (Timeserv.exe) es el responsable para mantener una vista consistente del tiempo en los nodos del cluster. El Administrador de Nodos se encarga de elegir a un nodo como el nodo que es el que lleva la hora correcta dentro del cluster, es decir que todos los nodos del cluster deben de sincronizar sus relojes a la misma hora que el nodo elegido.

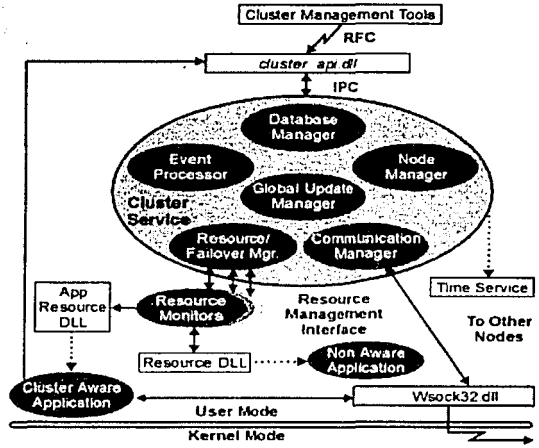


Figura 4.7 - El servicio de Tiempo

Es necesario que todos los nodos tengan las misma hora, esto con el fin de dar a las aplicaciones la habilidad de funcionar correctamente cuando se realiza un *failover*. Esto se realiza usando una especie de "fotografía de la aplicación", o algún otro objeto, el cual es almacenado en archivos. Si no se mantuviera una consistencia en el tiempo y ocurriera un *failover*, podría ocurrir que un archivo o cualquier otro objeto es creado en el futuro.

4.4.3 CÓMO SE COMUNICAN LOS NODOS DE UN CLUSTER

Existen tres métodos que usa MSCS para que se comuniquen los nodos de un cluster, los cuales son:

- Llamas a procedimientos remotos (Remote Procedure Calls RPCs) entre los servicios de MSCS en cada nodo
- Cluster *heartbeats*
- Recurso de *Quórum*

4.4.3.1 RPC ENTRE LOS SERVICIOS DE MSCS DE CADA NODO

Cuando ambos nodos en el cluster se encuentran trabajando, el servicio de MSCS utiliza RPCs para comunicarse entre ellos. La comunicación mediante RPCs se lleva igual que la comunicación de otros servicios de Windows NT, por ejemplo, cuando un grupo o recurso es borrado de un nodo, se utiliza un RPC para comunicar de este cambio al otro nodo del cluster. Cuando uno de los nodos no está corriendo, los nodos existentes utilizan el recurso de quórum para almacenar cualquier cambio con el fin de comunicárselo más tarde a este nodo.

4.4.3.2 CLUSTER HEARTBEATS

Cada nodo en un cluster periódicamente intercambia datagramas con el otro nodo en el cluster con el fin de determinar si ambos nos se encuentran arriba y corriendo correctamente. Este proceso se conoce como envíos de "latidos" o *heartbeats*. Si un nodo no responde a una señal de *heartbeat*, entonces este nodo es declarado como fallido.

El primer nodo del cluster que levanta es el nodo responsable de enviar las señales de *heartbeats*. El primer nodo comienza a mandar las señales cuando es notificado de que el segundo nodo ha levantado. El segundo nodo debe de responder a cada señal de *heartbeat*.

El primer nodo manda señales de *heartbeats* aproximadamente cada 0.5 segundos. El segundo nodo responderá a cada señal de *heartbeat* en aproximadamente 0.2 segundos. Cada uno de los datagramas de *heartbeat* tiene un tamaño de 82 bytes.

La manera en cómo MSCS utiliza a las señales de heartbeat para determinar si un nodo ha sufrido alguna falla depende de qué nodo ha fallado.

Si el segundo nodo en ponerse en línea falla, el primer nodo en ponerse en línea manda 18 señales de *heartbeat* después de la última respuesta obtenida del segundo nodo. Si no existe respuesta alguna a cualquiera de estas señales, entonces se determina que el segundo nodo ha fallado. La manera en cómo las 18 señales de *heartbeat* son mandadas es la siguiente:

- Las primeras 4 señales se envían en intervalos aproximados de 0.7 segundos.

- Las señales 5, 6 y 7 son enviadas dentro de los siguientes 0.75 segundos.
- La señales 8 y 9 se envían en intervalos de aproximadamente 0.3 segundos.
- Las señales 10, 11, 12, 13 y 14 se envían dentro de los siguientes 0.9 segundos.
- Las señales 15 y 16 se envían en intervalos aproximados de 0.3 segundo
- Las señales 17 y 18 se envían 0.3 segundos después.

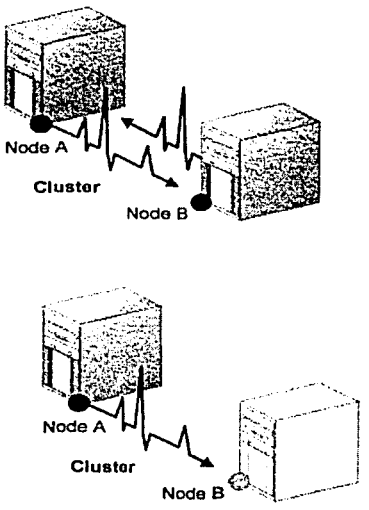


Figura 4.8 - Cluster Heartbeats

Como resultado, al primer nodo del cluster le toma aproximadamente 5.3 segundos el determinar que el segundo nodo ha fallado.

Ahora, si el primer nodo del cluster en poner en línea falla, el segundo nodo en ponerse en línea va a comenzar el proceso definido arriba cuando deje de recibir señales de *heartbeat* dentro de aproximadamente 0.7 segundos.

4.4.3.3 Recurso de Quórum

Como ya se definió con anterioridad, el recurso de quórum es un recurso que almacena los datos de administración del cluster y es accesible para todo los nodos del cluster. *Cluster Server* actualmente sólo soporta como recurso de quórum a aquellos discos *SCSI* que tengan formato NTFS.

Los datos de administración del cluster almacenados en el recurso de *quórum* consiste de un archivo de *log* de los cambios de configuración del cluster, el cual es llamado *quórum log*, el archivo quolog.log es almacenado por default dentro del subdirectorio \Mscs en el recurso de quórum. Por default, el tamaño máximo que puede alcanzar este archivo es de 64 KB.

La justificación de que exista este archivo de log dentro del recurso de quórum es por que es posible que ambos nodos pueden que no se encuentren en línea la mismo tiempo. Por ejemplo, si observamos la siguiente figura, en donde supongamos que el nodo B se encuentra fuera de línea y un administrador realiza un cambio en la configuración del cluster en el nodo A. Si por alguna razón el nodo A se pone fuera de línea y el nodo B se pone en línea, el nodo B no sabría de estos cambios sin el archivo de *log* en el recurso de *quórum*. Aunque, el nodo B levantaría al cluster usando la configuración con la que cuenta actualmente, la cual por lógica no está actualizada. Pero como existe un archivo de *log* en el recurso de *quórum*, el nodo checa este archivo para ver si se han realizado algún cambio antes de poner el cluster en línea.

Cuando todos los nodos han tomado los cambios que están almacenados en el archivo de *log*, los registros que tiene este archivo son borrados del archivo.

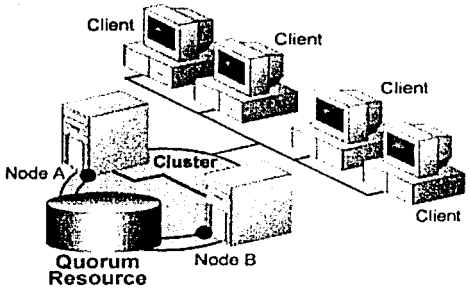


Figura 4.9 - Recurso de Quórum

Para verificar que un nodo tiene acceso al recurso de *quórum* podemos usar al explorador de Windows NT. El nodo que tiene al grupo que contiene al recurso de quórum debe de aparecer como una unidad física de almacenamiento listada en el explorador de Windows NT y el usuario debe de ser capaz de tener acceso a esta unidad. Si la unidad no está lista o no puede ser accedida, se debe de verificar si el cable *SCSI* o la terminación en el bus se encuentran bien.

4.5 MANTENIENDO LA DISPONIBILIDAD DE LOS DATOS

Cuando se planea la implementación de un cluster, siempre hay que examinar cómo es que se va a mantener disponible a la información. Existen generalmente dos pasos en el proceso:

- Realizar una auditoria bastante rigurosa para determinar los puntos comunes de falla.
- Hay que implementar mecanismos que mantengan la disponibilidad de los datos cuando ocurra una falla.

4.5.1 REALIZANDO AUDITORIAS

Una auditoria identifica las posibles fallas en la red y ayuda a determinar si un cluster es apropiado. Específicamente, identifica puntos donde una falla potencial puede prever el acceso a datos que son importantes y valiosos, y también identifica en dónde puede ser utilizando un Servidor de Cluster para así eliminar estos puntos y mantener la disponibilidad de los datos.

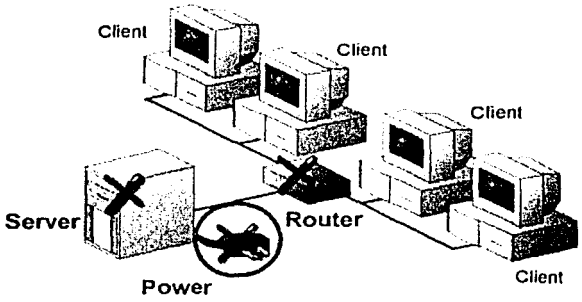


Figura 4.10 - Detectando Puntos de Falla

La siguiente tabla lista algunos de los puntos más comunes de falla. Esta tabla también identifica si un Servidor de Cluster puede o no proveer una solución si no, identifica otra posible solución para la falla.

Punto de Falla	*Solución dada por el Cluster*	*Solución Alterna*
Componente de Red, tal como un *hub*, enrutador, y similares	Ninguna	Componentes de Reserva, rutas redundantes, y similares.
Falla de Poder	Ninguna	Fuente de Poder Ininterrumpible (UPS.- Uninterruptible power supply)
Fallas de hardware en el servidor, tales como CPU, memoria, tarjeta de red, etc.	El proceso de *Failover* toma los recursos, ya sea individualmente o en grupo, y los pone fuera de línea y los vuelve a poner en línea en el otro nodo del cluster. La transición de poner fuera de línea y poner en línea se realiza en un orden preferido, los recursos que dependen de otros son puestos fueras de línea antes y son puestos en línea después de los recursos de los cuales éstos dependen.	Ninguna
Disco no compartido	*Failover*	Ninguna
Disco compartido	Ninguna	Arreglos Redundante de Discos
Falla con conexión al Servidor	*Failover*	Ninguna
Falla en el software del servidor, tal como el sistema operativo, un servicio, o una aplicación.	*Failover*	Ninguna

Hay que notar que un cluster no puede eliminar todos los posible puntos de falla. Un Servidor de Cluster está diseñado para proteger la disponibilidad de los datos pero no puede proteger a los datos como tal. Aunque, a pesar de todo sigue siendo importante el contar con una estrategia de respaldo de información.

4.5.2 Implementando Mecanismos que Mantienen la Disponibilidad de los Datos

Una vez que los puntos potenciales de falla han sido identificados, el siguiente paso es el determinar qué recursos, tales como aplicaciones, *spool* de impresoras, o archivos compartidos, necesitan moverse al cluster con el fin de mantener su disponibilidad.

4.5.2.1 Determinando qué Recursos van a ser Movidos al Cluster

Típicamente, los recursos movidos al cluster son aquellos que proveen acceso a los datos importantes y donde la pérdida de acceso a estos pueden impactar seriamente las operaciones del negocio o empresa. Estos son recursos que requieren de alta disponibilidad tal como un archivo compartido en donde las transacciones diarias son almacenadas y dicho archivo requiere ser accedido continuamente.

Cuando sea determinando qué recursos se van a añadir al cluster, cada nodo en el cluster debe de contar con la capacidad (Memoria, CPU, etc.) para mantener a los recursos del cluster funcionando.

Un Servidor de Cluster incluye algunas librerías dinámicas (DLLs) que permiten a los siguientes recursos ser movidos a un cluster:

- Archivos Compartidos
- Impresoras Compartidas
- Sitios de Red de Microsoft Information Server (IIS)
- Servidores Coordinadores de Transacciones Distribuidas (MS SQL, Oracle, etc.)

- Servidores de Correo de Microsoft
- Servicios Genéricos
- Aplicaciones Genéricas

Un servicio genérico o aplicación genérica es cualquier servicio o aplicación que no están incluidas en las librerías estándar del cluster pero cumplen con los requerimientos de ser usados en un cluster.

Las librerías dinámicas en un Servidor de Cluster únicamente provee de una funcionalidad básica de *failover* para aplicaciones y servicios. Aunque cualquier casa de software puede, sin embargo, escribir nuevas librerías para sus propias aplicaciones o servicios para tomar mejores ventajas de las características del Cluster.

Para determinar qué archivos compartidos, impresoras compartidas, Sitios de Red de ISS, y servicios van a moverse a un cluster puede ser determinado basándose en los requerimientos de disponibilidad. Sin embargo, cuando se decide qué aplicaciones mover a un cluster, existen consideraciones adicionales.

Para una aplicación que va a ser usada en un cluster, deben de tomarse en cuenta los siguientes criterios:

- Contar como protocolo de comunicación a TCP/IP.

Todas la aplicaciones de red usadas con *Cluster Server* deben usar como protocolo de comunicación TCP/IP (Transmission Control Protocol/Internet Protocol). Microsoft Cluster Server no puede realizar un *failover* sobre aplicaciones que no soportan TCP/IP.

- Capacidad de Almacenamiento Remoto

Todas las aplicaciones que almacenan datos deben de tener la habilidad de ser configuradas para almacenar sus datos sobre una unidad de almacenamiento compartido, es decir que dicha unidad se encuentre sobre un mismo bus *SCSI* (Small Computer System Interface). Cualquier aplicación que no sea capaz de

almacenar sus datos sobre una unidad de almacenamiento compartido no puede ser puesta en cluster, debido a que no va poderse realizar un *failover* sobre ésta aplicación, porque la información después del proceso de *failover* no se va a encontrar disponible.

4.5.3 Modificando los Procedimientos de Operación del Servidor

Debido a un cluster consiste en dos servidores (por el momento; posteriormente un cluster podrá formarse hasta 16 nodos), administrar y realizar procedimientos de operación pueden ser menos quebrantador para los usuarios.

Por ejemplo, cuando una tarea administrativa requiere que un servidor se encuentre no disponible hacia los usuarios, dado que un cluster cuenta con un segundo servidor éste podrá continuar otorgando el servicio a los usuarios. Esto nos da como ventaja, el que ya no tengamos que esperar a que pasen las horas picos para poder realizar:

- Reinicializar el servidor.
- Instalar nuevo hardware
- Instalar nuevo software
- Realizar mantenimientos preventivos a uno de los nodos del cluster.

Típicamente, un administrador tiene que seguir un conjunto de procedimientos para poder realizar cierta tareas administrativas, tales como reinicializar un servidor, realizar un respaldo, o instalar nuevo hardware o software. Ahora con el cluster, estas tareas tienen que ser reexaminadas, debido a que muchos de estos procedimientos tienen que ser modificados cuando el cluster sea instalado en la red.

4.5.3.1 Usando el Administrador de Cluster

Antes de realizar cualquier tarea administrativa, se debe utilizar la aplicación "Cluster Administrator" para mover todos los grupos que tiene el nodo que se va a poner fuera de línea al otro nodo del cluster. Esto con el fin de minimizar el tiempo de impacto hacia los usuarios, en el sentido en que el cluster no va a perder nada de tiempo en detectar que un nodo está fuera de línea y por lo tanto comience el proceso de *failover*.

Realizando Respaldo de Información

Cuando se instala un cluster, también deben de ser cambiadas las políticas de respaldo. Cuando se está utilizando MS Cluster Server, se debe de respaldar los siguiente:

- El sistema operativo del Nodo A.
- El sistema operativo del Nodo B.
- Los datos que se encuentran en la unidad de almacenamiento compartida.

Para poder realizar el respaldo de los datos que se encuentran en la unidad de almacenamiento compartida, se debe de realizar una conexión de red hacia el nombre del cluster. El nombre de red que se va usar debe de ser miembro del mismo grupo de cluster que contiene el disco compartido de tal manera que si llegase a ocurrir un *failover*, tanto el nombre de red como el disco compartido seguirán disponibles sobre el mismo nodo virtual.

Por ejemplo, los discos *SCSI* compartidos cuentan con un "share" escondido con propósitos de administración. Por lo que, si en un cluster un disco compartido tiene como letra asignada la "W", es posible acceder a este disco de la siguiente manera \\Nombre_De_Red_Del_Cluster\w$.

Realizando esta conexión, el administrador va a ser capaz de realizar el respaldo de los datos sin importar qué nodo tiene el control del disco compartido, siempre y cuando tanto el nombre de la red como el disco sean miembros del mismo grupo.

4.6 ELIGIENDO UN MODELO DE CLUSTER

Existen 5 configuraciones diferentes de implementar un cluster:

- Modelo A: Solución de Alta-Disponibilidad con Carga de Trabajo Balanceada.
- Modelo B: Solución de "*Hot Spare*" con Máxima Disponibilidad.
- Modelo C: Solución Parcial de Cluster.
- Modelo D: Solución de Servidor Virtual (No hay *Failover*)
- Modelo E: Solución Híbrida

4.6.1 MODELO A: SOLUCIÓN DE ALTA-DISPONIBILIDAD CON CARGA DE TRABAJO BALANCEADA

En un cluster de Modelo A, cada nodo en el cluster tiene recursos del cluster que ofrecen a los usuarios. Esto provee un alto desempeño dado que se balancean las cargas de trabajo entre los dos nodos del cluster. Sin embargo, en este modelo, cada nodo debe de ser capaz de poner sus propios recursos en línea con sus propias capacidades y también debe de ser capaz de poder con la carga de trabajo del otro nodo para el caso en que ocurra un *failover*. Dependiendo de la eficiencia y capacidad de los nodos del cluster, el desempeño de las operaciones pueden decaer si por alguna razón se realiza un *failover* pues esto quiere decir que un solo nodo tiene todos los recursos en su poder.

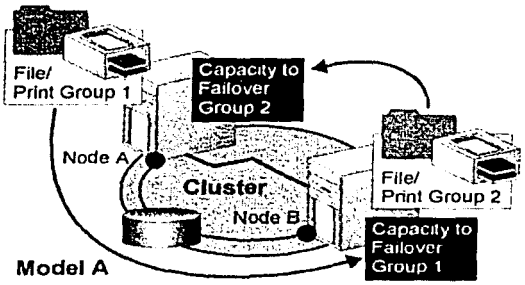


Figura 4.11 - Cluster Modelo A.

Cuando se utiliza un modelo como el que se muestra en la figura de arriba, Se utilizarán los siguientes nombres de red y direcciones IP:

- Un nombre y una dirección IP para el nodo A.
- Un nombre y una dirección IP para el nodo B.
- Un nombre de red y una dirección IP para el cluster.
- Un nombre de red y una dirección IP para el grupo 1 de *file/print*.
- Un nombre de red y una dirección IP para el grupo 2 de *file/print*.

Este modelo es el que más se recomienda cuando lo que el cluster va a contener principalmente son archivos y colas de impresión que se desean compartir.

Como se puede observar, se pueden crear dos grupos (uno por nodo), de tal manera, que si un nodo falla, el otro nodo de manera temporal tomará el control de ambos grupos. Cuando el nodo fallido vuelva a estar en línea, el grupo que originalmente le pertenecía regresará a éste, y entonces el desempeño regresará a sus niveles normales. Cabe mencionar que este modelo no otorga una alta disponibilidad de recursos y un gran desempeño pues las cargas de trabajo están balanceadas.

Por último se recomienda que cuando se implemente este modelo, se configuren a los grupos con un "servidor de preferencia" (*preferred server*). Esta opción proveerá al cluster con la capacidad de poder realizar un *failback*.

4.6.2 Modelo B: Solución de "Hot Spare" con Máxima Disponibilidad

En un cluster de Modelo B, contaremos con máxima disponibilidad y eficiencia en el manejo de los recursos, pero con la limitante de que ambos nodos nunca son completamente utilizados. Un nodo del cluster tiene todos los recursos en su poder y los pone disponibles a los usuarios, mientras que el otro nodo se encuentra disponible y en espera para el caso en que llegase a presentarse un *failover*.

En este modelo, el nodo que se encuentra disponible es conocido como "*Hot Spare*", el cual siempre está listo para ser usado si un *failover* llegase a ocurrir. Si el nodo que cuenta con todos los recurso falla, el nodo "*hot spare*" va a realizar un *failover* y entonces éste continuara poniendo los recursos del cluster en línea con la misma eficiencia en que lo estaba realizando el otro nodo (siempre y cuando ambos nodos cuenten con la misma configuración de hardware CPU, Memoria, etc.).

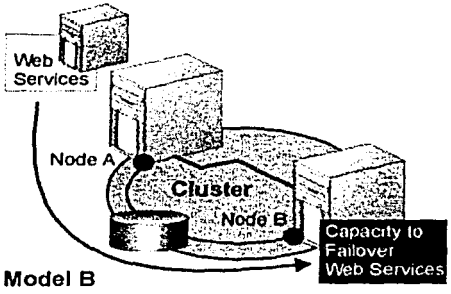


Figura 4.12 - Cluster Modelo B.

