UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.

FACULTAD DE INGENIERÍA

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES CAMPUS ARAGÓN

"APLICACIÓN DE LA TECNOLOGÍA CLUSTER EN PLATAFORMA UNIX COMO SOLUCIÓN AL PROBLEMA DE ALTA DISPONIBILIDAD EN SISTEMAS DE CÓMPUTO DE MISIÓN CRÍTICA"

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO EN COMPUTACIÓN

PRESENTAN

Ezequiel Cárdenas Climaco
Ignacio García Cortes
Miguel Angel García Palacios
Víctor Manuel Gomar Matú
Sandro Mendoza Castrejón

DIRECTOR DE TESIS
FIS. Raymundo Hugo Rangel Gutiérrez

MÉXICO D.F. NOVIEMBRE, 2002

TESIS CON
FALLA DE ORIGEN

Universidad Nacional Autónoma de México
UNAM
Biblioteca Central
Dirección General de Bibliotecas de la UNAM

UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

PAGINACIÓN DISCONTINUA

Autorizo a la Dirección General de Bibl... UNAM a difundir en formato electrónico ... contenido de mi trabajo recepc...
NOMBRE: EZEQUIEL CARDENAS CISNEROS
FECHA: 13-nov-2002
FIRMA:

Agradecimientos

A la **Universidad Nacional Autónoma De México** que nos brindó la oportunidad de realizar nuestra formación profesional, ofreciéndonos apoyo incondicional en la búsqueda de este objetivo.

A la **Facultad de Ingeniería**, por la formación académica que nos otorgó y el espacio brindado para la realización de nuestros estudios profesionales y, por todo lo que representa la Facultad de Ingeniería.

Al director de tesis **Fís. Raymundo Hugo Rangel Gutiérrez** por su dirección y apoyo.

Al **Instituto de Geofísica** y especialmente a **Miguel**, por brindarnos un espacio de trabajo en el cuál pudiéramos concluir esta tesis.

Dedico este trabajo de Tesis:

A Dios

Que si en verdad no existirá, no se quien me hubiese proporcionado la fuerza para seguir adelante y lograr este objetivo.

A mi esposa Rocío.

Que gracias a su insistencia y a las varias noches en espera, dieron como resultado esta tesis. Así que, por todo el amor y ayuda que me brindas además del esfuerzo compartido te doy las GRACIAS.

A mis Padres

Gracias a las carencias, ahora parcialmente solventadas, que me enseñaron a darle una valor significativo a todo. Los tropiezos que tuvimos en nuestros caminos ya fueron superados ya que ahora nos levantamos para seguir adelante y continuar con todos nuestros objetivos. Por todo el apoyo gracias mamá Agustina. Por ser un ejemplo de una persona que sabe sobreponerse gracias papá Moisés.

A mis hermanos Rocío, Angélica y Reynaldo

Que muy a su manera tuve un apoyo, además a la paciencia que me tuvieron cada uno de ellos.

A la Familia Granada Ramírez

Por el apoyo incondicional, y por estar en esos momentos más difíciles de mi vida, con nada podré pagar todo eso.

A mis amigos

Por su apoyo incondicional, sabiendo que siempre contaré con ellos cuando los necesite.

Y por supuesto, **a ti** que aportaste algo valioso a mi vida y que nunca te separaste de mi lado proporcionándome ánimos para lograr este objetivo.

Sinceramente

Exequiel Cárdenas Climaco

Agradecimientos

A Dios:
Por darme la oportunidad de terminar mis estudios profesionales.

A mis padres y hermanos:
Por el apoyo que me han brindado en todos estos años de formación profesional.

A la familia Neyra Martínez:
Que me brindaron su apoyo cuando realmente lo necesitaba.

A mis amigos y compañeros:
Que además de que fueron un buen equipo de trabajo, soportaron todos mis desplantes y enojos.

Dedicatorias

A Dios, mis padres, mis hermanos, a todas las personas que estuvieron a mi lado brindándome su apoyo y comprensión.

Miguel Ángel García Palacios

Agradecimientos

A Dios

Por la fuerza para no desmayar y seguir hasta el final. Por la sabiduría y la inteligencia otorgadas cada segundo de mi vida. Por las bendiciones de cada mañana.

A mis padres

Porque sin el esfuerzo y el apoyo incondicional brindados hubiera sido imposible concebir un proyecto de vida.

A Saraí Chávez

Por el cariño mostrado, por el apoyo y tiempo dedicados en la realización de este trabajo.

A mis compañeros y amigos

Por las palabras de apoyo enunciadas con profunda amistad. Por el apoyo, tiempo y esfuerzo, proporcionados a lo largo de la carrera y en la realización de este trabajo.

Dedico este trabajo de tesis a:

A ti **DIOS** por todo lo que me has dado, a ustedes padres **Víctor y Maria Elena** por darme todo en la vida, por sus enseñanzas, por su apoyo, cariño, comprensión y esfuerzo, a ti hermano **Francisco** por tu comprensión y apoyo.

Con Sinceridad,

Victor Gomar Matú

Í N D I C E

1 INTRODUCCIÓN

En el pasado solo unas cuantas aplicaciones eran consideradas suficientemente críticas para requerir operación continua. Éstas, se encontraban en empresas grandes, dentro de ciertos segmentos del mercado tales como el de servicios financieros y telecomunicaciones. Debido a que el costo del cómputo ha disminuido en los últimos tiempos y las nuevas tecnologías como el Internet han aparecido, las empresas han introducido nuevas y diversas aplicaciones que no hubiese sido posible concebir hace unos años. Las aplicaciones recientes han permitido incrementar la productividad proveyendo más y mejor información a los profesionales encargados de la toma de decisiones en las empresas e incrementando la velocidad de las transacciones de negocios. Pero éstas también han provocado que las compañías sean mucho más dependientes de sus sistemas de cómputo.

Debido a que las aplicaciones críticas de los negocios que han sido portadas a un sistema de cómputo han crecido de manera considerable, la tolerancia de la empresa para tener su sistema caído ha disminuido de manera dramática.

Hace solo unos años las empresas podían tolerar horas de sistema caído causado por fallas o mantenimientos planeados de los equipos de cómputo. La gran mayoría de las empresas actualmente solo permiten segundos o minutos de sistema caído. Un estudio reciente, realizado por el Standish Group International, demuestra que el 29% de las empresas puede tolerar solamente entre cero y tres segundos de sistema caído por evento en sus aplicaciones de misión crítica. Otro 37% de ellas puede tolerar hasta tres minutos de sistema caído. Estas cifras demuestran que en total un 66% de las empresas consideran "tolerable" el rango de minutos o segundos de sistema caído. Las cifras acentúan el hecho de que el acceso continuo y a tiempo de la información, aplicaciones y redes, se ha convertido en una actividad imperativa. También ayudan a explicar porque la implementación de productos y servicios innovadores de alta disponibilidad, se ha convertido en una actividad usual dentro de los corporativos a escala mundial.

En los últimos años la tecnología de la información ha entrado a una nueva era. La popularización del Internet, intranets corporativos y redes comerciales, así como la proliferación de cómputo personalizado, ha resultado en cientos de aplicaciones que millones de usuarios accesan de manera remota al mismo tiempo. Aplicaciones que requieren operación de 24 x 7 se han convertido ahora en un requerimiento estándar de organizaciones del sector financiero, de manufactura, público, telecomunicaciones y otros.