Cuando se utiliza un modelo como el que se muestra en la figura de arriba, Se utilizarán los siguientes nombres de red y direcciones IP:

- Un nombre y una dirección IP para el nodo A.
- Un nombre y una dirección IP para el nodo B.
- Un nombre de red y una dirección IP para el cluster.
- Un nombre de red y una dirección IP para el grupo de servicios de Web.

Este modelo es el que más se recomienda cuando lo que el cluster va a contener las aplicaciones y/o recursos críticos de la empresa. Por ejemplo, una organización que cuenta con un servidor de páginas de Web donde los clientes ponen órdenes de compra, pueden optar por tomar un modelo como éste. El costo del servidor que se encuentra en estado de espera, puede ser justificado, garantizando que se va a contar con un acceso contínuo a las páginas de Web, sin ningún degradamiento en el desempeño.

Con un modelo como éste, podemos decir que la empresa contará con una muy alta disponibilidad de recursos.

La política de *failback* sugerida depende de la capacidad de los nodos. Si el nodo de "*hot spare*" cuenta con las misma especificaciones de hardware y puede proveer el mismo desempeño que el nodo primario, entonces no se recomienda implementar ninguna política de *failback*. Mas si el nodo primario otorga mejor desempeño, entonces sí se debe de implementar una política de *failback* sobre el grupo, de tal manera que siempre dicho grupo se encuentre en el nodo que otorga mejor desempeño.

4.6.3 MODELO C: SOLUCIÓN PARCIAL DE CLUSTER

Un cluster de Modelo C es muy similar a un cluster de Modelo B. Pero en este modelo, lo que se va a ejecutar adicionalmente en el nodo primario son aplicaciones "*non-cluster aware*".

Cuando se implementa un cluster como éste, van a existir aplicaciones que no van a ser configuradas para que sean parte del cluster como tal. Son aplicaciones que no van a poder realizar un *failover*, esto puede ser debido a que:

- Estas aplicaciones no cumplen con los requerimientos de *failover* (que utilicen TCP/IP como protocolo de comunicación y que sean capaces de almacenar sus datos en un disco *SCSI* compartido).
- No requiere de alta disponibilidad.

Estas aplicaciones pueden ser instaladas y usadas sobre cualquier nodo del cluster. Sin embargo, si el nodo que las ejecuta falla, estas aplicaciones se van a volver inaccesibles.

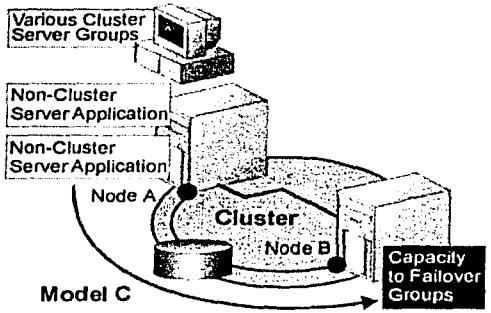


Figura 4.13 - Cluster Modelo C.

En el diagrama de arriba, el nodo A tiene dos aplicaciones que son "*non-cluster aware*" que no van a realzar un *failover* si este nodo falla. Aunque, si el nodo A falla, estas aplicaciones se pondrán inaccesibles hasta que este nodo vuelva a funcionar otra vez.

Cuando se utiliza un modelo como el de arriba, se utilizarán los siguientes nombres de red y direcciones IP:

- Un nombre y una dirección IP para el nodo A.
- Un nombre y una dirección IP para el nodo B.
- Un nombre de red y una dirección IP para el cluster.
- Un nombre de red y una dirección IP para cada grupo del cluster.

Este modelo es muy útil cuando existen tanto aplicaciones "*non-cluster aware*" y aplicaciones "*cluster aware*". Por ejemplo, en vez de comprar dos nuevos servidores para crear un cluster para una aplicación que es crítica para la empresa, un servidor puede ser agregado para crear un cluster con el servidor existente. En este caso el servidor original va a continuar ejecutando las aplicaciones que ya tenía y el nuevo servidor va a ser usado para las aplicaciones criticas que requieran del cluster.

Con un modelo como éste, podemos decir que la empresa contará con una alta disponibilidad de recursos que se encuentran configurados dentro del cluster y una disponibilidad normal para aquellas aplicaciones que son "*non-cluster aware*".

La política de *failback* puede o no ser implantada, ésta va a depender de la capacidad de los servidores, es decir si uno es más "grande" que el otro, pues esta política si debe de ser implementada, mas si ambos servidores cuentan con la misma capacidad esta política puede o no ser habilitada.

4.6.4 MODELO D: SOLUCIÓN DE SERVIDOR VIRTUAL (NO HAY FAILOVER)

Un cluster de Modelo D utiliza el concepto de "servidor virtual". En este modelo no existe un cluster como tal, debido a que sólo existe un solo servidor. Debido a esto, no pueden utilizarse las capacidades de *failover* por lo que MS Cluster Server no puede ser utilizado. En su lugar, este modelo puede ser usado para agrupar los recursos de la organización o con propósitos administrativos, de tal manera que a los usuarios les sea mucho mas fácil localizar los recursos.

Una ventaja de este modelo es que en el futuro, cuando se deseen alcanzar niveles más altos de disponibilidad, otro nodo puede ser agregado para crear un cluster. Debido a que los grupos de recursos ya se encuentran creados, únicamente las políticas de *failover* necesitan ser configuradas.

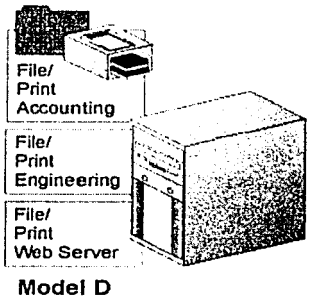


Figura 4.14 - Cluster Modelo D.

Cuando se utiliza un modelo como el de arriba, se utilizarán los siguientes nombres de red y direcciones IP:

- Un nombre y una dirección IP para el nodo A.
- Un nombre de red y una dirección IP para el cluster.
- Un nombre de red y una dirección IP para el grupo de *accounting*.
- Un nombre de red y una dirección IP para el grupo de *engineering*.
- Un nombre de red y una dirección IP para el grupo de *web server*.

Este modelo puede ser usado en una organización que tiene tanto sus archivos como colas de impresión configurados sobre servidores virtuales, en donde cada departamento en la organización cuenta con sus propios grupos. Es decir, que cuando los usuarios de un departamento requieran acceder ya sea a un archivo o cola de impresión compartida, existe un solo servidor virtual al que estos usuarios deberán de acceder.

Con un modelo como éste podemos decir que la empresa no contará con ningún dispositivo de alta disponibilidad adicional. La política de *failback* no puede ser implementada debido a que sólo existe un solo nodo.

4.6.5 Modelo E: Solución Híbrida

El último de los modelos del Cluster, el Modelo E, es un híbrido de los modelos anteriores, el cual combina las ventajas de los otros cuatro modelos dentro de un solo cluster. Siempre y cuando se cuente con suficiente capacidad, se pueden implantar múltiples escenarios de *failover*, los cuales van a coexistir dentro de un mismo cluster.

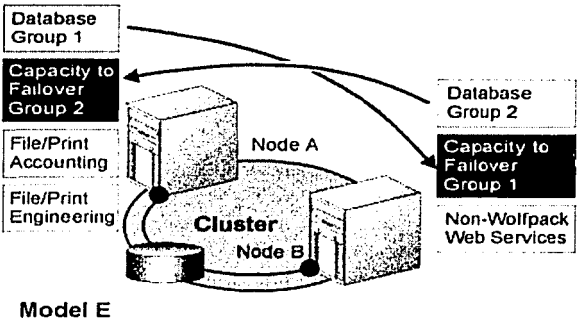


Figura 4.15 - Cluster Modelo E.

El ejemplo de arriba nos muestra un cluster con un balanceo de cargas de dos bases de datos compartidas. Aunque el desempeño se va a ver un poco afectado cuando se presente el caso en que ambos grupos residan en un solo nodo.

Los dos grupos de *file & print sharing* en el nodo A no requieren de contar con la habilidad de realizar un *failover*, aun que existen en grupos lógicos o en servidores virtuales, éstos existen por propósitos administrativos. Por otro lado, también existen aplicaciones "*non-cluster aware*" en el nodo B, estas aplicaciones operan de modo normal sin ninguna protección de *failover*.

Cuando se utiliza un modelo como el aquí ilustrado, se utilizarán los siguientes nombres de red y direcciones IP:

- Un nombre y una dirección IP para el nodo A.
- Un nombre y una dirección IP para el nodo B.
- Un nombre de red y una dirección IP para el cluster.
- Un nombre de red y una dirección IP para cada grupo del cluster.

Este modelo es recomendado para aquellas organizaciones que requieren correr aplicaciones "*non-cluster aware*" sobre los nodos del cluster.

Con un modelo como éste podemos decir que la empresa contará con un nivel de disponibilidad que va de alto a muy alto sobre los recursos configurados para el *failover*.

La política de *failback* va a ser variable, ésta va a depender de las necesidades de cada grupo.

4.7 INSTALANDO MICROSOFT CLUSTER SERVER

Los requerimientos de instalación de Microsoft Cluster Server pueden ser agrupados dentro de las dos siguientes categorías:

- Requerimientos de Software
- Requerimientos de Hardware
- *Member servers*

  Cada nodo es configurado como "*member server*" corriendo Windows NT; cada nodo debe de ser un miembro del mismo dominio.

- BDC y BDC

  Cada nodo es un *backup domain controller* (BDC) de un mismo dominio.

- PDC y BDC

  Ambos nodos están configurados dentro de un mismo dominio, con un nodo que actúa como *primary domain controller* (PDC), y el otro actúa como BDC. Esta configuración requiere de relaciones de confianza con cualquier dominio existente con el fin de que los usuarios sean capaces de ganar acceso al cluster.

4.7.1 CUENTA DE SERVICIO DE MICROSOFT CLUSTER SERVER

El servicio de cluster requiere de una cuenta de usuario bajo la cual el servicio del cluster puede correr cuando el servicio del cluster comience. Esta cuenta de usuario debe ser creada antes de comenzar la instalación del cluster

debido a que en el proceso de instalación se requiere saber cuál es el "*username*" y un "*password*" de la cuenta de servicio del cluster. El proceso de instalación no continuará si no hasta que se haya introducido una cuenta válida en el dominio.

Esta cuenta requiere tener privilegios de administrador y el privilegio especial de "*logon as a service*". Aparte la cuenta debe de crearse deshabilitando la cajas de "*User must change password at next logon*" y "*Password never expires*".

4.7.2 Requerimientos de Hardware

En cuanto a los requerimientos de hardware Microsoft Cluster Server debe de cumplir con todos los requerimientos de hardware que pide Windows NT Server Enterprise Edition, y además con los siguientes requerimientos:

- Dos computadoras, cada una con las siguientes características:
  - Que tengan un bus ISA (Industry Standard Architecture)y un bus PCI (Peripheral Component Interconnect)..
  - Un disco de "boot" con Windows NT Server Enterprise Edition instalado. Este disco no debe de estar instalado en ningún bus *SCSI* compartido.
  - Un adaptador *PCI SCSI*
  - Como mínimo una tarjeta de red. Es recomendado que se tengan como mínimo dos tarjetas de red de tal manera que los nodos del cluster puedan tener una interconexión privada.

- Uno o varios discos *SCSI* externos para conectar ambas computadoras, los cuales van a ser usados como los discos compartidos *SCSI*.

- Los cables y terminadores *SCSI* necesarios para unir los discos a ambas computadoras y terminar apropiadamente el bus.

Es recomendado que todo el hardware sea similar para ambos servidores, esto con el fin de hacer mas fácil la configuración y eliminar los posibles problemas de compatibilidad.

Antes de realizar cualquier elección de hardware, hay que verificar que el hardware que se va a utilizar se encuentra en la Lista de Compatibilidad de

Hardware de Microsoft Cluster Server. La última versión de esta lista de compatibilidad se puede encontrar en la siguiente dirección de internet http://www.microsoft.com/ introduciendo la siguiente búsqueda *"Cluster Server Hardware Compatibility List"*.

4.7.3 CONFIGURANDO EL BUS SCSI COMPARTIDO

El bus *SCSI* que es listado en los requerimientos de hardware debe de ser configurado antes de instalar el cluster. El configurar este bus implica:

- Configurar los dispositivos *SCSI*.

  Los controladores *SCSI* y discos *SCSI* deben de configurarse de tal manera que éstos puedan trabajar en un bus *SCSI* compartido.

- Terminar correctamente al bus.

  El bus *SCSI* compartido debe de tener un terminador en cada uno de los extremos. Es posible tener más de un bus *SCSI* compartido entre los nodos de un cluster, pero cada uno de éstos deben de estar correctamente terminados.

4.7.3.1 CONFIGURANDO LOS DISPOSITIVOS SCSI

Cada dispositivo sobre el bus *SCSI* compartido debe de tener un *SCSI ID* único (tanto los controladores *SCSI* como los discos *SCSI*). La gran mayoría de los controladores *SCSI* vienen configurados de fábrica con el *SCSI ID* 7, parte de la configuración de bus *SCSI* compartido es el cambiar el *SCSI ID* de uno de los controladores *SCSI* a otro *SCSI ID* diferente, como por ejemplo el *SCSI ID* 6. Si existe más de un disco que se va a encontrar dentro del bus *SCSI* compartido, cada disco debe de tener también un *SCSI ID* único.

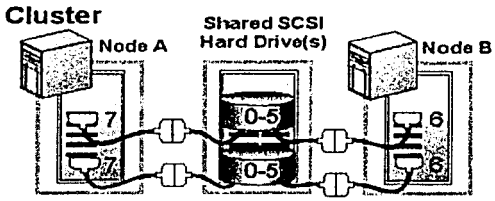


Figura 4.16 - Configurando los Dispositivos *SCSI*.

Algunas controladoras *SCSI* inicializan (dan "reset") al bus *SCSI* cuando el equipo está inicializando. Si esto ocurre, el inicializar el bus *SCSI* puede interrumpir cualquier transferencia de datos entre el otro nodo y los discos que están en el bus *SCSI* compartido. Por lo que, esta característica debe de ser deshabilitada de ser posible.

4.7.3.2 Terminando el Bus SCSI Compartido

Existen diferentes métodos que pueden ser usados para terminar el bus *SCSI* compartido, los cuales son:

- Controladoras *SCSI*
  Las controladoras *SCSI* cuentan con terminación interna que puede ser usada para terminar el bus, sin embargo este método no es recomendado por Microsoft Cluster Server; porque si uno de los nodos está fuera de línea o es removido del bus, con esta configuración, el bus *SCSI* no va a estar propiamente terminado y no operará correctamente.

- Unidades Externas de Almacenamiento
  Algunas unidades de almacenamiento también cuentan con terminación interna, la cual puede ser usada para terminar el bus *SCSI*, siempre y cuando esta unidad se encuentra al final del bus *SCSI* compartido.

- Cables Y

Los cables Y pueden se conectados a cualquier dispositivo, siempre y cuando este dispositivo se encuentre al final del bus *SCSI* compartido. Un terminador *SCSI* debe de estar conectado a un extremo del cable Y, esto con el fin de terminar el bus *SCSI*. Este método de terminación requiere deshabilitar o remover cualquier terminación interna que los dispositivos puedan tener.

- Conectores *Trilink*
  Los conectores *trilink* pueden ser conectados a ciertos dispositivos. Si un dispositivo está al final del bus, un conector *trilink* puede ser usado para terminar el bus. Este método de terminación requiere deshabilitar o remover cualquier terminación interna que los dispositivos puedan tener.

Hay que notar que cualquier dispositivo *SCSI* que no se encuentra al final del bus *SCSI* compartido debe de tener su terminación interna deshabilitada.

De los métodos de terminación anteriormente mencionados, los más recomendados son los dos últimos (el utilizar cables Y o conectores *trilink*), debido a que estos métodos mantienen la terminación sobre el bus sin importar si está en línea o no.

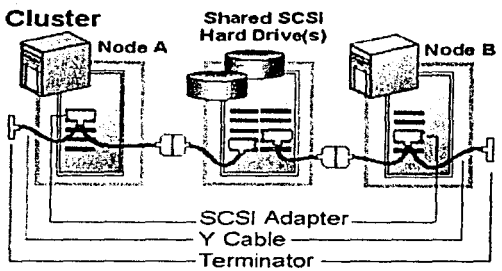


Figura 4.17 - Terminando el Bus *SCSI* Compartido.

4.7.3.3 ASIGNANDO DRIVE LETTER A LOS DISCOS DE LA UNIDAD DE ALMACENAMIENTO COMPARTIDO

Una vez que el bus *SCSI* compartido ha sido configurado, las *drive letter* deben de ser asignadas a los discos que se encuentras en el bus *SCSI* compartido.

Asignar *drive letter* es necesarios porque los discos que se encuentran sobre el bus *SCSI* compartido deben de tener la misma *drive letter* en ambos nodos del cluster. Por lo que, después de configurar el bus *SCSI* compartido y antes de instalar el software de cluster, las *drive letters* deben de ser configuradas usando la utilería "*Disk Administrator*".

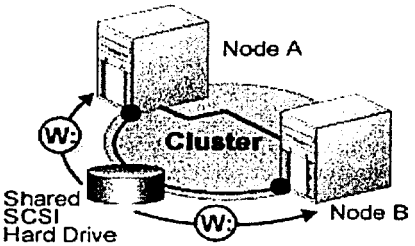


Figura 4.18 - Asignando *Drive letter* a los Discos de la Unidad de Almacenamiento Compartido.

Hay que hacer notar que una vez que el bus *SCSI* compartido ha sido configurado y conectado a ambos nodos del cluster, no se debe de correr Windows NT Server Enterprise Edition en ambos nodos a la vez hasta que *Cluster Server* haya sido instalado en al menos un nodo.

Para asignar las *drive letters* en ambos nodos, hay que realizar los siguientes pasos:

1. Iniciar Windows NT Server Enterprise Edition en un nodo.

2. Encender el segundo nodo, pero **no hay que permitir** que cargue el sistema operativo.

3. Ejecutar la utilería de "*Disk Administrator*", y después asignar las *drive letter* a los discos que están sobre el bus *SCSI* compartido; es recomendado por cuestiones de administración el comenzar a asignar las letras a partir de la letra "Y" hacia la letra "A", y asignar a la unidad de CD-ROM la letra "Z".
4. Dar de baja el primer nodo, pero no hay que apagarlo.

5. Iniciar Windows NT Server Enterprise Edition en el segundo nodo.

6. Ejecutar la utilería de "*Disk Administrator*", y asignar las *drive letters* a los discos que se encuentran en el bus *SCSI* compartido, usando las mismas letras que se utilizaron en el primer nodo.

4.7.4 INSTALANDO EL SOFTWARE DE MICROSOFT CLUSTER SERVER

Antes de instalar el software de Microsoft Cluster Server, se debe de contar con la siguiente información:

- Los permisos apropiados para instalar *Cluster Server*

  Para poder ejecutar el programa de instalación de *Cluster Server* se requiere estar validado con una cuenta con privilegios de Administración.

- Una cuenta de usuario para los servicios de *Cluster Server*

  El nombre de la cuenta de usuario y la contraseña (password), de la cuenta de usuario bajo la cual los servicios del cluster van a correr.

- El nombre del directorio donde se va a instalar *Cluster Server*

  El directorio por defecto es %windir%\Cluster.

- El nombre del cluster

  Este debe de ser un NetBIOS único. El segundo nodo usa este nombre para conectarse al primer nodo para obtener la información de configuración. Los clientes pueden ser configurados para que usen este nombre para ganar acceso a algunos recursos de cluster. Este nombre también puede ser usado en la herramienta de "*Cluster Administrator*" para conectarse y así puedan administrar el cluster.

- Una dirección de *internet protocol* (IP) y submáscara para el cluster

Cluster Server utiliza el protocolo TCP/IP (Transmission Control Protocol/Internet Protocol) para comunicarse sobre la red y requiere de una dirección de IP estática. No es posible configurar un cluster para que obtenga una dirección de IP desde un servidor de DHCP (Dynamic Host Configuration Protocol).

El proceso de instalación de Microsoft Cluster Server es un proceso de dos fases, las cuales son:

- Instalar *Cluster Server* en el primer nodo.

  La información de configuración inicial del cluster debe de ser introducida, y después de eso el cluster puede ser creado.

- Instalar *Cluster Server* en el segundo nodo.

  La información de configuración es obtenida del primer nodo del cluster.

4.7.4.1 Instalando Cluster Server en el Primer Nodo

En la primera fase de instalación, toda la información de configuración del cluster debe de ser introducida, con el fin de poder crear al cluster. Esta información puede introducirse a través del asistente de instalación de *Cluster Server*. Este asistente provee una guía de instalación, que cuenta con los siguientes pasos:

1. Confirma qué *Cluster Server* está siendo instalado sobre un hardware soportado.
2. Determinar si se desea crear un nuevo cluster o unirse a uno ya existente, o instalar la herramienta de administración "Cluster Administrator".
3. Especificar el subdirectorio y la cuenta de servicio.
4. Configurar cuál disco va a ser usado como disco de quórum.
5. Configurar las tarjetas de red que van a ser usadas en el cluster.
6. Configurar la prioridad de uso de las tarjetas para la comunicación entre nodos.
7. Copiar los archivos de *Cluster Server* e iniciar el servicio de *Cluster Server*.

Confirmando si el Hardware es Soportado

Las primeras dos pantallas que son mostradas por el asistente son la pantalla de Bienvenida y la pantalla de Hardware Soportado. La pantalla de Bienvenida no requiere que se le introduzca ninguna información.

Después de la pantalla de Bienvenida viene la pantalla de Hardware Soportado. La pantalla de Hardware Soportado advierte a los usuarios que *Cluster Server* esta únicamente soportado cuando se instala sobre configuraciones de hardware que ya han sido instaladas y se encuentran dentro de la lista de compatibilidad de hardware de *Cluster Server*. Para continuar con el proceso de instalación, hay que dar click sobre "I Agree" para aceptar la condición de que *Cluster Server* únicamente está soportado sobre hardware probado. Dar click sobre "I Agree" habilita el botón de "Next".

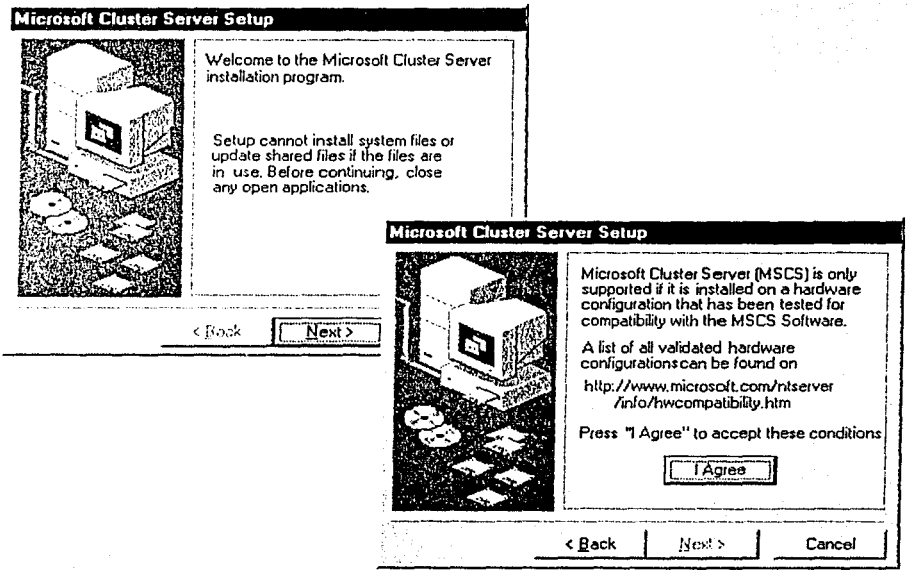


Figura 4.19 - Confirmando si el Hardware es Soportado.

Determinando el Tipo de Instalación

El siguiente paso del proceso de instalación es el especificar el tipo de instalación a realizar:

- "Form a new cluster"

  Hay que seleccionar esta opción cuando se está instalando el primer nodo de un cluster y el primer nodo es el que va a crear el cluster. Cuando esta opción es seleccionada, un nombre de red (NetBIOS) debe de ser introducido.

- "Join an existing cluster"

  Hay que seleccionar esta opción cuando se está instalando el segundo nodo del cluster. Cuando esta opción es elegida, el nombre del cluster debe de ser introducido.

- "Install Cluster Administrator"

  Hay que seleccionar esta opción cuando se está instalando sobre una computadora que va a ser un nodo de un cluster, pero va a ser utilizada para administrar al cluster.

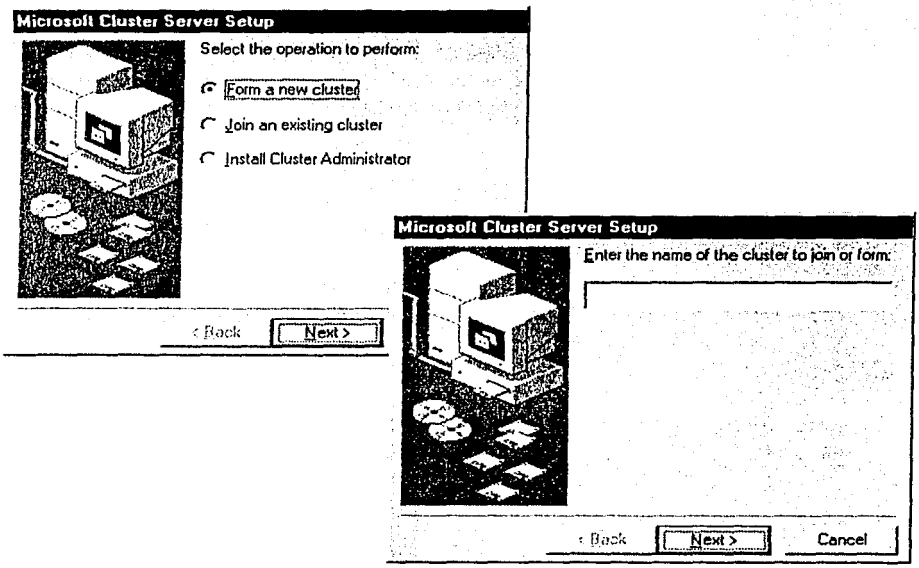


Figura 4.20 - Determinando el Tipo de Instalación.

Hay que notar que cuando se ejecuta el programa de instalación de *Cluster Server* sobre una computadora que no tiene Windows NT Server Enterprise Edition instalado, las pantallas mostrada arriba no van a ser mostradas. En su lugar, la pantalla que aparece indica que Windows NT Server Enterprise Edition no se encuentra instalado y únicamente la herramienta "Cluster Administrator" va a ser instalada.

Especificando el Subdirectorio de Instalación y la Cuenta de Servicio

En esta parte de la instalación se nos pide el subdirectorio en donde se instalarán los archivos de *Cluster Server*, y el nombre de la cuenta y la contraseña, y el nombre del dominio para los servicios del cluster.

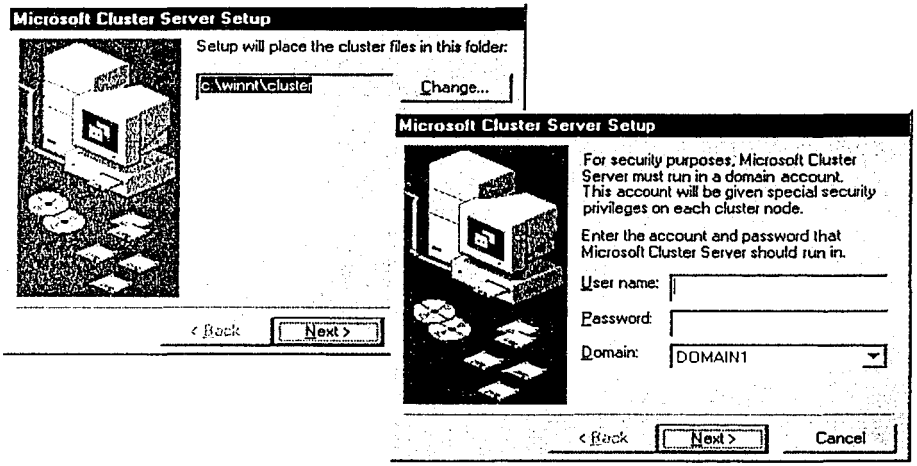


Figura 4.21 - Especificando el Subdirectorio de Instalación y la Cuenta de Servicio.

En la primer pantalla de arriba se nos pregunta la ruta en donde se van a colocar los archivos de *Cluster Server*. Esta pantalla trae como opción por defecto %windir%\Cluster.

La siguiente pantalla nos pregunta el nombre de usuario, la contraseña y el nombre del dominio para la cuenta que se va usar para ejecutar los servicios del cluster. Una vez que ya se introdujo esta información, hay que dar click sobre el botón "Next", una vez hecho esto el asistente de instalación valida si la cuenta existe y cuenta con los suficientes privilegios. Es importante, que la cuenta ya exista para este punto del proceso, porque en caso de no existir, no se podrá continuar con el proceso de instalación.

Configuración de los Discos

En esta parte de la instalación se especifica qué discos pertenecen al bus *SCSI* compartido y también se designa qué disco del bus *SCSI* compartido va a actuar como disco de quórum.

En la primera pantalla de la figura siguiente es usada para especificar qué discos pertenecen al bus *SCSI* compartido, los cuales van a ser usados por el cluster. Por defecto, todos los discos que se encuentren sobre cualquier bus *SCSI* que no sea el bus *SCSI* del disco del sistema operativo van a aparecer en la lista de "*Shared cluster disk*". Aunque, si el nodo tiene múltiples buses *SCSI*, algunos discos pueden ser listados y no está en un bus *SCSI* compartido. Estos discos deben de ser removidos de esta lista.

La siguiente pantalla que aparece es usada para seleccionar el disco sobre el bus *SCSI* compartido que va a ser usado como disco de quórum. El disco de quórum puede ser cualquier disco sobre el bus *SCSI* compartido y éste puede ser cambiado por el administrador del cluster utilizando la herramienta de *cluster administrator*.

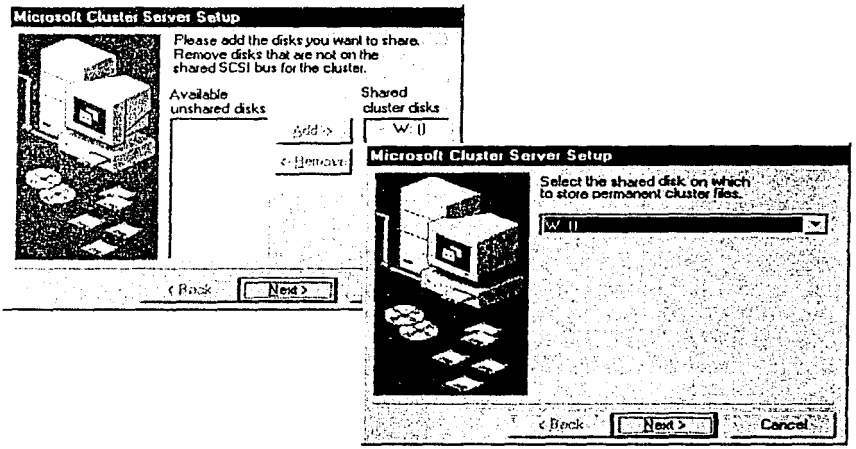


Figura 4.22 - Seleccionando Disco del Bus *SCSI* Compartido y eligiendo el disco de Quórum.

Instalación de las Tarjetas de Red.

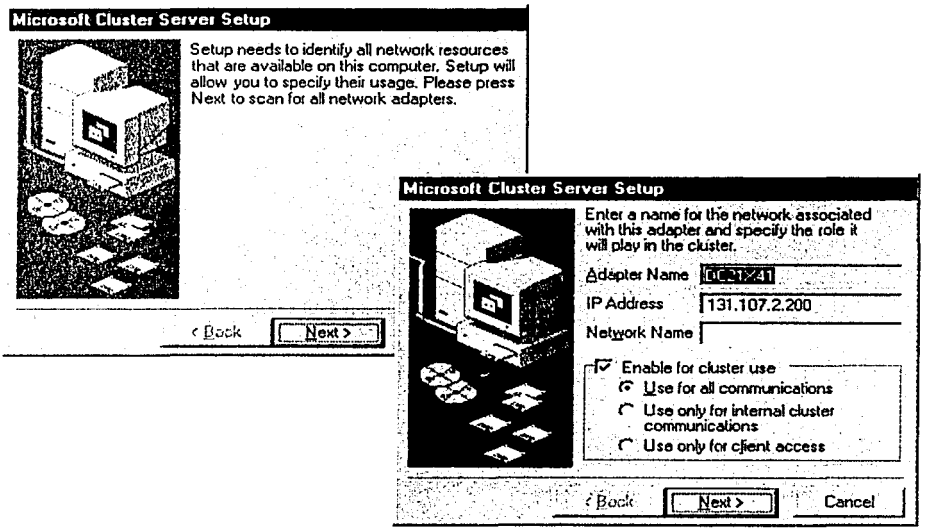
La siguiente fase del asistente de instalación es la configuración de los adaptadores de red en el nodo.

En la primera pantalla de la figura 4.23 se nos informa que el asistente comenzará a buscar los adaptadores de red en el nodo. No se requiere introducir ningún dato en esta pantalla, lo único que hay que hacer es apretar el botón de **Next** para comenzar el proceso de búsqueda.

En la siguiente pantalla nos muestra el primer adaptador de red detectado y nos permite configurar los siguientes parámetros:

- **Network Description (Descripción del Adaptador de Red)**

Ésta es una descripción que va a ser usada por el Administrador del Cluster para identificar a qué red está conectado el adaptador de red. Por ejemplo, si el adaptador de red está conectado a una red privada que va a ser usada por lo nodos para comunicarse, entonces podemos llamara a estar tarjeta **ClusterNet** o Red Privada o **Interna**.



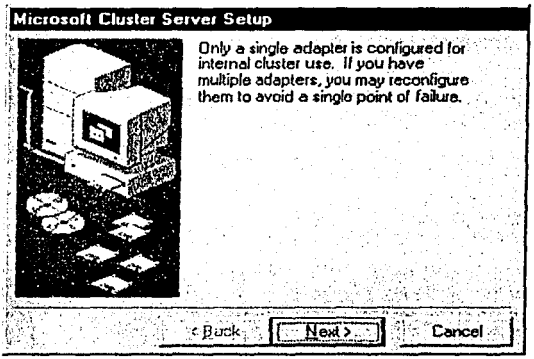


Figura 4.23 - Instalación de los Adaptadores de Red.

- ***Enable for cluster use***

  Si esta caja está seleccionada, entonces estaremos dando permiso a los servicios de Cluster a usar este adaptador de red. Esta opción está seleccionada por defecto.

- ***Use for all communications*** **(Úsese para todas la comunicaciones)**

  Si se selecciona esta opción, entonces le estaremos diciendo a los servicios del Cluster que use este adaptador para la comunicación entre los nodos y para la comunicación hacia los clientes. Esta opción está seleccionada por defecto.

- ***Use only for internal cluster communications*** **(Úsese para la comunicación interna del cluster)**

  Si seleccionamos esta opción, entonces los servicios del cluster van a usar a este adaptador para que únicamente los nodos de comuniquen entre ellos.

- ***Use only for client access*** **(Úsese únicamente para la comunicación hacia los clientes)**

Si se selecciona esta opción, entonces los servicios del cluster utilizarán a este adaptador de red para atender a los clientes. No va a existir comunicación entre los nodos a través de este adaptador.

Esta pantalla será desplegada una vez por cada adaptador de red en el nodo. Por ejemplo, si un servidor cuenta con tres adaptadores de red, esta pantalla aparecerá tres veces durante la instalación del cluster, es decir una vez por adaptador de red.

Si un nodo únicamente cuenta con un solo adaptador una pantalla aparecerá recomendando que se deben de usar múltiples tarjetas de red en el cluster, con el fin de evitar el punto de falla que existe al contar con únicamente una sola tarjeta de red.

Configuración de Red

La siguiente fase del asistente de instalación es entablar la prioridad de los adaptadores de red e introducir los datos para la configuración de la dirección de IP.

Después de configurar cómo deben de ser usados los adaptadores de red por el Cluster, la siguiente pantalla es usada para prioritizar los adaptadores de red. Las flechas de prioridad en el lado derecho de la pantalla pueden ser usados para especificar el orden en el cual el Cluster usará las tarjetas de red para la comunicación interna entre los nodos. *Cluster Server* siempre intentará usar la primera tarjeta de red listada para la comunicación entre los nodos. MS Cluster Server usa la siguiente tarjeta en la lista únicamente si no es posible realizar la comunicación a través de la primera.

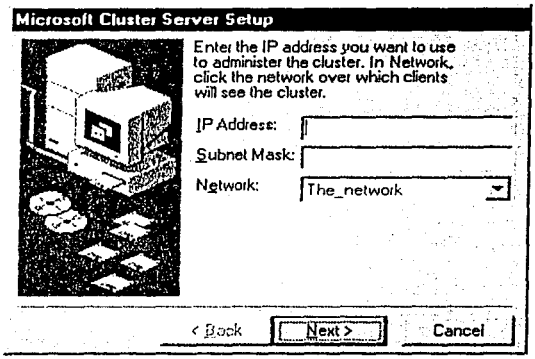


Figura 4.24 - Configuración de Red.

La siguiente pantalla es usada para configurar la dirección de IP y la mascara de red que va a ser usada por el cluster. También se especifica la red bajo la cual los clientes van a acceder al cluster.

Completando la Instalación en el Primer Nodo

El asistente de instalación completa el proceso de instalación del primer nodo copiando los archivos necesarios para así dar por terminada la instalación del Cluster. Una vez que los archivos del cluster fueron copiados, *Cluster Server* introduce sus llaves en el *registry*, los archivos de *log* sobre el disco de quórum son creados, y por último el servicio de *Cluster Server* es iniciado en el nodo.

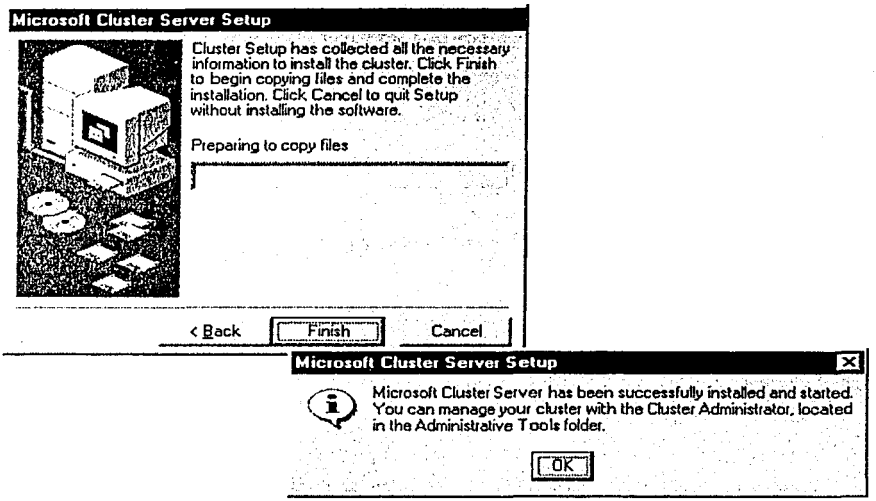


Figura 4.25 - Completando la Instalación en el Primer Nodo.

4.7.4.2 Instalando Cluster Server en el Segundo Nodo

La instalación de *Cluster Server* sobre el segundo nodo de un cluster requiere de muy poca información porque la mayoría de la información es obtenida del primer nodo del cluster. Por ejemplo, no se requiere de ninguna configuración para la red cuando se está instalando el segundo nodo del cluster. Durante el proceso de instalación de la red en el segundo nodo, la red es configurada basada en la configuración del primer nodo.

El asistente de instalación para el segundo nodo emplea el siguiente conjunto de pantallas usadas para la instalación del primer nodo:

- Bienvenida

- Aceptación de la Licencia de Hardware soportado
- Tipo de instalación a ejecutar

  Cuando se está instalando en segundo nodo, se debe de elegir la opción de Join an existing cluster y se debe de introducir el nombre del cluster al cual se desea unirse.

- Subdirectorio en donde se van a copiar los archivos del cluster.
- El password del usuario para iniciar los servicio del cluster

  El segundo nodo obtiene el nombre del usuario y del dominio del primer nodo del cluster. Estos datos son mostrados en sus respectivas cajas, pero no pueden ser modificados. Sin embargo, el administrador que está instalando el cluster debe de introducir el *password* para que el cluster pueda iniciar los servicios.

Cuando la configuración del segundo nodo es completada, el asistente de instalación copia los archivos, modifica el *registry*, e inicia los servicios del cluster.

4.7.4.3 Verificando si la Instalación fue Exitosa

Una instalación exitosa contiene valores estándar en el *registry* y pueden ser probados mediante varios métodos.

Los valores en el *registry* pueden localizarse bajo la llave de HKEY_LOCAL_MACHINE. Las pruebas pueden realizarse utilizando la utilería de *Cluster Administrator* y de los Servicios en el Panel de Control, y realizando "*ping´s*" al nombre del cluster.

Los valores dentro del *registry* en el cluster se encuentran bajo la llave de HKEY_LOCAL_MACHINE\CLUSTER. Bajo esta llave se encuentran las siguientes sub-llaves:

- *Groups*

- *NetworkInterfaces*
- *Networks*
- *Nodes*
- *Quorum*
- *Resources*
- *Resource Type*

Cada de una de la sub-llaves de arriba contienen la información que va a ser desplegada por la herramienta de *Cluster Administrator*. Por ejemplo, cuando un nuevo grupo es creado un nuevo registro es añadido bajo la llave de \HKEY_LOCAL_MACHINE\Cluster\Groups.

No existe ningún parámetro que pueda ser configurado en el *registry* para el Cluster. Sin embargo, un método para verificar si se instaló correctamente el cluster es el verificar que las llaves de arriba fueron creadas.

Esta llaves están almacenadas en *%windir%*\Cluster\Clusdb.