¿Que sucedería si una aplicación crucial para los negocios de una empresa no contara con el nivel requerido de disponibilidad?. Cada falla ocasionaría pérdida en las ganancias, pérdida de oportunidades y pérdida de productividad. La disponibilidad es uno de los aspectos más cruciales en un sistema empresarial, el costo de una hora de tiempo fuera de servicio puede variar de entre miles de dólares en una empresa de paquetería y envío, a millones de dólares para una empresa de servicios financieros. Un análisis reciente de la Universidad de Minnesota revela que una de cada cinco de las 500 empresas más grandes del mundo quebraría si sus sistemas y/o redes no están disponibles por cuarenta y ocho horas o más. De hecho, se ha calculado que el costo de reconstruir la infraestructura tecnológica de Wall Street, después de los atentados del 11 de septiembre, es de tres mil doscientos millones de dólares

Ahora más que nunca, muchos más negocios no pueden operar de manera efectiva si las aplicaciones principales están caídas. Afortunadamente, cada día existen más opciones de solución para sistemas de alta disponibilidad y el costo de los mismos está disminuyendo.

A diferencia de la disponibilidad tradicionalmente manejada en servidores en la cual el hardware es la clave, actualmente y basándonos en estudios realizados por diversas instituciones, sabemos que las principales causas de detención no planeada de un sistema pueden ser:

- Software Básico
- Hardware
- Error Humano
- Problemas de transmisión
- Desastre Natural

Como hemos mencionado, la alta disponibilidad es uno de los atributos más importantes de un sistema de cómputo en la actualidad. La disponibilidad, una medida de que tan accesible es un sistema y una aplicación para entregar el servicio esperado al usuario, es definida también como el porcentaje de tiempo que un sistema y una aplicación están corriendo contra el tiempo que los mismos están abajo por mantenimiento o reparación; puede ser alcanzada de diversas formas, desde soluciones que utilizan hardware y redundancia, hasta las soluciones de software que usan componentes específicos de hardware para proveer alta disponibilidad.

A lo largo de esta exposición hablaremos acerca de algunas soluciones de alta disponibilidad que han sido desarrolladas. Profundizaremos aún más en la tecnología cluster, la cual actualmente se aplica en tres principales áreas: Alto Desempeño (High Performance – HP), Alta Disponibilidad (High Availability – HA) y Procesamiento Distribuido.

El cluster de alto desempeño, permite el procesamiento simétrico y se utiliza porque brinda escalabilidad e incrementa el poder de procesamiento al habilitar una aplicación que es atendida por todos los nodos del cluster a la vez. Por otro lado, en la perspectiva de una aplicación de misión crítica; un cluster forma un grupo de servidores independientes que se manejan como un sistema simple, comparten un mismo espacio de nombres y es diseñado específicamente para tolerar fallas eliminando el punto simple de falla generado por el servidor en si mismo e impidiendo que el usuario siquiera la perciba.

El procesamiento distribuido permite que los componentes de una aplicación sean distribuidos en varios nodos con el objetivo de obtener mayor eficiencia en el procesamiento de los datos.

2 SOLUCIONES DE ALTA DISPONIBILIDAD

2.1

INTRODUCCIÓN

Dos tendencias en la computación hacen que las soluciones de hardware sean factores significantes para alcanzar la disponibilidad de un sistema.

- El incremento de la complejidad de los sistemas y de las aplicaciones de software
- El incremento de la dependencia que un negocio tiene de su información y aplicaciones

Estas tendencias fueron evidenciadas en un estudio[1] sobre las fallas de los sistemas realizado entre 1985 y 1993. Durante este periodo ocurrió un cambio, del modelo predominante de servidores simples (Stand-alone) sirviendo terminales mudas a un modelo distribuido más poderoso y complejo de cliente – servidor. Mientras este modelo se fue haciendo predominante, nuevas y variadas soluciones de hardware surgieron.

[1] Brendan Murphy and Ted Gent. "Measuring system and software reliability using an automated data collection process". Quality and Reliability Engineering International, Page 13, 1995. CCC 0748-8017/95/050341

El estudio concluye que la porción de interrupciones de sistema, red y aplicaciones causadas por la confiabilidad y disponibilidad del hardware va del 15 a! 35% por ciento de todas las caídas del sistema durante el periodo evaluado. El mismo sugiere que una buena planeación de la solución de hardware a utilizar en la implementación de un sistema o aplicación es uno de los componentes importantes para crear un ambiente de cómputo cliente – servidor confiable y altamente disponible. Una solución de hardware bien planeada y bien documentada ayudará a disminuir las fallas de un sistema.