Además de la prueba anterior existen otros métodos para verificar si el cluster quedo bien instalado, y éstos son:

- Cluster Administrator

Una vez que se termina la instalación en el primer nodo del cluster, hay que ejecutar la herramienta del Cluster Administrator, y una vez que ésta inicia hay que intentar conectarse al cluster. Cuando la instalación sobre el segundo nodo acaba, se ejecuta la herramienta del Cluster Administrator en cualquiera de los nodos, nos conectamos al cluster, y después se verifica que el segundo nodo se encuentre listado.

- Los Servicios en el Panel de Control

  Hay que usar el programas de Services en el Panel de Control para verificar que el servicio del cluster se encuentra listado e iniciado.

- Subdirectorio del Cluster

  Hay que verificar que el proceso de instalación copió los archivos del cluster al subdirectorio que se especificó durante la instalación.

- Dar "Ping´s" al nombre del cluster

  Hay que abrir una ventana de DOS, y después se intenta darle un ping al nombre del cluster. Si el "ping" es exitoso, se verificará que el servicio del cluster se encuentra iniciado y fue capaz de registrar el nombre de red del cluster.

Una vez que se realizaron las pruebas pertinentes sobre si el cluster se encuentra bien instalado o no, se procede a observar la vista por defecto de la herramienta del Cluster Administrator. El proceso de instalación crea al menos dos grupos en cada cluster:

- *Disk Group* 1

  Este grupo contiene como recurso el disco que sirve como de quórum que fue seleccionado durante la instalación. Este grupo puede ser usado para almacenar otros recursos, y puede ser renombrado, si es necesario.

- Cluster Group

  Este grupo contiene un recurso de nombre de red para que el nombre del cluster que fue especificado durante la instalación. Además, también contiene un recurso de dirección de IP la cual es la dirección de IP del cluster y el recurso de *time service*. No existe ninguna limitante para agregar otros recursos a este grupo, sin embargo no es recomendable por que este grupo se recomienda que se use única y exclusivamente para la administración del cluster.

La herramienta de Cluster Administrator puede ser instalada en computadoras que tengan como sistema operativo Windows NT Server Enterprise

Edition, Windows NT Workstation 4.0 o Windows NT Server 4.0 con Service Pack 3 o superior, con el fin de administrar los clusters.

No es necesario que se tenga Windows NT Server Enterprise Edition para iniciar la instalación de la herramienta de Cluster Administrator, el asistente es lo bastante inteligente para detectar que no está corriendo sobre este sistema operativo y lo único que va a poder realizar es la instalación de la herramienta, las opciones de crear o unirse a un cluster no se encuentran disponibles.

Por último para instalar esta herramienta sobre una computadora corriendo Windows NT Server Enterprise Edition únicamente hay que seleccionar la opción de **Install Cluster Administrator.**

4.7.4.4 FUNCIONES DEL SERVICIO DE CLUSTER SERVER

El servicio de cluster server es responsable de ejecutar varios procesos al final de la instalación. Como son:

- Unirse a un cluster

Este proceso se ejecuta al final de la instalación del segundo nodo del cluster. Este proceso también se ejecuta cada vez que un nodo es reiniciado y este encuentra el otro nodo del cluster activo.

Cuando se está uniendo a un cluster, el Servicio de Cluster se inicia automáticamente, por defecto. El nodo únicamente configura y actualiza aquellos dispositivos que son locales al nodo, es decir que no pertenecen al cluster. Todos los dispositivos del cluster deben de estar fuera de línea durante la inicialización.

Una vez que el servicio inició, el nodo intenta comunicarse con el otro nodo del cluster. Cuando el nodo identifica a otro miembro del cluster, éste intenta autenticarlo, utilizando la base de datos del cluster.

Si el nodo localiza un nodo que es parte de un cluster existente, el nodo en este cluster intenta autentificar al nodo que se está uniendo y le regresa una

señal de éxito si la autenticación fue exitosa. Si el nombre del cluster es incorrecto, entonces el cluster rechaza la unión. Si el nombre del cluster es válido, pero la base de datos de la instancia está dañada, entonces se manda una señal de éxito parcial, indicando que la base de datos no se encuentra actualizada o en buen estado. Una vez que el nodo que se desea unir recibe esta señal procese a actualizar su base de datos del nodo del cluster que se encuentra activo. El nodo que se está uniendo traerá cualquier recurso que no se encuentre fuera de línea dentro del cluster en línea. Esto lo realiza utilizando los datos que tienen en el *registry* del cluster para así encontrar cualquier recurso compartido.

- Crear un cluster

Crear un cluster es un proceso muy similar al de unirse a uno. Cuando se está uniendo a un cluster, el nuevo nodo trata de identificar a un nodo activo en el cluster y se une a éste, como se describió anteriormente. Si el nuevo nodo falla en descubrir a un nodo activo durante el proceso de instalación, éste nodo formara un cluster.

Después de que transcurre el tiempo en el que un nodo trata de conectarse al cluster, este nodo entonces procede a formar su propio cluster. Esto lo realiza usando la información del cluster que se encuentra en su *registry* para determinar dónde se encuentra el recurso de quórum (que es el disco de quórum). El nodo intenta tomar a este recurso, y si tiene éxito, entonces verifica los posibles cambios en la base de datos del cluster que se encuentran registrados en los archivos de *log*. Si no hay ningún cambio, entonces el nodo es ya un cluster. Mas sin en cambio si existen algunos cambios registrados, entonces el nodo actualiza su base de datos local de acuerdo a los archivos de *log*, y después forma el cluster. Por último procese a poner en línea todos los recursos del cluster.

Si se llegase a presentar una situación en donde ambos nodos de un cluster inician el servicio de cluster al mismo tiempo, y ambos intentan apoderarse el disco de quórum, el servicio de cluster no iniciará en ninguno de los nodos. Por ejemplo, si se encienden al mismo tiempo dos computadoras idénticas después de una falla de energía. Para evitar esta situación, hay que incrementar o decrementar el valor de la opción **Show list for** dentro de la aplicación de **System** en el Panel de Control para uno de los nodos del cluster.

- Apaga un nodo del cluster

Hay ocasiones que se requiere apagar a uno de los nodos con el fin de poderles realizar operaciones de mantenimiento y/o actualizaciones. Cuando se apaga un nodo, el nodo manda un mensaje *ClusterExit* al otro nodo del cluster. Este mensaje le notifica al otro nodo que el nodo está dejando al cluster. El nodo que se separa del cluster, no espera ninguna respuesta, de manera inmediata procede a poner todos los recursos que el tenía fuera de línea y cierra todas la conexiones manejadas por el cluster.

El mandar un mensaje *ClusterExit* al otro nodo, ayuda a este nodo a poner los recursos que tenía el nodo que está dejando el cluster en línea de manera inmediata; mientras que si no se mandara este mensaje el nodo que queda tiene que esperar el tiempo de verificación de ver si el otro nodo está en línea para comenzar a poner a los recursos en línea.

4.8 CONFIGURANDO GRUPOS, DISCOS Y RECURSOS DE RED

La utilería de *Cluster Administrator* es usada para la configuración, manejo y administración de un cluster de Microsoft. En este punto hablaremos de esta utilería y explicaremos cómo se puede usar para configurar grupos, recursos de disco y recursos de red.

4.8.1 TAREAS ADMINISTRATIVAS

Una vez que ya se instaló el cluster, en el grupo de *Administrative Tools* se encontrará la utilería de *Cluster Administrator*. Un administrador puede usar esta utilería para configurar y manejar ambos nodos del cluster.

Esta herramienta puede ser utilizada para realizar las siguientes tareas administrativas:

- Agregar, borrar, y renombrar grupos y recursos.
- Cambiar el estado de los grupos y recursos.

4.8.2 REQUERIMIENTOS DE SOFTWARE

Microsoft Cluster Server debe de instalarse en una computadora con Windows NT Server Enterprise Edition. Microsoft Cluster Server soporta las siguientes configuraciones de Windows NT Server Enterprise Edition:

- Iniciar una falla.
- Mover grupos de un nodo a otro, y cambiar recursos de un grupo a otro.
- Conectarse a un cluster.

4.8.2.1 CONFIGURANDO NOMBRES DE GRUPOS Y RECURSOS

Un grupo o recurso puede ser agregado seleccionando la opción de **New** en el menú de **File** del *Cluster Administrator*, y después hay que seleccionar si se desea crear un **Grupo** o **Recurso**. Esto ocasionará que se ejecute ya sea el asistente de new Group o new resource. Estos asistentes guían al administrador a través del proceso de configuración.

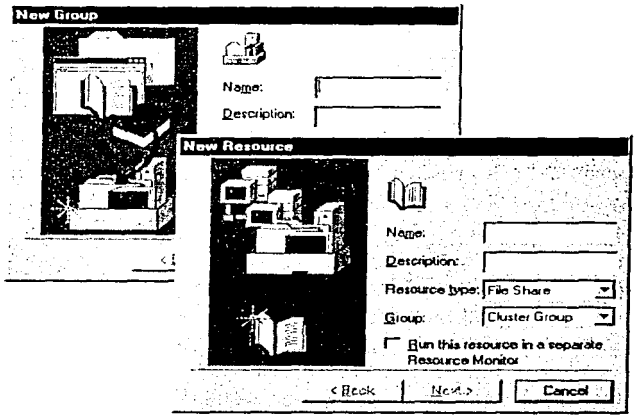


Figura 4.26 - Configurando Nombres de Grupos y Recursos.

Hay que notar que los siguientes datos no son configurables a través del asistente y tienen que ser configurados a través de la página de propiedades del *Cluster Group* (Grupo del Cluster) y *Cluster Resource* (Recurso del Cluster):

- Grupos – Los valores de *Failover* y *Failback*.
- Recursos – Los valores de *Restart, LooksAlive, IsAlive,* y *Pending Timeout*.

Para borrar o renombrar un grupo o un recurso hay que utilizar la utilería de *Cluster Administrator*. Cuando un grupo es borrado, todos los recursos que eran miembros del grupo son borrados también. Un recurso no puede ser borrado hasta que todos los recursos que dependen de él sean borrados.

4.8.2.2 Cambiando el Estado de Grupos y Recursos

Cluster Administrator puede ser usado para cambiar el estado de grupo y recursos de en línea a fuera de línea y viceversa. Existen tres maneras de realizar esta función:

- Sobre la barra de herramientas, hay que seleccionar ya sea el botón de **Bring Online** o **Take offline.**
- Accediendo al menú **File,** y seleccionar ya sea la opción de **Bring Online** o **Take offline.**
- Apretar el botón derecho del mouse sobre un grupo o recurso, y después dar un click en **Bring Online** o **Take offline.**

Cada vez que se cambia el estado de un grupo también de cambiarán el estado de todos los recursos que son miembros de ese grupo. Estos recursos son cambiados de estado según sean sus dependencias.

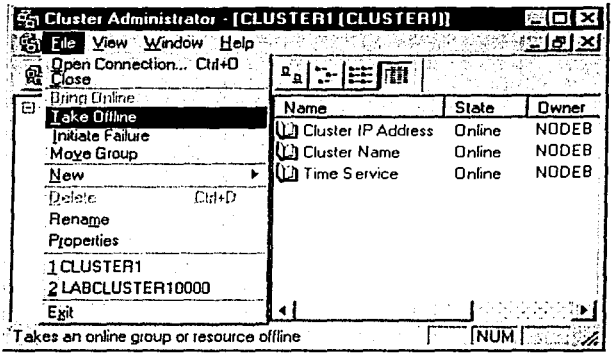


Figura 4.27 - Cambiando el Estado de Grupos y Recursos.

4.8.2.3 INICIAR UNA FALLA

La herramienta de Cluster Administrator incluye una opción que puede ser usada para "iniciar una falla". Esta opción puede ser usada para probar los valores de reinicio para un recurso o para probar también los valores de *failover* del grupo al cual el recurso es miembro.

Por ejemplo, esta opción se puede usar cuando un administrador desea que un nodo intente reiniciar un recurso 10 veces antes de que el recurso realice un *failover* hacia el otro nodo. El administrador puede seleccionar el recurso, seleccionar esta opción 11 veces, y verificar que el recurso realiza un *failover* hacia el otro nodo.

Una falla es iniciada cuando se usa la opción **Initiate Failure**, la cual se encuentra disponible en el menú **File**, o apretando el botón derecho del mouse sobre el recurso que se desea que falle.

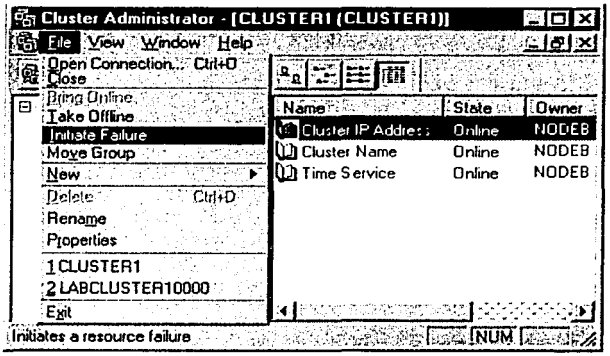


Figura 4.28 - Iniciando una Falla.

4.8.2.4 Transfiriendo Grupos y Recursos

Los administradores de Cluster pueden mover grupos de un nodo a otro nodo y cambiar recursos de un grupo a otro.

La herramienta de *Cluster Administrator* puede ser usada para transferir la propiedad de un grupo, lo único que tienen que realizar es mover el grupo de un nodo al otro. Esta operación se realiza típicamente cuando se está preparando un nodo para ser apagado del cluster (por ejemplo, cuando se le está instalando un *Service Pack*).

Un grupo puede ser movido usando una de las siguientes opciones:

- La opción de **Move Group** en el menú **File**.
- Apretando el botón derecho del mouse sobre el grupo, y después dar click en **Move Group**.
- Arrastrando y dejando el grupo con el mouse entre los nodos en el *cluster administrator*.

Cuando un grupo es movido, los recursos que se encuentran dentro de él son puestos fuera de línea, el grupo es movido al otro nodo, y después todos los recursos son puestos nuevamente en línea.

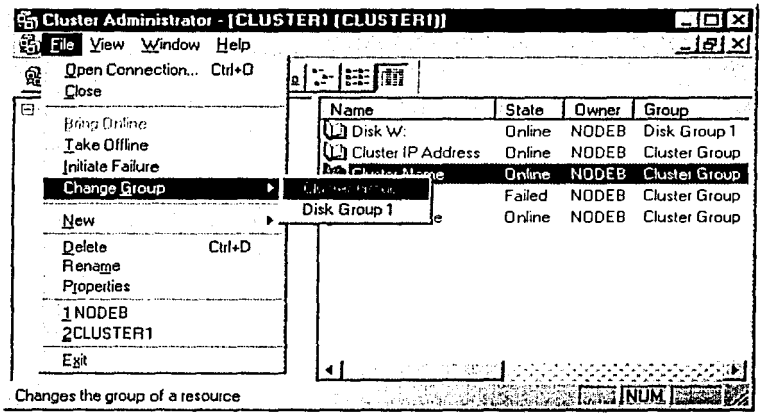


Figura 4.29 - Transfiriendo Grupos y Recursos.

Para transferir la propiedad de un recurso, el grupo al cual pertenece el recurso debe de ser cambiado. Un recurso puede ser cambiado de grupo usando una de las siguientes opciones:

- Sobre el menú de **File**, dar click en **Change Group**.
- Apretar el botón derecho del mouse, y después dar click en la opción de **Change Group**.
- Arrastrar y dejar el recurso entre los grupos del cluster con la herramienta de Cluster Administrator.

Si el recurso es movido a un grupo que se encuentra en el otro nodo, éste se pondrá fuera de línea, y una vez que sea movido éste se pondrá nuevamente en línea. Si el recurso es movido entre grupos que se encuentran en el mismo nodo, éste permanece en línea.

4.8.2.5 CONECTÁNDOSE A UN CLUSTER

La primera vez que se ejecuta la herramienta de *Cluster Administrator*, ésta le preguntará al administrador el nombre del cluster al cual desea conectarse. El administrador puede introducir ya sea el nombre del cluster o el nombre de un nodo del cluster. Sin embargo, es preferible usar el nombre del cluster. Se recomienda usar el nombre del nodo sólo cuando el nombre del cluster no funciona.

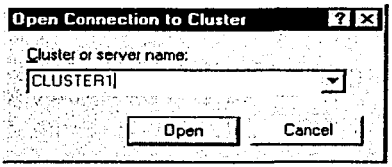


Figura 4.30 - Conectándose a un Cluster.

Hay que tomar en cuenta que cuando se introduce el nombre del cluster, la ventana de administración tendrá como encabezado **Cluster Administrator** - ***ClusterName (Nombre del Cluster)***. Si lo que se introduce es el nombre de un nodo, la ventana de administración tendrá como encabezado **Cluster Administrator** - ***ClusterName (Nombre del Nodo)***.

Después de que se usó la herramienta de *Cluster Administrator* por primera vez, y ésta se vuelve a usar, el cluster administrator tratará de restablecer la ultima conexión de manera automática. Otra manera para poderse conectar a un cluster o a cualquiera de sus nodos es usar la opción de **Open Connection** dentro del menú **File** de esta herramienta.

No es necesario que se administre el cluster utilizando cualquiera de los nodos, la utilería de *Cluster Administrator* puede ser instalada sobre cualquier computadora que no sea necesariamente un nodo del cluster únicamente ejecutando el programa de instalación de *Cluster Server* e instalar únicamente la herramienta de *Cluster Administrator*. La computadora a la cual se le instale esta utilería podrá ser usada para administrar al cluster de manera remota.

4.8.3 PARÁMETROS DE CONFIGURACIÓN PARA GRUPOS

Cuando se configura un grupo en un Cluster, las siguientes páginas de configuración se encuentran disponibles:

- Propiedades Generales de Configuración (General Properties Configuration)
- Propiedades de Configuración de *Failover* (Failover Properties Configuration)
- Propiedades de Configuración de *Failback* (Failback Properties Configuration)

Las páginas de configuración de un grupo pueden ser accedidas utilizando una de las siguientes opciones:

- Dar un click sobre el grupo, y después dar otro sobre el menú FILE, y elegir la opción de PROPERTIES.
- Dar click con el botón derecho sobre un grupo, y después dar click en PROPERTIES.
- Dar click sobre un grupo, y después dar click sobre el icono de PROPERTIES.

4.8.3.1 PÁGINA DE PROPIEDADES GENERALES

Para acceder la página de propiedades generales, en la ventana de propiedades del grupo, hay que dar click en el folder de General.
En este folder podemos encontrar las siguientes opciones de un grupo:

Opción	*Descripción*
Name (Nombre)	Este nombre debe de ser único y es el nombre del grupo.
Description (Descripción)	Descripción del grupo. (Opcional).
Preferred Owners (Orden de Preferencia)	El nodo o nodos que pueden ser dueños del grupo. Se debe de especificar el orden de preferencia en el que un grupo debe de estar para cuando ocurra un *failback*.

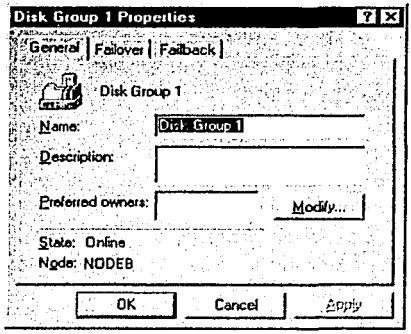


Figura 4.31 - Página de Propiedades Generales.

Estos valores pueden ser editados en cualquier momento simplemente accediendo esta página de propiedades.

4.8.3.2 Página de Propiedades de Failover

Para acceder la página de propiedades de Failover, en la ventana de propiedades del grupo, hay que dar click en el folder de Failover. En esta página se configuran los valores de *threshold* y *period*.

A continuación se da una breve descripción de lo que son estos valores:

Opción	*Descripción*
Threshold	El número máximo de veces de *failover* que un grupo tiene como permitido hacer durante el periodo que se le indique. Si este número es excedido dentro del periodo especificado, *Cluster Server* pondrá a este grupo fuera de línea. El valor

	de default de este parámetro es de 10.
Period	Es el numero máximo de horas durante el cual el *failover* threshold no debe de exceder. Si el *failover* threshold es excedido dentro de este intervalo de tiempo, *Cluster Server* pondrá a este grupo fuera de línea, El valor de default de este parámetro es de 6 horas.

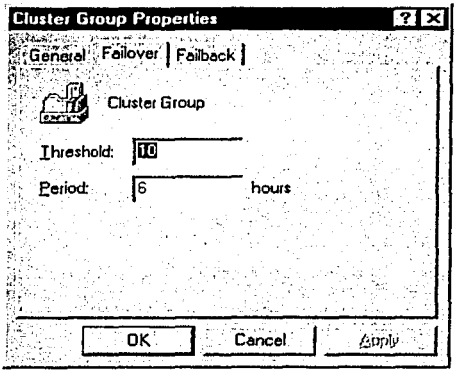


Figura 4.32 - Pagina de Propiedades de Failover.

Asumiendo los valores por default, si un grupo realizara un *failover* por décima primera ocasión en un periodo de 6 horas, este grupo será puesto fuera de línea.

4.8.3.3 PÁGINA DE PROPIEDADES DE FAILBACK

La página de propiedades de *Failback* permite al administrador ya sea prevenir que ocurra un *failback* o configurar que ocurra éste. Para acceder la página de propiedades de *failback*, en la ventana de propiedades del grupo, hay que dar click en el folder de *failback*.

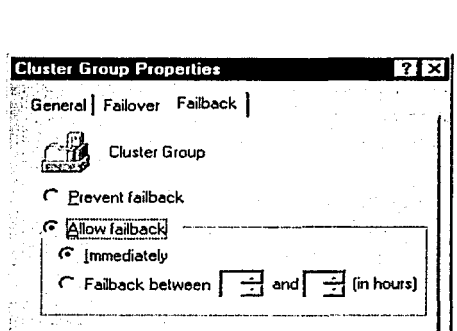


Figura 4.33 - Página de Propiedades de *Failback*.

Esta página de propiedades cuenta con cuatro parámetros de configuración:

Opción	*Descripción*
Prevent failback	Cuando un nodo se encuentra en línea, un grupo no realizará una operación de *failback* al nodo en el cual estaba corriendo antes de que ocurriera el *failover*. Éste es el valor por default.

Allow failback	Cuando un nodo vuelve a estar en línea, el grupo va a realizar una operación de *failback* al nodo en el cual estaba corriendo antes de que ocurriera un *failover*.
Immediately	El *Failback* se ejecutará en el momento en que el cluster detecta que el nodo original ya está nuevamente en línea.
Failback between	Se realizará la operación de *Failback* únicamente en el horario especificado.

4.8.4 Parámetros Comunes de Configuración de Recursos

Sin importar el tipo de recurso, los siguientes parámetros de configuración son comunes para todos los tipos de recursos:

- Propiedades Generales (General properties).
- Propiedades de Dependencia (Dependencies properties).
- Propiedades Avanzadas (Advanced properties)

Estos valores pueden ser encontrados en la página de propiedades del recurso y pueden ser accedidas utilizando una de las siguientes opciones:

- Dar un click sobre un recurso, y después dar otro sobre el menú FILE, y elegir la opción de PROPERTIES.
- Dar click con el botón derecho sobre un recurso, y después dar click en PROPERTIES.
- Dar click sobre un recurso, y después dar click sobre el icono de PROPERTIES.

4.8.4.1 PROPIEDADES GENERALES

Para acceder la página de propiedades generales, en la ventana de propiedades del recurso, hay que dar click en el folder de General. A continuación se listan las opciones que vienen en esta página y sus descripciones:

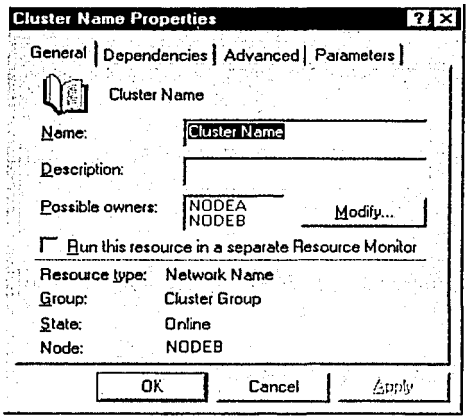


Figura 4.34 - Propiedades Generales.

Opción	*Descripción*
Name (Nombre)	El nombre único del recurso. Este valor no puede ser cambiado.
Description (Descripción)	Una descripción opcional acerca de para qué es el recurso.
Preferred Owners (Orden de Preferencia)	El nodo o nodos que pueden ser dueños del recurso. Ambos nodos deben de ser posibles dueños del recurso, con el fin de que pueda existir un *failover* del mismo. Para editar esta opción, hay que dar click en el botón de **Modify**.

Run this resource in a separate Resource Monitor (Ejecuta este recurso en un Monitor de Recursos Separados)	Configura al recurso para que se ejecute en un monitor de recursos independiente. Esto es mucha utilidad para un recurso que no se encuentra funcionando del todo bien. En este caso, el recurso únicamente va a afectar un Monitor de Recursos en lugar de que afecte un Monitor de Recursos que se encuentra monitoreando múltiples recursos.
Status	Muestra el tipo de recurso, a que grupo es miembro, el estado del recurso, y el nodo en el cual se encuentra ejecutando en ese momento. Estos valores no pueden ser cambiados.

4.8.4.2 Propiedades de Dependencias

Para acceder la página de propiedades de dependencias, en la ventana de propiedades del recurso, hay que dar click en el folder de Dependencies.

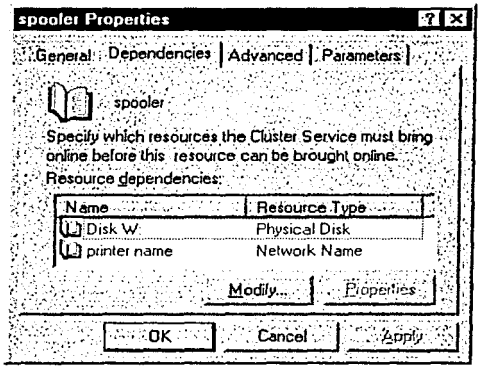


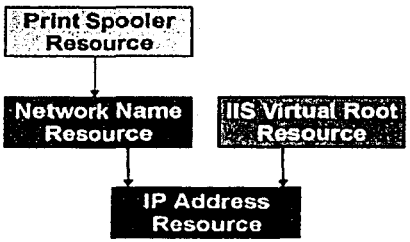
Figura 4.35 - Propiedades de Dependencias.

Entendemos como dependencias a la relación entre dos o más recursos en un mismo grupo sobre un mismo nodo. Por ejemplo, una aplicación es dependiente del disco en donde se encuentran sus datos.

Las siguientes relaciones de dependencias gobiernan la interacción entre recursos:

- Un recurso puede depender de cualquier número de recursos.
- Un recurso es puesto en línea si y solo si, cuando todos los recursos de los cuales éste depende son puestos en línea.
- Un recurso es puesto fuera de línea antes de que cualquiera de los recursos de los cuales él depende sea puesto fuera de línea.
- Un recurso y todos los recursos de los cuales éste depende deben de realizar una operación de *failover* juntos.

Para definir las dependencias entre recursos, es recomendable crear un árbol de dependencias. Un árbol de dependencias es útil para visualizar la relación de dependencias entre recursos y determinar cómo los recursos van a interactuar. Por ejemplo, si un recurso necesita ser puesto fuera de línea para una tarea administrativa, un árbol de dependencias va a mostrar que otros recursos van a ser afectados.



En el diagrama, el recurso de *Print Spooler* depende directamente del recurso de *Network Name*, el cual a su vez éste depende del recurso de *IP Address*. Por lo tanto, el recurso de *Print Spooler* indirectamente depende del recurso de IP Address.

En este ejemplo, tanto el recurso de *Network Name* y el recurso de *Microsoft Internet Information Server* (IIS) depende del recurso de *IP Address*. Sin embargo, no existe dependencia alguna entre el recurso de *Network Name* y el recurso de IIS. De hecho, estas relaciones pueden ser vistas como dos árboles de dependencias separados: un árbol incluye el recurso de *IIS* y el recurso de *IP Address* y el otro árbol contiene el recurso de *Print Spooler*, el recurso de *Network Name* y el recurso de *IP Address*.

Si un administrador quisiera modificar al recurso de *IP Address*, este recurso deberá ser puesto fuera de línea. Esto también va a afectar al nombre de red, quien a su vez a afectar a los recursos de *IIS* y *Print Spooler*.

4.8.4.3 Propiedades Avanzadas

Para acceder la página de propiedades avanzadas, en la ventana de propiedades del recurso, hay que dar click en el folder de *Advanced*. A continuación se listan las opciones que vienen en esta página y sus descripciones:

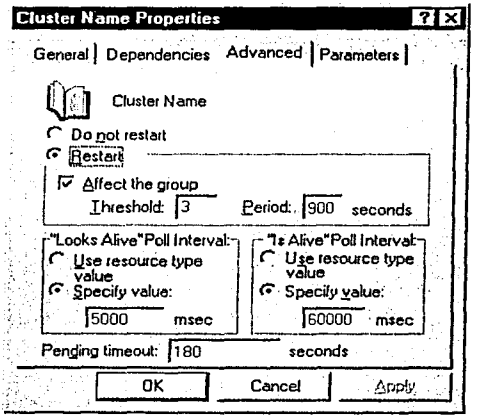


Figura 4.36 - Propiedades Avanzadas.

Opción	*Descripción*
Do not restart	Configura al recurso para que no reinicie en caso de falla.
Restart	Configura al recurso para que reinicie en caso de falla. Ésta es la opción por defecto.
Affect the group	Permite al grupo que puede realizar una operación de *failover* en caso de que el recurso falle. Si esta caja no está seleccionada, una falla en el recurso nunca va a causar que el grupo realice una operación de *failover*. Ésta es la opción seleccionada por defecto.
Threshold	El número de veces que se va a reintentar reiniciar el recurso antes de que se realice un *failover*.
Period	El tiempo en el cual el número de *threshold* va a intentar reiniciar el recurso.
"LooksAlive" Poll Interval	Especifica qué tan seguido el servicio de Cluster va a checar el estado del recurso para determinar si éste se encuentra activo.
Use resource type value (under "LooksAlive" Poll Interval	Si esta opción es seleccionada, el número por defecto para el tipo de recurso es usado.
Specify value (under "LooksAlive' Poll Interval)	Especifica que tan seguido el servicio de Cluster checa el estado del recurso.
"IsAlive" Poll Interval	Especifica qué tan seguido el servicio de Cluster checa el estado del recurso para determinar si éste está en línea. El valor por defecto para este parámetro es de 60,000 milisegundos.
Use resource type value (under	Si esta opción es seleccionada, el

"IsAlive" Poll Interval	valor por defecto del tipo de recurso es usado.
Specify value (under "IsAlive" Poll Interval)	Especifica qué tan seguido el servicio del Cluster checa si el recurso se encuentra en línea.
Pending Timeout	Especifica el tiempo que un recurso se puede encontrar en el estado de "pending", sin importar si se va a poner el recurso en línea o fuera de línea. El valor es de 900 segundos

4.8.5 PARÁMETROS ESPECÍFICOS DE CONFIGURACIÓN DE RECURSOS

Además de los parámetros de configuración comunes definidos en las paginas de propiedades Generales, de Dependencias y Avanzada, existen parámetros de configuración específicos a cada tipo de recurso. Dando un click en el folder de *Parameters* en la ventana de propiedades se permite configurar a cualquiera de los siguientes tipos de los recursos:

- File Share.
- IIS virtual root.
- Network Name
- Physical disk
- IP address

Cabe mencionar que el recurso de *time service* no tiene parámetros de configuración específicos. Este recurso puede ser completamente configurado y

administrado usando las páginas de propiedades Generales, de Dependencias y Avanzada.

4.8.5.1 Valores Particulares del Recurso File Share

El recurso de *file share* (compartición de archivos) es usado para crear puntos de compartición sobre cualquier disco o discos compartidos entre los nodos.

Si el recurso de *file share* fue identificado cuando el recurso fue creado, dando un click en el folder de *Parameters* en la ventana de propiedades del recurso nos proveerá acceso a la página de propiedades de este tipo de recurso. A continuación se describen las opciones que aparecen en este folder:

Opción	*Descripción*
Share Name (Nombre)	El nombre único del recurso.
Path (Ruta)	Ruta en donde se encuentra el directorio a compartir sobre el disco compartido.
Comment (Comentario)	Descripción opcional para este recurso
User Limit (Limite de Usuarios)	Número máximo de conexiones simultáneas al recurso.
Permissions (Permisos)	Este botón permite acceder a la ventana de configuración de permisos para el directorio compartido.

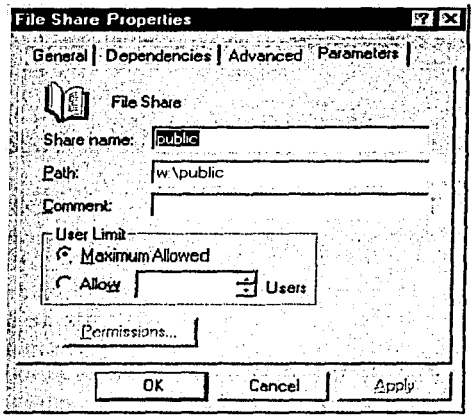


Figura 4.37- Parámetros Específicos para un Recurso de File Share.

Cuando se está utilizando el *file system* de NT (NTFS), se pueden poner permisos sobre el subdirectorio y así compartir los archivos. Sin embargo, existen dos puntos que deben de ser recordados cuando se están poniendo permisos sobre una partición NTFS:

- Ambos nodos del cluster deben de ser miembros del mismo dominio para que los permisos se encuentren disponibles cuando el recurso se encuentre en línea en cualquiera de los nodos.

- La cuenta de usuario del servicio del cluster que usa para validarse debe de tener al menos permiso de sólo lectura sobre el directorio que se desea compartir. Si esta cuenta no tiene al menos este permiso, el servicio del Cluster va a ser incapaz de poner el recurso en línea.

Como se discutió en propiedades de dependencias, todos los "*file shares*" dependen de un recurso de disco; aunque, el grupo que contiene al *file share* debe también de contener al recurso de disco. Además, es recomendable que este grupo también contenga al recurso de "*network name*" que va a ser usado para el *file share*, en vez de que se use el nombre del cluster. Esto es por que si ocurre un *failover* sobre el grupo en donde está el disco y el directorio compartido, pero no

del grupo que contiene el nombre del Cluster, los usuarios van a seguir siendo capaces de tener acceso al *file share* (archivo compartido).

4.8.5.2 VALORES PARTICULARES DEL RECURSO IIS VIRTUAL ROOT

El recurso de *IIS virtual root* (Directorio Raíz Virtual del Servidor de Información de Internet) es usado para proveer capacidades de *failover* para los directorios virtuales del Servidor de Información de Internet. Un directorio virtual es un directorio que se encuentra fuera del directorio raíz que aparece en el *browser* como un subdirectorio del directorio raíz. Éste se crea automáticamente en el servidor de información de internet cuando un recurso de *IIS virtual root* es creado en el Cluster. Aunque, no es necesario crear el directorio virtual en el servidor de información de internet antes de crear el recurso.

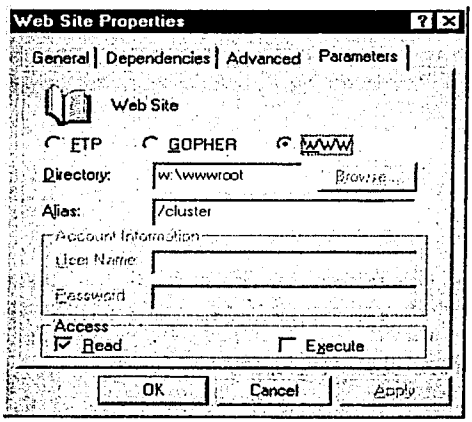


Figura 4.38 - Parámetros Específicos del Recurso IIS Virtual Root.

A continuación se nombran y describen la opciones específicas del recurso IIS *virtual root*.

Opción	*Descripción*
FTP, GOPHER, WWW	Especifica si el directorio es un directorio virtual de FTP, GOPHER, o de WWW.
Directory (Directorio)	La ruta del directorio.
Alias	Nombre que va a ser usado para tener acceso al directorio.
Account Information	El *username* y el *password* introducidos aquí deben de contar con los permisos suficientes para tener acceso a la ruta del UNC (Universal naming convention) sobre la red. Un directorio de IIS virtual que es accedido por la red debe de estar en el mismo dominio que el servidor de IIS. Esta caja se activa únicamente si el directorio especificado es una ruta UNC (por ejemplo \\IIS\ClusterServer)
Access (Acceso)	Controla los atributos de los directorios virtuales de FTP y WWW. Estos atributos son exactamente los mismos que los valores puestos en el *Internet Service Manager*.

4.8.5.3 VALORES PARTICULARES DEL RECURSO NETWORK NAME

El recurso *network name* (Nombre de Red), se utiliza para crear un nombre NetBIOS para el cluster. Este nombre es el que va a ser usado para poder acceder cualquier recurso en el grupo que depende de un nombre NetBIOS, como por ejemplo un *file share*.

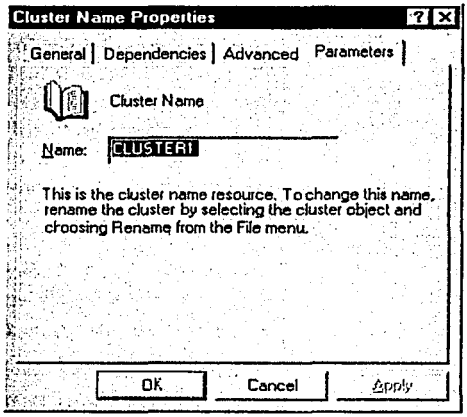


Figura 4.39 - Parámetros Específicos del Recurso *Network Name*.

A continuación se nombran y describen la opciones específicas del recurso *Network Name*.

Opción	*Descripción*
Name (Nombre)	Es el nombre NetBIOS a utilizar. Este nombre debe ser único.

4.8.5.4 VALORES PARTICULARES DEL RECURSO PHYSICAL DISK (DISCO FÍSICO)

El recurso physical disk es usado para agregar más discos compartidos al cluster. Para poder hacer esto, éste debe de estar en línea y visible en uno de los nodos.

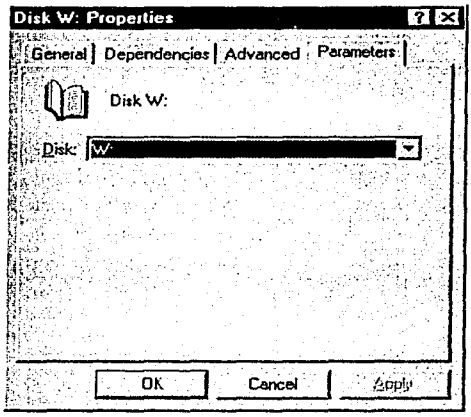


Figura 4.40 - Parámetros Específicos del Recurso *Physical Disk*.

A continuación se nombran y describen la opciones específicas del recurso *Physical Disk*.

Opción	*Descripción*
Disk (Disco)	Letra de la unidad del disco.

4.8.5.5 VALORES PARTICULARES DEL RECURSO IP ADDRESS (DIRECCIÓN DE IP)

Cada servidor virtual (tales como un servidor de impresión y un servidor de IIS) debe de contar con su propio recurso de *IP address* y *network name*. El recurso de *IP address* es usado para darle una dirección de IP adicional al cluster para podérsela asignar a un servidor virtual.

A continuación se nombran y describen las opciones específicas del recurso *IP Address*.

Opción	*Descripción*
Network to use (Red a utilizar)	La tarjeta de red bajo la cual se va a utilizar la dirección de IP.
Address (Dirección)	La dirección de IP.
Subnet mask	La *subnet mask* de la dirección de IP.

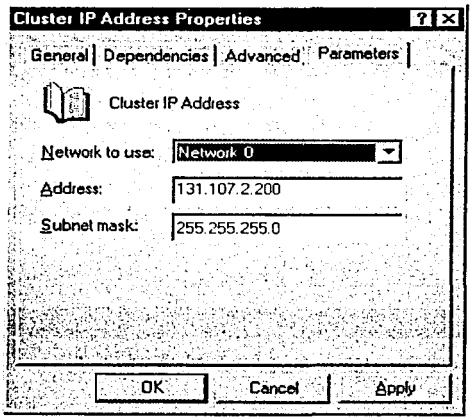


Figura 4.41 - Parámetros Específicos del Recurso *IP Address*.

4.8.6 CONFIGURACIÓN DEL CLUSTER

Además de los grupos de configuración y los recursos, existen también los siguientes parámetros de configuración del cluster:

- Quorum log.
- Network priority (Prioridad de red).
- Network usage (Utilización de la red).
- Network adapter (Adaptador de red)

4.8.6.1 CONFIGURACIÓN DEL QUORUM LOG

Cuando el recurso de quórum necesita ser cambiado o configurado, como en el caso de que se instala un nuevo disco en el cluster con el fin de que éste sea el nuevo *quórum disk*, el folder del *quórum disk* en la ventana de propiedades del Cluster es usada.

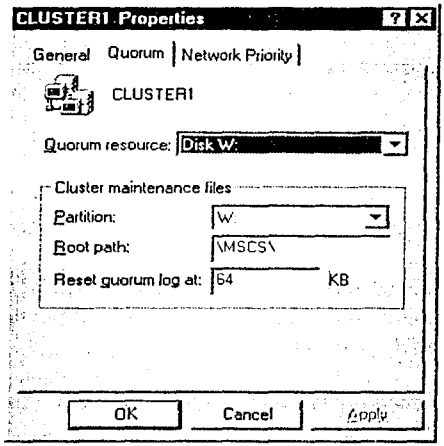


Figura 4.42 - Configuración del *Quorum Log*.

A continuación se nombran y describen la opciones y su uso.

Opción	*Uso*
Quorum resource	Para seleccionar el disco *SCSI* compartido que se va a usar como *quórum resource.*
Partition (Partición)	Se utiliza para seleccionar la partición en donde el *quórum log* va a ser almacenado.
Root Path	El subdirectorio en el cual el *quórum log* está almacenado. Este parámetro tiene como valor por defecto de \MSCS\.
Reset quorum log at	Aquí se define el tamaño en el cual el archivo de *log* va a comenzar a rescribirse. Por defecto este valor es de 64 KB.

Para acceder esta página de propiedades, hay que dar click en el nombre del cluster y después hay que darle un click derecho en el nombre del cluster, y después hay que darle un click en *Properties*, o sobre el menú de *File* hay que seleccionar *Properties*.

4.8.6.2 Configurando la Prioridad de las Conexiones de Red

Para cambiar la prioridad bajo la cual el servicio del Cluster va a usar las conexiones de red entre los nodos del cluster , se usa el folder de Network Priority en la ventana de propiedades del cluster.

A continuación se describen los botones que aparecen en este folder y su uso.

Opción	*Uso*
Move Up (Mover hacia arriba)	Mueve la conexión de red seleccionada un nivel arriba.

Move Down (Mover hacia abajo)	Mueve la conexión de red seleccionada un nivel hacia abajo.
Properties (Propiedades)	Despliega la caja de propiedades de la conexión seleccionada. Otra manera de poder desplegar esta caja de propiedades es utilizando el folder de Networks, dar un click derecho en la conexión seleccionada y después dar un click en Properties.

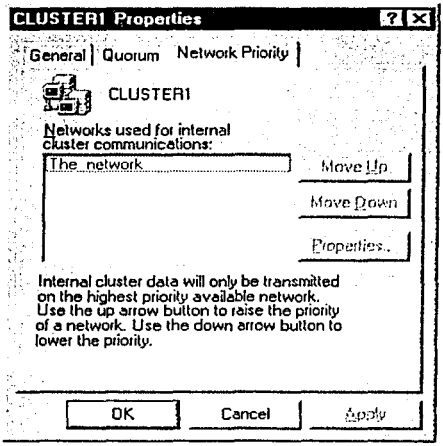


Figura 4.43 - Configurando la Prioridad de las Conexiones de Red.

Para acceder esta página de propiedades, hay que dar click en el nombre del cluster y después hay que darle un click derecho en el nombre del cluster, y después hay que darle un click en *Properties*, o sobre el menú de *File* hay que seleccionar *Properties*.

4.8.6.3 CONFIGURANDO LA UTILIZACIÓN DE LA CONEXIÓN DE RED

Para configurar cómo el Cluster va a utilizar las conexiones de red, ya sea para todas las comunicaciones, para la comunicación interna entre nodos, o para atender las peticiones de los clientes únicamente se utiliza la ventana de propiedades de las conexiones de red.

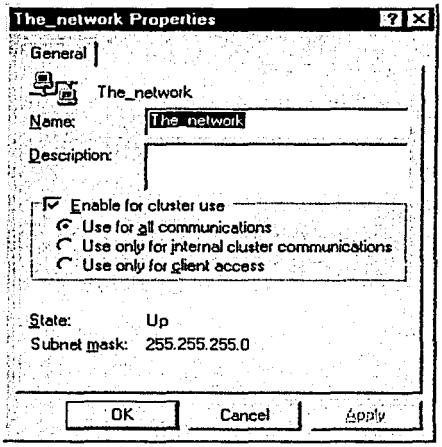


Figura 4.44 - Configurando la Utilización de la Conexión de Red.

A continuación se describen las opciones que aparecen en esta ventana y su uso.

Opción	*Uso*
Name (Nombre)	El nombre que se le dio a la conexión de red durante la instalación.
Description (Descripción)	Una breve descripción de para qué es la conexión.

Enable for cluster use (Habilita al cluster a utilizar esta conexión)	Si se selecciona esta caja, entonces se le permite al servicio de cluster usar este adaptador de red. Esta caja esta seleccionada por defecto.
Use for all communications (Usar para todas las comunicaciones)	Seleccionando esta opción causará que el servicio de cluster use al adaptador para las comunicaciones entre los nodos y para las comunicaciones con los clientes. Esta opción está seleccionada por defecto.
Use only for internal cluster communications (Úsese únicamente para la comunicación interna del cluster)	Seleccionando esta opción causará que el servicio del cluster use este adaptador para la comunicación interna del cluster entre los nodos.
Use only for client access (Úsese únicamente para dar acceso a los clientes)	Seleccionando esta opción causará que el servicio del cluster use este adaptador de red para las comunicaciones con los clientes. La comunicación entre nodos no tomará lugar cuando se utilice este adaptador de red.

Para acceder esta ventana de propiedades, hay que expandir *Networks*, dar click en el nombre de la conexión, y después hay que darle un click derecho y por último dar click en *Properties*.

4.8.6.4 CONFIGURACIÓN DEL ADAPTADOR DE RED

La ventana de propiedades para un adaptador de red es utilizada únicamente para configurar la descripción del adaptador de red. Sin embargo, las propiedades del adaptador de red nos provee de la siguientes información, que puede llegar a ser útil:

- State (Estado)
  Éste indica si el adaptador de red se encuentra o no funcionando.

- *Address* (Dirección)
  Nos indica la dirección IP que el adaptador de red está usando actualmente.

Para acceder esta ventana, hay que expandir *Networks*, dar un click en el nombre del adaptador de red, después dar un click derecho sobre éste, y por último hay que elegir *Properties*.

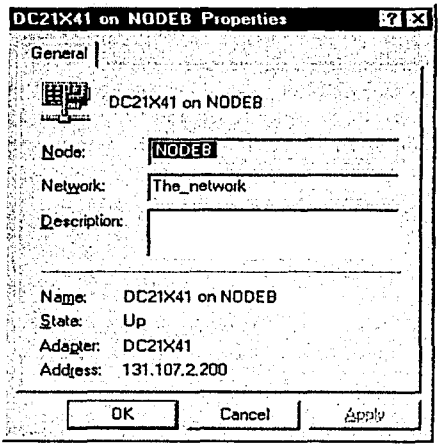


Figura 4.45 - Configuración del Adaptador de Red.

4.9 CONFIGURANDO IMPRESORAS, APLICACIONES, Y SERVICIOS

En el modulo 5.8 "Configurando Grupos, Discos y Recursos de Red" se cubrió el cómo configurar grupos, discos, y recursos de red anterior. La utilería de *Cluster Administrator* también se utiliza para configurar impresoras, aplicaciones, y servicios. Este módulo explica estas tareas.

4.9.1 CREANDO UN SERVIDOR DE IMPRESIÓN EN CLUSTER

El proceso para crear un *print share* en un cluster es mucho más complejo que el proceso de crear un *file share*. Para hacerlo se requieren los siguientes pasos (los primeros cuatro pasos fueron cubiertos en el Modulo 4.8 "Configurando Grupos, Discos y Recursos de Red"):

Para crear un printer share en un cluster

1. Crea un grupo para el *print spooler*.
   Este grupo es va a ser usado para almacenar todos los recursos necesarios para crear un *print share*.

2. Agrega un recurso de *Internet Protocol* (IP) al grupo.
   Un recurso de *IP address* debe ser agregado al grupo. Esto es por que el recurso de *Network name* que se va a crear a continuación va a depender de esta dirección de IP. Si los nodos del cluster tienen instalado el servicio de Transmission Control Protocol/Internet Protocol (TCP/IP) *Print service* e iniciado, todos los clientes van a poder acceder a la impresora a través de esta dirección de IP utilizando la utilería de *Line Printer* (LPR).

3. Agrega un recurso de *network name* al grupo.
   Este es el nombre del servidor de impresión que va a ser usado por los clientes cuando se conecten al *print share*.

4. Agregar un recurso de disco al grupo.
   En este recurso de disco es donde se van a almacenar los archivos del *spooler* de impresión.

5. Agregar un recurso de *print spooler* en el grupo.
   Este es el *spooler* para cualquier *printer shares*.

6. Crear los puertos de impresión necesarios e instalar los *drivers* de impresión necesarios. Esto se necesita realizar en ambos nodos del cluster.
   Los puertos y los *drivers* se instalan utilizando el asistente de *Add Printer*.

7. Agrega una impresora al *spooler* del cluster.
   Una vez que los recursos han sido creados, y los puertos y los drivers sean instalados, el *share* (recurso compartido) es creado.

Todos los pasos anteriores se pueden *realizar remotamente utilizando ya sea el Administrador del Cluster o el asistente de Add Printers*. Con excepción del paso 6, en donde se requiere de un administrador en el nodo.

4.9.1.1 RECURSOS DE SPOOLER

El recurso de *spooler* es usado para crear un spooler de impresión en el cluster de tal manera que el cluster pueda ser usado como un servidor de impresión. Sin la creación de este recurso, no es posible agregar impresoras al cluster.

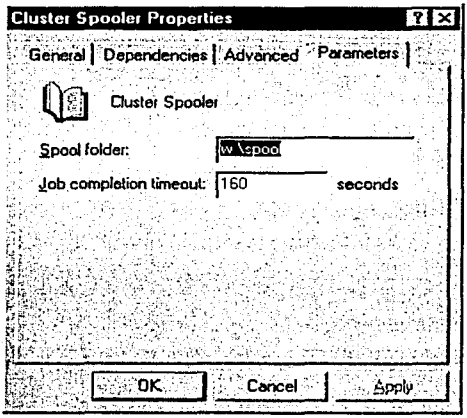


Figura 4.46 - Recursos de Spooler.

Los parámetros de la página de propiedades del recurso de *print spooler* contiene las siguientes opciones de configuración:

Opción	*Descripción*
Spool folder	La ruta en donde el archivo de spool va a ser almacenado.
Job completion timeout	Cuánto tiempo le puede tomar al documento llegar a la impresora desde la computadora, antes de que la impresora pare de tratar de imprimir el documento.

Un recurso de *print spooler* depende de un recurso de nombre de red y de un recurso de disco para el archivo de *spool*.

4.9.1.2 Creando Puertos e Instalando Drivers de Impresión

Después de crear el grupo y los recursos que son requeridos para hacer de un cluster un servidor de impresión, es necesario realizar los siguientes dos pasos en ambos nodos del cluster:

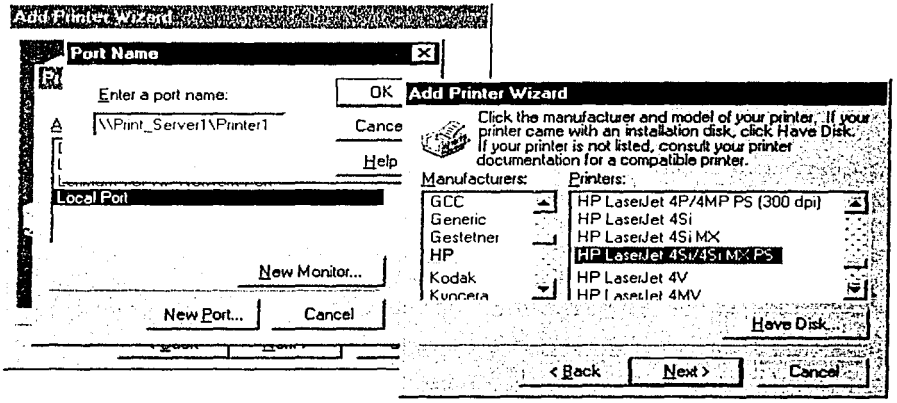


Figura 4.47 - Creando Puertos e Instalando Drivers de Impresión.

1. Crear los puertos de impresión.

   Este paso requiere que una conexión sea establecida desde cada nodo a la impresora de red o al servidor de impresión. Por ejemplo, una conexión puede ser establecida a una impresora TCP/IP Line Printer Daemon (LPD) o a otra computadora corriendo Microsoft Windows NT que está compartiendo una impresora, como por ejemplo \\Print server\printer. Para que el *failover* funcione, no es posible usar un puerto local, tal como LPT1.

2. Instalar los *drivers* de impresión para las impresoras.

   Ambos nodos del cluster deben de tener los *drivers* de impresión instalados para los sistemas operativos de todos los clientes que se van a conectar a la impresora compartida.

Realmente hay que asegurarse de realizar estos pasos en ambos nodos del cluster. Si por alguna razón únicamente se realizaran estos pasos sobre un nodo, y después se realiza un *failover* del grupo de impresión, el *spooler* de impresión sí se pondrá en línea y todos los clientes van a ser capaces de enviar sus trabajos de impresión. Sin embargo, todos los trabajos de impresión fallarán debido a que el *spooler* no va a tener un puerto de impresión en el cual pueda enviar los trabajos de impresión. Por último cabe mencionar que estos pasos no se realizan utilizando la herramienta de *cluster administrator*.

4.9.1.3 AGREGANDO UN SHARE DE IMPRESIÓN

Después de crear el grupo y los recursos en el cluster, los puertos de impresión, y agregar los *drivers* de impresión en cada nodo, un *share* de impresión puede entonces ser creado en el cluster. Un *share* de impresión es creado utilizando el asistente de *Add Printer*, en gran manera esto se realiza como si se estuviera creando un servidor de impresión en un servidor Windows NT. Sin embargo, el método usado para acceder al asistente de *Add Printer* es un poco diferente.

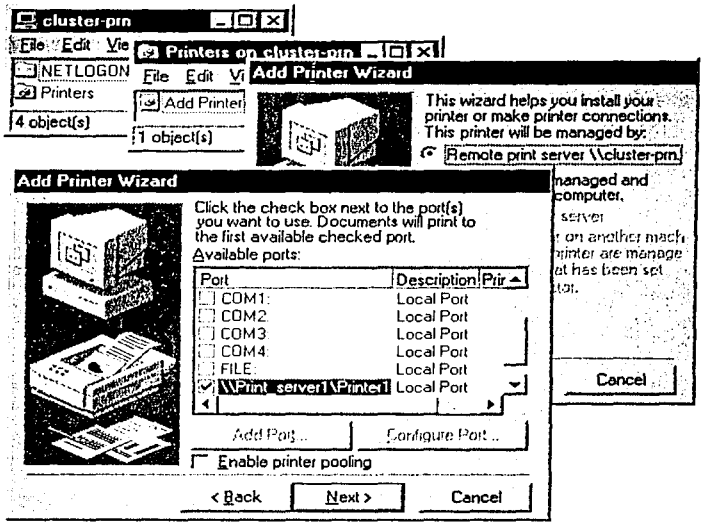


Figura 4.48 - Agregando un *Share* de Impresión.

Para crear un *share* de impresión en un cluster

1. Hay que dar un click en el botón **Start**, y después hay que dar click en **Run**.

2. En la caja de **Open** (en la caja de diálogo de **Run**), hay que introducir el nombre de red que se agregó en el grupo de impresión en el cluster (por ejemplo, \\cluster-prn).

   Aparecerá una ventana del explorador de windows, listando todos los *shares* y el folder de *Printers*.

3. Hay que dar click dos veces sobre el folder de **Printers**

   El folder de *Printer* va a tener un icono del asistente de *Add Printer*.

4. Para agregar el *share* de impresión, hay que dar doble click en el icono del asistente de *Add Printer*.

   Las opciones disponibles en el asistente de *Add Printer* son muy limitadas:
   - En la ventana de diálogo del asistente de **Add Printer**, la única opción es agregar una impresora a un servidor remoto de impresión.
   - En la ventana de diálogo del asistente de **Add Printer**, los botones de **Add Port** y **Configure Port** no se encuentran disponibles.

4.9.2 CONFIGURANDO LOS RECURSOS DE APLICACIONES GENÉRICAS Y DE SERVICIOS

Además de los tipos de recursos que hemos estado discutiendo, MSCS también cuenta con los recursos de tipo DLL (dynamic-link libraries) para aplicaciones genéricas y del tipo de servicios genéricos. Estos tipos de recursos son para aplicaciones y servicios que no son *Cluster Server-aware*. Por lo tanto, si usamos este tipo de recursos vamos a poder realizar operaciones básicas de *failover* sobre aplicaciones y servicios que no son *Cluster Server-aware*.

4.9.2.1 CONFIGURANDO EL RECURSO PARA APLICACIONES GENÉRICAS

El recurso para aplicaciones genéricas es usado para configurar las aplicaciones de los usuarios para que funcionen con el Cluster. La página de parámetros del recurso para aplicaciones genéricas cuenta con las siguientes opciones de configuración:

4. Para agregar el *share* de impresión, hay que dar doble click en el icono del asistente de *Add Printer*.

   Las opciones disponibles en el asistente de *Add Printer* son muy limitadas:
   - En la ventana de diálogo del asistente de **Add Printer**, la única opción es agregar una impresora a un servidor remoto de impresión.
   - En la ventana de diálogo del asistente de **Add Printer**, los botones de **Add Port** y **Configure Port** no se encuentran disponibles.

4.9.2 CONFIGURANDO LOS RECURSOS DE APLICACIONES GENÉRICAS Y DE SERVICIOS

Además de los tipos de recursos que hemos estado discutiendo, MSCS también cuenta con los recursos de tipo DLL (dynamic-link libraries) para aplicaciones genéricas y del tipo de servicios genéricos. Estos tipos de recursos son para aplicaciones y servicios que no son *Cluster Server-aware*. Por lo tanto, si usamos este tipo de recursos vamos a poder realizar operaciones básicas de *failover* sobre aplicaciones y servicios que no son *Cluster Server-aware*.

4.9.2.1 CONFIGURANDO EL RECURSO PARA APLICACIONES GENÉRICAS

El recurso para aplicaciones genéricas es usado para configurar las aplicaciones de los usuarios para que funcionen con el Cluster. La página de parámetros del recurso para aplicaciones genéricas cuenta con las siguientes opciones de configuración:

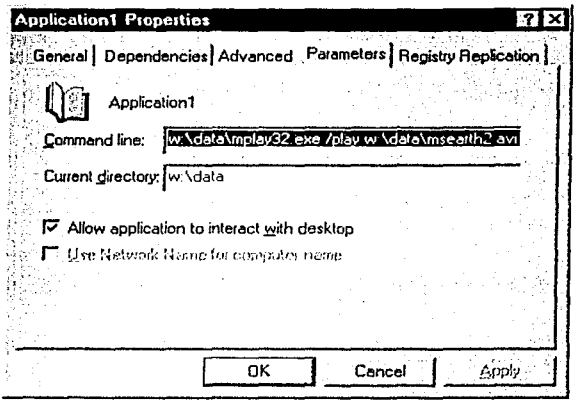


Figura 4.49 - Configurando el Recurso para Aplicaciones Genéricas.

Opción	*Descripción*
Command Line	El comando de línea para la aplicación. La ruta en donde debe de estar localizada sobre un disco del cluster para que así el *failover* tome lugar en caso de que ocurra algún evento.
Current directory	El directorio de trabajo para la aplicación.
Allow application to interact with desktop	Cuando se selecciona esta opción, la aplicación aparecerá sobre el escritorio del nodo en donde se está ejecutando.
Use Network Name for computer name	Cuando se selecciona esta caja, el nombre de red para acceder a esta aplicación va a ser el nombre del nodo.

La página de propiedades de Registry Replication es utilizada para aquellas aplicaciones que guardan información en el *registry* de NT, en particular en la llave de HKEY_LOCAL_MACHINE. Esta pagina de propiedades es usada para especificar las llaves del *registry* que van a ser replicadas entre los nodos para que la aplicación funcione de manera correcta cuando se presente un *failover*.

4.9.2.2 Configurando el Recurso de Servicios Genéricos

El recurso de servicio genéricos es usado para configurar aquellos servicios que se van usar con *Cluster Server*. La página de parámetros del recurso de servicios genéricos cuenta con las siguientes opciones de configuración:

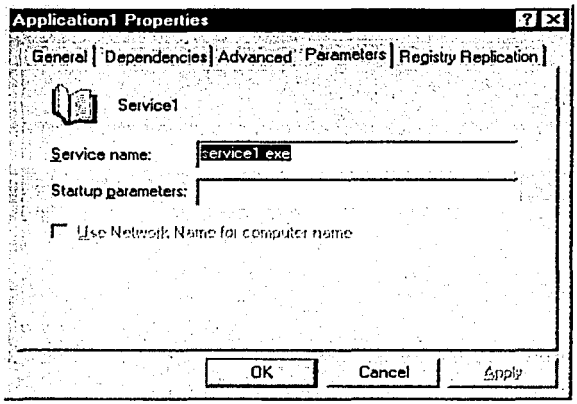


Figura 4.50 - Configurando el Recurso de Servicios Genéricos.

Opción	*Descripción*
Service name	Aquí se debe de introducir el nombre del servicio de Windows NT tal y como aparece en el programa de *Services* en el panel de control.

Startup parameters	Aquí hay que introducir los *switches* o parámetros necesarios para que inicie el servicio.
Use Network Name for computer name	Cuando se selecciona esta caja, el nombre de red para acceder a esta aplicación va a ser el nombre del nodo.

De igual manera que para el recurso de aplicaciones genéricas, la página de propiedades de Registry Replication es utilizada para aquellos servicios que guardan información en el *registry* de NT, en particular en la llave de HKEY_LOCAL_MACHINE. Esta página de propiedades es usada para especificar las llaves del *registry* que van a ser replicadas entre los nodos para que el servicio funcione de manera correcta cuando se presente un *failover*.

4.9.3 Configurando Otras Aplicaciones para que se Ejecuten sobre Cluster Server

Además de los recursos que MSCS soporta, existen muchas otras aplicaciones y servicios que pueden ser configurados para que trabajen con *Cluster Server*. Esto es posible utilizando a los recursos de servicios genéricos y de aplicaciones genéricas, y además del kit de desarrollo de software para *Cluster Server* (*Cluster Server software development kit* - SDK). Este kit de desarrollo permite a los desarrolladores independientes de software creen sus propias librerías dinámicas (DLL's).

4.9.3.1 Configurando los Servicios de Windows NT Server

La siguiente tabla lista los servicios que vienen incluidos con Windows NT Server y si estos pueden o no ser configurados para poder realizar un *failover* sobre un Cluster.

Servicio de Windows NT	*¿Es posible realizar un failover?*
Distributed File System (Dfs)	No es posible realizar un *failover* con este servicio, sin embargo ambos nodos del cluster pueden funcionar sin ninguna limitante como servidores de Dfs
Domain controller	No es posible realizar un *failover* con este servicio, sin embargo ambos nodos del cluster pueden funcionar correctamente como controladores de dominio.
Domain Name System (DNS) Server	No es posible realizar un *failover* con este servicio, sin embargo ambos nodos del cluster pueden funcionar correctamente como servidores de DNS.
Dynamic host configuration protocol (DHCP)	No es posible realizar un *failover* con este servicio, sin embargo ambos nodos del cluster pueden ser servidores de DHCP.
Multiple protocol router (MPR)	No es posible realizar un *failover* con este servicio, sin embargo ambos nodos del cluster puede funcionar como enrutadores.
Remote Acess Service (RAS) server	No es posible realizar un *failover* con este servicio, sin embargo ambos nodos del cluster pueden ser servidores de RAS.
Windows Internet Name Service (WINS)	No es posible realizar un *failover* con este servicio, sin embargo ambos nodos del cluster pueden ser servidores de WINS.

4.9.3.2 CONFIGURANDO APLICACIONES DE MICROSOFT BACKOFFICE

La siguiente tabla lista las aplicaciones de *Microsoft BackOffice* y si éstas pueden o no ser configuradas para poder realizar un *failover* sobre un cluster.

Aplicación de BackOffice	*¿Es posible realizar un failover?*
Microsoft Distributed Transaction Coordinator	Esta aplicación sí puede ser configurada en el cluster. MSCS incluye una DLL para que funcione correctamente.
Microsoft Exchange Server	Esta aplicación sí puede ser configurada para que funcione en cluster a partir de su versión 5.5 Enterprise Edition.
Microsoft Internet Information Server (IIS)	Esta aplicación sí puede ser configurada para que funcione en cluster. MSCS incluye una DLL para que funcione correctamente.
Microsoft Proxy Server	Esta aplicación no puede ser configurada para que funcione con MSCS; sin embargo, es posible que ambos nodos del cluster funcionen como servidores de proxy.
Microsoft Site Server	Esta aplicación no puede ser configurada para que funcione con MSCS; sin embargo, es posible que ambos nodos del cluster puedan ejecutar esta aplicación.
Microsoft Systems Management Server (SMS)	Esta aplicación no puede ser configurada para que funcione con MSCS; sin embargo, es posible que ambos nodos del cluster funcionen como servidores de SMS.
Microsoft SNA Server	Esta aplicación no puede ser configurada para que funcione con MSCS; sin embargo, es posible que

	ambos nodos del cluster funcionen como servidores de SNA
Microsoft SQL Server 6.5 Enterprise	Sí es posible configurar SQL Server 6.5 Enterprise para que funcione en cluster. En los manuales de esta versión viene una guía para poder configurar esta aplicación en cluster.
Microsoft Message Queue Server	Sí es posible configurar esta aplicación para que funcione en cluster. Una librería DLL vienen incluida en MSCS para poder soportar esta aplicación.

4.9.3.3 CONFIGURANDO EL MICROSOFT DISTRIBUTED TRANSACTION COORDINATOR

Microsoft Transaction Server puede ser usado para ejecutar aplicaciones que han sido construidas con componentes de ActiveX. MSDTC entrega los componentes, incluyendo transacciones, servicios de escalabilidad, administración de conexión, y administración "point and click" para proveer a los desarrolladores una manera de construir y desarrollar aplicaciones escalables de servidor para negocios y el Internet.

El recurso de MSDTC no requiere de ningún parámetro al momento de ser configurado. Sin embargo, se requiere crear algunas dependencias sobre un recurso de disco y de un nombre de red.

Para que MSDTC trabaje en cluster, requiere ser instalado en ambos nodos del cluster, usando la misma ruta de instalación. Por ejemplo, si MSDTC es instalado en C:\Mtx en el primer nodo, éste también debe de instalarse en C:\Mtx en el segundo nodo. Una vez que MSDTC es instalado, un recurso de DTC (Distributed Transaction Coordinator) puede ser creado en un grupo con un recurso de disco y un recurso de nombre de red utilizando los siguientes pasos:

1. Crear un nuevo recurso del tipo *Distributed Transaction Coordinator*, con el nombre y la descripción de MSDTC y apretar el botón de *Next*.

2. Seleccionar todos los nodos que pueden ser dueños de este recurso y apretar el botón de *Next*.

3. Seleccionar un recurso de disco y un recurso de nombre de red como dependencias, y dar click en el botón de *Finish*.

4.9.3.4 CONFIGURANDO MICROSOFT QUEUE SERVER

Microsoft Queue Server es una parte importante de la plataforma *Microsoft Active Server*. MSQS provee a ciertas aplicaciones una manera de comunicarse con otras aplicaciones sobre la red enviando y recibiendo mensajes. Los mensajes de MSQS pueden contener datos en cualquier formato que únicamente son compatibles con entre la aplicación mensajera y la aplicación receptora. Cuando una aplicación recibe un mensaje de petición, ésta la procesa leyendo el contenido del mensaje y realiza un operación según haya sido el mensaje. Si se requiere, la aplicación receptora puede enviar un mensaje de acuse de recibo a la aplicación que origino la petición.

Cuando se está instalado MSQS en un cluster éste automáticamente crea un recurso de MSQS y un recurso de MSDTC si es que este recurso no existiera. El recurso de MSQS no requiere de configurar ningún parámetro, pero sin embargo es necesario crear algunas dependencias sobre un recurso de disco, un recurso de nombre de red, un recurso SQL Server y un recurso de MSDTC.

Durante el proceso de instalación de MSQS el recurso de este tipo es creado de manera automática y también configura la dependencia sobre un recurso de disco pero no crea la dependencia sobre el recurso de red; por lo que esta dependencia debe de ser creada manualmente una vez que el proceso de instalación haya finalizado. Si durante el proceso de instalación el recurso de MSDTC es creado, entonces el proceso de instalación crea la dependencia sobre el recurso de disco pero no sobre el recurso de nombre de red, por lo que esta dependencia debe de ser creada una vez que el proceso de instalación haya terminado.

4.10 SOLUCIÓN DE PROBLEMAS EN MICROSOFT CLUSTER SERVER

En todo sistema de cómputo siempre pueden aparecer problemas de software o de algún servicio, y aunque una configuración de cluster se busca lograr contar con alta disponibilidad en los sistemas, éstos no están exentos de aquellos. Estos problemas pueden ser causados por mantenimientos generales, configuraciones de dispositivos *SCSI* (small computer system interface), instalaciones, falla de recursos, o problemas de red. Este módulo examina posibles problemas y soluciones para estas áreas.

4.10.1 CÓMO DAR MANTENIMIENTO A UN CLUSTER

El darle mantenimiento a los nodos que están ejecutando MSCS es bastante parecido a darle mantenimiento a una computadora corriendo Windows NT Server. Las diferencias que existen entre darle mantenimiento a un nodo de un cluster y darle mantenimiento a un servidor de *Windows NT Stand-Alone*, se listan a continuación:

- No se debe de cambiar el nombre del servidor bajo ninguna condición una vez que ya se haya instalado el software de MSCS.
- No se debe reparticionar los discos que se encuentran en el bus *SCSI* compartido sin haber removido el recurso de disco primero.
- Después de haber reparticionado cualquier disco físico sobre el bus *SCSI* compartido, hay que reiniciar ambos nodos del cluster.
- No se debe de cambiar la dirección de IP de cualquier recurso de IP, una vez que exista una dependencia de nombre de red sobre esta dirección.
- No se debe de cambiar la letra de *drive* asignada a los discos de sistema de los nodos.

Otras funciones de mantenimiento incluye el otorgar permisos para la administración del cluster y ser capaz de administrar el cluster desde la línea de comandos.

4.10.1.1 OTORGANDO PERMISOS PARA ADMINISTRAR AL CLUSTER

La herramienta de Cluster Administrator puede ser usada para otorgar permisos a usuarios o a grupos de usuarios para que éstos puedan administrar el cluster. Por defecto el grupo de *Administrators* cuenta con permisos de *Full Control* para poder administrar el cluster.

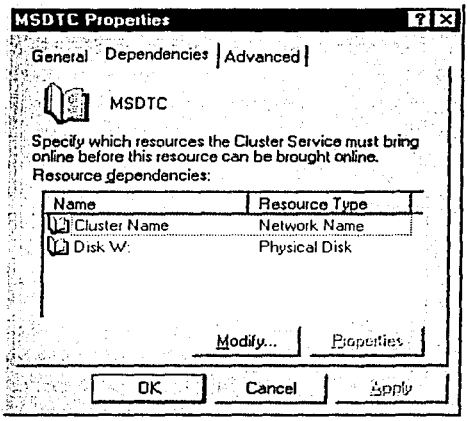


Figura 4.51 - Otorgando Permisos para Administrar al Cluster.

Para cambiar los permisos, hay que usar el botón de *Permissions* en la página de propiedades **Generales** del cluster. Para acceder a la página de propiedades **Generales** del cluster, en la herramienta de *Cluster Administrator*, hay que dar click sobre el nombre del cluster, y después hay que dar un click derecho sobre el nombre del cluster, y después hay que dar click en la opción de **Properties**; o sobre el menú de **File** hay que dar click sobre **Properties**.

Una vez que aparece la ventana de **Cluster Access Permissions**, ésta puede ser usada para otorgar permisos en la misma manera en como se otorgan permisos en otras herramientas de Windows NT, como por ejemplo el dar permisos sobre un

archivo bajo el explorador de Windows NT. Existen dos diferencias entre otorgar permisos sobre el cluster y otros permisos en Windows NT, los cuales se describen en la siguiente tabla:

Diferencias	*Descripción*
Se pueden otorgar dos tipos de accesos	La herramienta de *Cluster Administrator* cuenta con dos *tipos de acceso, los cuales son Full* Control o *No Access*. El permiso de *Full Control* permite al usuario con este permiso el ejecutar todas las tareas administrativas sobre el cluster. El permiso de *No Access* negará al usuario el acceso al cluster cuando éste desee administrarlo con la herramienta de *Cluster Administrator*
Permisos para administra el cluster	Los permisos otorgados en la ventana de *Cluster Access Permissions* no afectarán a los clientes que están siendo usados para acceder al cluster y a cualquiera de sus recursos disponibles. El permiso otorgado en esta ventana únicamente tiene efecto en los permisos de administración del cluster.

Los permisos otorgados para administrar el cluster se almacenan en la siguiente llave dentro del *registry* HKEY_LOCAL_MACHINE\Cluster.

4.10.1.2 ADMINISTRANDO CLUSTERS DESDE LA LÍNEA DE COMANDOS

Además de la herramienta de *Cluster Administrator*, MSCS incluye una herramienta de comandos de línea, Cluster.exe, la cual puede ser utilizada para administrar clusters. Dado que Cluster.exe es un comando de línea, éste puede ser usado en scripts y archivos batch para automatizar tareas de administración comunes entre los posibles clusters que se encuentren dentro de la red. Esta utilería es instalada de manera automática en el subdirectorio *\windir*\Cluster cuando el servicio de *Cluster Server* es instalado en un nodo o cuando se instala la herramienta de *Cluster Administrator*. Cluster.exe cuenta con los siguientes parámetros:

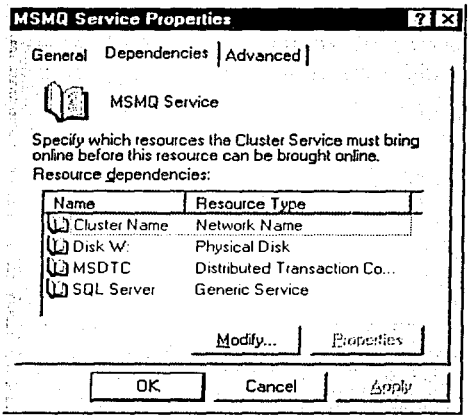


Figura 4.52 - Administrando Clusters desde la Línea de Comandos.

Comando	*Permite al administrador realizar*
Cluster	Cambia el nombre del cluster o del recurso de quórum y checa la versión de *Cluster Server*.
Cluster node	Verifica el estado de un nodo y realiza algunas tareas de administración, tales como pausar un nodo.
Cluster group	Verifica el estado de los grupos y realiza tareas comunes

	de administración, como por ejemplo el crear o borrar grupos.
Cluster resource	Verifica el estado de los recursos y realiza tareas comunes de administración, tales como el poner un recurso fuera de línea o en línea
Cluster resourcetype	Muestra las propiedades de un recurso en específico.
Cluster network	Verifica el estado y las propiedades de las redes del cluster.
Cluster interface	Verifica el estado y las propiedades de las interfaces de red del cluster.

Cuando se utilizan nombres de grupos y recursos con este comando, y estos nombres cuentan con espacios, entonces el nombre debe de escribirse entre comillas para que éstos puedan ser administrados.

4.10.2 SOLUCIÓN DE PROBLEMAS

El primer paso a analizar para solucionar algún problema con un cluster es verificar si el hardware utilizado se encuentra en la Lista de Compatibilidad de Hardware de *Cluster Server* (HCL – Hardware Compatibility List). La última versión de esta lista puede ser encontrada en http://www.microsoft.com/ lanzando la búsqueda de ***Cluster Server Hardware Compatibility List***.

Una vez que se ha verificado que todo el hardware que se está usando en el cluster está soportado, los problemas típicamente recaen en las siguientes categorías:

- Configuraciones *SCSI*
- Instalación
- Fallas en los recurso y grupos del cluster
- Comunicaciones de red, entre los nodos, y entre los clientes y el cluster.

Existen varias herramientas que se encuentran incluidas en Windows NT que pueden ser utilizadas para diagnosticar los problemas de *Cluster Server*, dentro de estas herramientas también se encuentra la habilidad de realizar un diagnóstico mediante un archivo de *log*.

4.10.2.1 Herramientas de Diagnostico de Windows NT Enterprise

La versión empresarial de Windows NT incluye varias herramientas de diagnóstico, las cuales se enlistan en la siguiente tabla. Estas herramientas pueden ser utilizadas para diagnosticar posibles fallas en el Cluster:

Herramienta	*Utilizar Para*
Control Panel Services	Sirve para verificar si el servicio de Cluster se encuentra funcionando.
Disk Administrator	Determina si un disco se encuentra disponible en un nodo en particular. Si el disco puede ser seleccionado en esta herramienta, entonces este disco se encuentra en línea en ese sistema. Si este disco aparece sombreado, entonces este disco no se encuentra disponible en el nodo.
Dr. Watson	Detecta y lleva un registro de los errores de las aplicaciones.
Event Viewer	Nos muestra los eventos registrados del sistema, seguridad y aplicaciones.
Network Monitor	Monitorea y diagnostica los problemas de red, capturando y analizando el tráfico de la red.
Performance Monitor	Monitorea el desempeño de las aplicaciones y al sistema.
Task Manager	Monitorea aplicaciones, tareas, nos muestra métricas de desempeño y visualiza información detallada sobre el uso de memoria y CPU para cada aplicación y proceso.

Windows NT Diagnostics (WinMSD)	Examina la configuración del sistema, nos muestra datos como qué *drivers* de dispositivos se encuentran activos, uso de la red, y de los recursos de sistema (IRQ, DMA y direcciones I/O).

4.10.2.2 Bitácora de Cluster Server (Servicio de Logging)

Además de los mensajes que *Cluster Server* registra en el *Event Viewer*, *Cluster Server* también cuenta con la habilidad de llevar una bitácora de diagnóstico. Para habilitar esta bitácora, se tiene que crear la siguiente variable de sistema CLUSTERLOG, esta variable debe de tener como valor una ruta y nombre del archivo en donde se va a encontrar ubicado el archivo de *log*. Para crear esta variable hay que usar el programa de System dentro del Panel de Control.

El archivo de *log* que es creado después de darle valor a la variable es útil para encontrar más información acerca de los problemas que pueden ocurrir al iniciar un nodo, como por ejemplo:

- Detectar si el servicio de Cluster está fallando al iniciar sobre un nodo.
- Detectar si un nodo es incapaz de poner un recurso en línea.

Cuando se vaya a utilizar la variable CLUSTERLOG, hay que tomar en cuenta lo siguiente:

- La variable CLUSTERLOG no es case sensitive (sensible a mayúsculas).
- La variable CLUSTERLOG puede apuntar a cualquier ruta y nombre de archivo, sin importar la longitud; por ejemplo esta variable puede tener como ruta C:\Cluster.log o C:\Cluster\Cluster.log.

- Una vez que se creó esta variable, el nodo debe de ser reinicializado para que se comience a llevar la bitácora.

- Para que se lleve una bitácora más exacta hay que crear esta variable en ambos nodos del cluster.

- El archivo de *log* se sobre escribe cada vez que el nodo es reiniciado.

El archivo de *log* por defecto puede alcanzar un tamaño máximo de 8 MB. Cuando el archivo de *log* alcanza el tamaño de 8 MB los datos en este archivo se comenzaran a sobre escribir. Para especificar un tamaño mayor para el archivo de *log*, hay que crear la llave ClusterLogSize en el *registry* del nodo, en la siguiente ruta **\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ClusSvc\ Parameters. ClusterLogSize** debe de ser del tipo **DWORD** y como valor se le debe de asignar el tamaño máximo que puede alcanzar el archivo de *log* (MB). Si se le asigna el valor de 0, entonces se deshabilitara la bitácora.

4.10.3 DIAGNOSTICANDO CONEXIONES SCSI

Cuando se está creando un cluster, una posible área de problemas puede ser la configuración del hardware del bus *SCSI* compartido. Dado que ambos nodos del cluster necesitan tener acceso al bus *SCSI* compartido, *Cluster Server* usa el *"Multiple Initiator SCSI bus topology"*, como se define en las especificaciones de *SCSI*. Sin embargo, el *Multiple Initiator SCSI bus topology* no esta soportada por la mayoría de las tarjetas controladoras *SCSI*, por lo que no todas las tarjetas *SCSI* pueden ser usadas para crear un cluster.

Los posibles problemas que podemos encontrar al estar configurando el bus *SCSI* compartido, pueden ser divididos en tres áreas, las cuales son:

- Controladores *SCSI*

- Terminación del Bus *SCSI*

- Cableado del Bus *SCSI*

Las especificaciones *SCSI* pueden ser solicitadas al *American National Standards Institute* (ANSI). Su sitio de web (http://www.ansi.org/) contiene un catálogo en donde pueden buscarse las especificaciones de *SCSI*.

4.10.3.1 CONTROLADORES SCSI

Esta sección describe los problemas más comunes que se encuentran cuando se configuran controladores *SCSI*.

Cada dispositivo que se encuentra en el bus *SCSI* compartido debe de tener un *SCSI ID* único. La mayoría de los controladores *SCSI* tienen por defecto el *ID* 7, por lo que en la mayoría de las veces se tiene que cambiar el *SCSI ID* de uno de los controladores *SCSI* por otro valor que no sea 7 (se recomienda que se cambie por el *SCSI ID* 6). Debido a que existen 2 controladores *SCSI* en bus *SCSI* compartido, y éstas usan dos de los ocho *ID*'s, únicamente podremos tener hasta 6 dispositivos sobre el bus.

Cuando se están iniciando los sistemas, algunos controladores *SCSI* "limpian" el bus *SCSI*. Si esto ocurre, esta "inicialización" de bus puede interrumpir cualquier transferencia de datos entre el otro nodo y los dispositivos del bus *SCSI* compartido. Por lo que, esta opción debe de ser deshabilitada de ser posible; no todas las tarjetas permiten deshabilitar esta opción.

Es muy importante verificar que los controladores *SCSI* que se están usando en el Cluster se encuentren dentro de la lista de compatibilidad de Hardware. Para que un controlador *SCSI* trabaje con *Cluster Server*, es importante que soporte:

- "Multiple initiator on the bus"
- El comando *SCSI* **reserve**
- El comando *SCSI* **release**

4.10.3.2 TERMINACIÓN DEL BUS SCSI

En esta sección se describirán los posibles problemas que se pueden presentar en la terminación del bus *SCSI*.

Existen tres posibles formas de terminar un bus *SCSI*: terminación pasiva, terminación activa, y terminación "*forced perfect*" (perfectamente forzada). Dado que tanto la terminación activa como la forzada utilizan medios electrónicos para realizar la terminación, estas formas de terminación son las más recomendadas. La terminación pasiva usa un conjunto de resistencias, las cuales no proveen una muy buena terminación. Por lo tanto se recomienda que la terminación pasiva no se use en un cluster, por que este tipo de terminación puede causar problemas al momento de que se realice un *failover* o inhabilitar el acceso al disco de quórum.

Ahora, muchas controladoras proveen terminación integrada en la misma tarjeta; sin embargo, este tipo de terminación no provee esta característica cuando la computadora se encuentra apagada, por lo que se recomienda no utilizarla. Esta característica debe de ser deshabilitada cuando se esté utilizando terminación externa.

La tarjeta *SCSI* Adaptec 2940 es un controlador *SCSI* que provee terminación integrada, para habilitar esta opción basta mover el jumper JP4 sobre la tarjeta, ésta es de las pocas tarjetas que proveen terminación aún si se encuentra la computadora apagada. Para el controlador *SCSI* Adaptec 2944, existe un método diferente de terminación el cual provee esta aún si el servidor está apagado y sin la necesidad de mover ningún jumper.

4.10.3.3 CABLEADO DEL BUS SCSI

En esta sección se describirán los problemas que se pueden presentar cuando se esta cableando el bus *SCSI* compartido.

El conectar cables "Y" en los controladores *SCSI* es un método que puede utilizarse para mantener el bus *SCSI* compartido terminado aún si un nodo del cluster está apagado. El utilizar este tipo de cables permite la utilización de

terminadores externos, los cuales proveen terminación sobre el bus aún si un nodo del cluster está apagado o simplemente es desconectado del bus.

Es muy común el tener varios discos *SCSI* externos sobre el bus *SCSI* compartido. Cuando se está configurando de esta forma, es muy importante el no exceder la longitud máxima recomendada que puede tener el bus *SCSI* compartido, porque pueden existir problemas de acceso a los discos.

4.10.4 Diagnosticando Fallas del Archivo de Log del Cluster

Si por alguna razón el archivo de *log* del quórum se corrompe o el recurso de quórum sufre de una corrupción sobre el disco, *Cluster Server* intenta corregir el problema reinicializando el archivo de *log*. Si esto ocurre, un mensaje de *ERROR_CLUSTERLOG_CORRUPT* es registrado en la bitácora de eventos del sistema con la siguiente descripción:

The log file [name] was found to be corrupt. An attempt will be made to reset it.

Si el archivo de *log* no puede ser reiniciado, el servicio de Cluster no va a poder iniciar. Si el mensaje anterior está registrado y el servicio de Cluster no esta iniciado, el administrador del cluster va a tener que arreglar al recurso de quórum o al archivo de *log*. Los pasos que se enlistan a continuación pueden utilizarse para recuperarnos de cualquiera de estos eventos.

Para recuperarse de un recurso de quórum o archivo de log corrupto:

1. En el panel de control, hay que ejecutar la aplicación de *Services*, y después hay que parar el servicio de Cluster en ambos nodos.

2. En la ventana de diálogo de *Services*, seleccionado el servicio de Cluster, hay que teclear **-noquorumlogging**, en la caja de *Startup Parameter*. Esto hay que realizarlo en ambos nodos del cluster.

3. Iniciar el servicio de cluster en ambos nodos.

4. Hay que ejecutar la utilería de Chkdsk sobre el disco de quórum, y después hay que solucionar cualquier error reportado.

⇒ Si no se reporta error alguno, lo más probable es que el archivo de *log* es el que se encuentre corrupto. Para solucionar esto, sobre el recurso de quórum hay que borrar el archivo \MSCS\Quolog.log y los archivos \MSCS*.tmp.

5. En la ventana de diálogo de *Services*, hay que parar el servicio de Cluster en ambos nodos del cluster, y después hay que remover el parámetro, y después reiniciar el servicio de Cluster.

Hay que tener cuidado al realizar este tipo de operaciones, por que el configurar a *Cluster Server* para que se ejecute sin que exista un archivo de log, puede causar que los cambios más recientes realizados en el cluster se pierdan. Esto pasaría si uno de los nodos se encuentra fuera de línea, la configuración del cluster cambia, y después el archivo de *log* se corrompe antes de que los cambios sean comunicados al nodo que se encontraba fuera de línea.

4.10.5 Diagnosticando Fallas en el Proceso de Instalación

La siguiente tabla enlista los problemas que se pueden presentar durante el proceso de instalación de *Cluster Server* y provee la posible solución a estos.

Posible Problema	*Solución*
La cuenta de servicio del Cluster no puede ser validada.	Hay que verificar que esta cuenta ha sido creada. Hay que verificar que el password de la cuenta es correcto.
Los discos del bus *SCSI* compartido no están visible durante el proceso de instalación.	Hay que verificar que el hardware utilizado en el bus *SCSI* compartido se encuentre dentro de la lista de compatibilidad de hardware. Hay que verificar que la configuración de la tarjeta *SCSI* es correcta, el bus se encuentra correctamente terminado, y si la longitud de los cables *SCSI* utilizados cumplen con las especificaciones del protocolo.
Errores de nombres de red duplicados cuando se	Hay que utilizar un nombre único NetBIOS para el nombre del cluster.

esta instalando el primer nodo del cluster.	
Mensajes de error de que no es posible conectarse al cluster, cuando se esta instalando el segundo nodo del Cluster.	Hay que verificar que el servicio de Cluster se encuentra iniciado en el primer nodo. Hay que verificar que el grupo en donde se encuentra el nombre de red del cluster este en línea. Hay que verificar que el segundo nodo pueda resolver el nombre de red del cluster. Hay que verificar que el segundo nodo puede comunicarse a través de la red con el primer nodo.
Hardware no soportado.	Hay que verificar que el hardware utilizado en los nodos del cluster para construir el bus *SCSI* compartido se encuentra en la lista de compatibilidad de hardware.

4.10.6 DIAGNOSTICANDO FALLAS EN LOS GRUPOS Y RECURSOS DEL CLUSTER

La siguiente tabla enlista los problemas más comunes y sus soluciones para cuando se presentan fallas en grupos y recursos del Cluster.

Posible Problema	*Solución*
Un recurso falla, y esto no es puesto en línea otra vez.	Hay que verificar que la opción de Don't restart se encuentre no seleccionada en la ventana de Policies en la propiedades del recurso. Hay que verificar que las dependencias del recurso estén bien configuradas.
El recurso de quórum no está en línea.	Hay que verificar que no existen errores de hardware (mensajes de error de I/O) , utilizando el *Event Viewer*.

	Hay que verificar que el hardware utilizado para implementar el bus *SCSI* compartido se encuentra dentro de la lista de compatibilidad. Hay que verificar que los controladores *SCSI* sean correctos, el bus se encuentre bien terminado, y la longitud de los cables estén dentro de las especificaciones. Hay que verificar que cada componente *SCSI* sobre el bus *SCSI* compartido cuente con un *SCSI ID* único.
No se puede poner un grupo en línea	Hay que verificar que no existan problemas de hardware. Hay que verificar que las dependencias de los recursos se encuentran correctamente configuradas. Hay que intentar mover el grupo al otro nodo e intentar ponerlo en línea. Si esto funciona, hay que verificar que el primer nodo puede tener acceso a todo lo que sea necesario para poner a los recursos de este grupo en línea.
Un grupo no puede moverse o realizar una operación de *failover* al otro nodo del cluster.	Hay que verificar que el otro nodo está designado como un posible dueño para todos los recursos del cluster en el grupo. Hay que verificar que el otro nodo esté designado como un posible dueño del grupo.
Un grupo realiza operaciones de *failover* pero no de *failback*.	Hay que verificar se las políticas de *failback* en el grupo se encuentran configuradas. Hay que verificar si la opción de **Prevent failback** no se encuentra seleccionada para el grupo. Si la opción de **Failback immediately** se encuentra seleccionada, hay que asegurarnos de que estamos esperando lo suficiente para que el grupo

	realice la operación de *failback*. Hay que checar esta configuración para todos los recursos dentro del grupo. Por que con un recurso que no cuente con esta política, es suficiente para que afecte a todo el grupo y evite que se realicen operaciones de *failback*. Hay que asegurarnos de que el nodo para el cual deseamos se realice una operación de *failback*, esté configurado como "dueño preferido" (preferred owner) del grupo y de todos los recursos del grupo. Si esta opción no está configurada *Cluster Server* dejará al grupo en el nodo en donde realizó la tarea de *failback*.
Todo el grupo falló y no se reinició.	Si el nodo en el cual está corriendo el grupo es puesto fuera de línea, hay que verificar que el otro nodo tiene los permisos de poder tener a este grupo y a todos los recursos del grupo. Hay que asegurarse de que el grupo no ha excedido su *failover threshold* o su *failover period*. Trata de poner en línea a los recursos uno por uno con el fin de determinar qué recurso es el que está causando el problema. Crea un grupo temporal (con fines de prueba), y después hay que mover a los recursos uno por uno, y posteriormente hay que poner a cada recurso en línea después de que se haya movido.

4.10.7 DIAGNOSTICANDO FALLAS EN LA COMUNICACIÓN DE RED DEL CLUSTER

Cuando se utiliza *Cluster Server*, existen varias áreas por las que pueden existir problemas de comunicación a través de la red:

- Comunicación hacia los clientes.
- Problemas al realizar una operación de *failover* de *IP Address*.
- Problemas para resolver un nombre de red y registrarlo en la red después de una operación de *failover*.
- Problemas en la comunicación interna entre los nodos.

Conociendo cómo los dos tipos de comunicación que hay sobre la red con *Cluster Server* trabaja, y cómo se puede verificar si éstas están funcionando, nos provee del conocimiento necesario para diagnosticar los problemas de comunicación.

4.10.7.1 COMUNICACIÓN HACIA LOS CLIENTES

El acceso que tienen los clientes a un cluster es exactamente igual al acceso de red que tienen a cualquier otra computadora corriendo Windows NT Server, únicamente hay que utilizar el nombre de red, de preferencia el nombre NetBIOS o el nombre de la computadora. La conexión de los clientes a un cluster es transparente al cliente y al usuario. Si se utiliza un analizador de tráfico de red, para observar el tráfico entre un cliente y un cluster, este nos mostrará que no existe diferencia alguna con el tráfico de red que se puede observar cuando un cliente entabla comunicación con una computadora corriendo Windows NT Server, al menos que se realice una operación de *failover* durante la captura.

Como resultado, los problemas de comunicación entre un cliente y un cluster son ocasionados en la mayoría de las veces por que se realiza una operación de *failover* de cualquiera de los siguientes tipos:

- Un *failover* de una dirección de IP.
- Se intenta resolver un nombre y registrarlo después de una operación de *failover*.

4.10.7.2 Problemas al Realizar una Operación de Failover de IP Address

El proceso de "*IP address failover*", es el proceso de mover una dirección de IP de un nodo al otro nodo del cluster, la habilidad de *Cluster Server* para mover una dirección de IP es posible debido a dos características de Windows NT:

- El registro y la eliminación dinámica de direcciones de IP.
- La habilidad de actualizar la dirección de IP en la *physcal network address translation caches* (address resolution protocol [ARP] caches) de otros sistemas conectados a la misma *subnet*.

Si un nodo falla y el otro nodo es incapaz de poner la dirección de IP en línea, los clientes no serán capaces de tener acceso al cluster nuevamente.

4.10.7.3 Resolviendo Nombres de Red y Registrándolo en la Red Después de una Operación de Failover

Después de que se realiza un *failover*, un cliente debe de ser capaz de tener acceso al cluster de manera transparente, sin importar si estos van a acceder a un nodo diferente. Con el fin de que esto sea transparente, el cliente debe de ser capaz de resolver cualquier nombre de red que tenga el cluster, de tal manera que el cliente se pueda conectar al nodo en donde los recursos se encuentren en línea.

Debido a esto, no se deben de crear direccionamientos estáticos de direcciones IP para cualquier nombre del cluster dentro de una base de datos de un servidor de Windows Internet Naming Service (WINS). Un servidor WINS es el único método de resolución de nombres que puede causar problemas cuando se utilizan direccionamientos estáticos, debido a que un direccionamiento estático en un servidor WINS usa la dirección física (MAC Address) de la tarjeta como parte del direccionamiento estático.

Si un direccionamiento estático es creado, el nodo para el cual este registro fue creado va a ser capaz de poner el recurso de nombre de red en línea y por lo tanto los clientes va a ser capaces de poderse conectar. Sin embargo, si llegase a presentase un *failover*, el segundo nodo en el cluster va a ser capaz de poner el

recurso de *IP address* en línea, pero no así al recurso de *network name*. Cuando el segundo nodo intenta poner en línea al recurso de network name, el servidor de WINS regresará un error evitando así que se registre el nombre de red. Esto pasa por que el segundo nodo no tiene la misma dirección física que se registró en el direccionamiento dinámico del nombre de red.

4.10.7.4 VERIFICANDO LA COMUNICACIÓN INTERNA ENTRE LOS NODOS

Existen dos tipos de comunicación entre los nodos utilizando la red, una de ellas existe cuando *Cluster Server* utiliza RPC (Remote Procedure Call) y la otra es la comunicación de tipo *heartbeats*.

Para verificar si existe comunicación de tipo RPC entre los nodos de un cluster, hay que usar una utilería de captura de tráfico de red, como por ejemplo Microsoft Network Monitor.

Para verificar que no hay problemas en las comunicaciones RPC, hay que configurar a la utilería para que capture todo el tráfico entre los nodos de un cluster. Una vez que la captura de datos ha comenzado, hay que utilizar la herramienta de *Cluster Administrator* para crear un grupo para así generar algo de tráfico del tipo RFC entre los nodos.

De la misma manera podemos aplicar el método anterior para verificar que la comunicación de tipo *heartbeat* existe entre los nodos. Para verificar que esta comunicación está ocurriendo hay que estar seguros de que el servicio de Cluster está funcionando en ambos nodos del cluster. Una vez que se verificó esto, hay que comenzar la captura de tráfico entre los nodos revisando todo los *UDP frames*, que son los "*heartbeats*" del cluster.

CONCLUSIONES

De esta tesis podemos concluir que cuando los sistemas que se encuentran en cualquier organización están fuera de servicio (Downtime), los costos pueden ser devastadores, se pueden perder cientos de oportunidades, ingresos por posibles ventas, nos podemos hacer acreedores a multas por incumplimiento, etc.

A todo esto hay que sumarle el daño que se genera a la compañía por pérdida de confianza por parte de nuestros clientes, socios de negocios, y proveedores que son afectados por tener fuera el sistema, lo que puede generar un sentimiento de insatisfacción hacia sus necesidades.

Así, el punto de partida para cualquier discusión sobre Alta Disponibilidad tiene que ser el costo generado por el tiempo que está fuera de servicio el sistema de información de su organización. El costo es más alto mientras más es indispensable tener el sistema funcionando, es muy fácil justificar un esquema de alta disponibilidad en los sistemas de cómputo si se tiene identificado este costo.

Para determinar el tipo de sistema que brinde alta disponibilidad en los sistemas de cómputo de una organización se deben de considerar los siguientes puntos:

1. Obtener el costo de tener fuera de servicio los sistemas de cómputo.
2. Determinar cuánto es el tiempo máximo que un sistema puede estar fuera de servicio.
3. Determinar qué posibles eventos pueden ocasionar que el sistema falle.
4. Determinar qué otras partes de la organización se pueden ver afectadas cuando el sistema falla.
5. Una vez que se tienen identificados estos puntos, entonces proceder a elegir la tecnología necesaria para conseguir los niveles de disponibilidad que mi organización requiere.

Algo importante que se nota dentro de muchas organizaciones es que subestiman el costo de tener fuera de servicio sus sistemas de cómputo o el impacto que éstos generan dentro de su organización. Basta ver el siguiente estudio de *Gartner Group* para observar el costo que pagan ciertas industrias cuando sus sistemas de cómputo fallan.

Industria	Aplicación	Costo promedio por hora que el Sistema está fuera de servicio (USD)
Sector Financiero	Operaciones de Bolsa	$6,500,000
Sector Financiero	Sistemas de Tarjetas de Crédito	$2,600,000
Medios	Pagos por Evento	$1,150,000
Retail	Compras por TV	$ 113,000
Retail	Ventas por Catalogo	$ 90,000
Sector de Transportes	Sistema de Reservación de las Aerolíneas	$ 89,500

Para medir el impacto de *Downtime*, hay que hacerse la siguiente pregunta para así medir el nivel de disponibilidad que una organización podría necesitar.

¿Quién y qué es lo que se afecta cuando un sistema falla?

Los procesos: Los procesos vitales de nuestra organización pueden interrumpirse, perderse o corromperse. Tales procesos podrían incluir el manejo de pedidos, inventarios, los estados financieros, operaciones, líneas de producción, recursos humanos, sistemas médicos y de emergencias, operaciones de ATM (Automatic Telephone Machine), y más.

Los programas: Los programas de corto o largo plazo pueden verse afectados. Por que las actividades de nuestros empleados o socios de negocios podrían perderse.

El negocio: En esta época del comercio electrónico, si nuestros clientes o posibles clientes no pueden acceder a nuestro sitio, lo que podría pasar es que nuestros clientes se vayan con nuestra competencia perdiéndolos así para siempre.

Las personas: Pueden perderse vidas; los beneficios a empleados pueden alterarse ocasionando así un posible daño moral; problemas con programas gubernamentales que podrían afectar a la ciudadanía.

Los proyectos: Ciento de miles de horas-hombre pueden perderse, afectando así las fechas límite o podrían saltarse algunas etapas trayendo consigo el fracaso del proyecto.

Las pérdidas no únicamente pueden medirse en dinero. Pero el dinero es la medida en la que los altos ejecutivos entienden estos costos. En un reciente estudio, el *Standish Group* informa que el costo de *downtime* varía en el rango de $1,000 a $27,000 USD por minuto. Más aún, ellos informan que en algunos casos, el costo de *downtime* puede exceder los $10,000,000USD, y si consideramos las estimaciones del *Gartner Group*, los costos pueden alcanzar Billones. Es por todo esto que todas las organizaciones deben de pensar, en realidad cuál es su costo de *downtime.*

La alta disponibilidad puede tener diferentes significados dependiendo de la organización. Para grandes corporativos podría significar disponibilidad continua o "Nonstop Computing", en otras palabras significa tener sus sistemas de cómputo funcionando 99.999% del tiempo en el año, lo cual es aproximadamente unos cinco minutos por año que un sistema va estar fuera de servicio. ¿Pero cuál es su definición de disponibilidad alta? Quizás usted no necesita los " cinco-nueves" pero usted va a intentar estar tan cerca de ellos como pueda. Puede que su requisito no sea contar con una disponibilidad de 24 horas al día los 365 días del año, pero usted puede requerir que cuando su sistema está en operación éste no falle. Un sistema de monitoreo de aerotransporte y un sistema de adquisiciones pueden bien soportar estar fuera de servicio 8 horas al año. O un sistema de cadena de tiendas que realiza 90% de sus ventas durante los periodos festivos puede soportar que su sistema falle por semanas o meses. Cada tipo de disponibilidad puede exigir requisitos muy diferentes.

Para medir qué tipo de disponibilidad su organización necesita, usted debe de preguntarse: ¿necesita de una recuperación rápida, o una recuperación que lo deje en el punto exacto antes de presentarse la falla... o ambos? ¿Qué pasaría si usted no reanuda su servicio en el punto exacto en donde ocurrió la falla? ¿Sería esto inoportuno? ¿Dañino? O ¿Catastrófico? ¿Cuál es el método más efectivo y eficiente para recuperar la información? ¿Qué pasaría si usted no logra reanudar sus

procesos dentro del segundo siguiente? ¿Sería esto inoportuno? ¿Dañino? O ¿Catastrófico?

Una vez que hayamos identificado cuál es nuestro costo de *downtime* y cuáles son los niveles de disponibilidad que mi empresa necesita, el siguiente paso que requerimos analizar son las causas que pueden generar que mis sistemas dejen de brindar su servicio.

Las causas más comunes que pueden ocasionar que un sistema de cómputo falle son:

- Fallas por hardware, software o cuestiones de interoperabilidad.
- Intervenciones Administrativas
- Incidentes de las instalaciones (Incendios, fallas en el suministro eléctrico, etc.)
- Desastres (Terremotos, inundaciones, etc.)

Una vez que ya se cubrieron estos puntos entonces hay que pasar a analizar las siguientes áreas de tecnología:

- Hardware
- Sistema Operativo
- Almacenamiento
- Bases de Datos
- Redes
- Administración
- Aplicaciones

La manera de atacar las posibles fallas ocasionadas por el hardware es poniendo componentes redundantes; una manera para atacar las posibles fallas ocasionadas por el sistema operativo o por la base de datos o cualquier aplicación es utilizando un cluster.

El cual no sólo me va a brindar alta disponibilidad si no también me va a dotar de características de alto desempeño, escalabilidad, rentabilidad y fácil administración.

BIBLIOGRAFÍA

***** Configuración, actualización y mantenimiento
SOFTWARE Y HARDWARE de PC.
José A. Carballar Falcón.
Addison-Wesley Iberoamericana.
1994.
pag. 304, 306.

***** Todo sobre Multimedia
Winn L. Rosch
Prentice Hall Hispanoamericana, S.A.
1996.
pag. 188, 345-348.

***** Windows NT Server Professional Reference.
Karanjit S. Siyan, Ph. D.
New Riders Publishing, Indianapolis, IN
1995.
pag. 58, 960

http://www.microsoft.com/cluster

http://www.compaq.com/tru64Unix/cluster

http://www.compaq.com/openVMS/cluster

http://www.compaq.com/cluster

http://www.compaq.com/storage

http://www.digital.com/WindowsNTClusters

http://www.tandem.com

http://www.stratus.com

http://www.ibm.com/servers/clusters

http://www.compaq.com/storage/scsi

http://www.hp.com/marathon

http://www.unisys.com/