2.2

IMPLICACIONES DE LA ALTA DISPONIBILIDAD

Antes de entrar de lleno a los temas sobre métodos para incrementar la disponibilidad y de desarrollar el caso de estudio, es crucial entender algunos términos importantes.

Falla

Definimos falla como la desviación del comportamiento esperado en un sistema de cómputo o en un sistema de red.

Las fallas pueden incluir comportamientos que simplemente van más allá de los parámetros previamente definidos. Si el comportamiento especificado de un sistema incluye tiempos comprometidos tales como el requerimiento para completar cierto procesamiento en un lapso de tiempo específico, la degradación de desempeño (Performance) más allá del límite establecido, es considerado una falla.

Por ejemplo, un sistema que debe procesar una transacción en dos segundos, puede incurrir en un estado de falla si el procesamiento de la misma se degrada y no se cumple en el tiempo determinado.

El software, el hardware, los errores de procedimiento y de operador, así como los factores de ambiente; pueden ser causantes de falla en un sistema. Una encuesta reciente encontró que mientras las fallas de los componentes de hardware son responsables del treinta por ciento de las interrupciones en un sistema, las fallas del sistema operativo y de las aplicaciones ocasionan casi el treinta y cinco por ciento de downtime no planeado. Los componentes más comunes de hardware que pueden fallar son: ventiladores, unidades de disco, fuentes de alimentación eléctrica, etc.

La falla de un componente simple o del equipo completo, puede influenciar directamente la fiabilidad de todo el sistema.

Fiabilidad (Reliability)

La fiabilidad es una medida del tiempo que transcurre entre las fallas que ocurren en un sistema. Los componentes de hardware y software tienen diferentes características de falla. Aunque existen fórmulas que se basan en datos históricos para predecir la fiabilidad del hardware, es difícil encontrar fórmulas para predecir la fiabilidad del software.

Los componentes de hardware normalmente presentan lo que es conocido como una distribución exponencial de falla. Bajo circunstancias normales y después de una fase inicial; el componente de hardware que más tiempo de operación tenga será el que más frecuentemente falle.

Si el tiempo promedio entre falla (Mean Time To Fail -MTTF-) de un dispositivo es conocido, entonces es posible predecir cuando el componente de hardware fallará. Datos históricos sobre componentes mecánicos y eléctricos coinciden con la curva de "Bathtub" mostrada en la Figura 2.2-1.

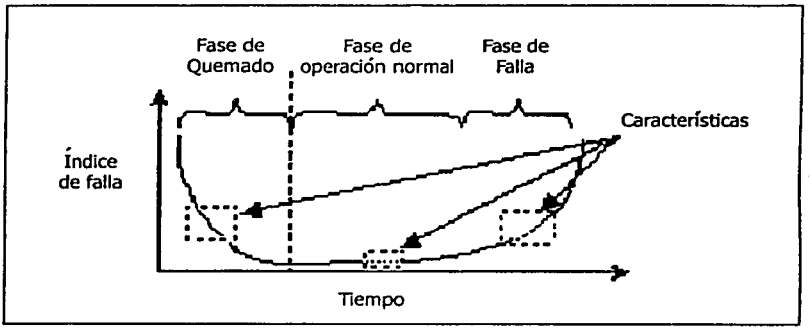


Figura 2.2-1 Curva de Bathtub[2].

En este modelo, el ciclo de vida de un componente está formado por tres fases claramente distinguibles:

- Fase de quemado
- Fase de operación normal
- Fase de falla

Cada fase se caracteriza por algún comportamiento particular. La incidencia de fallas es muy alta durante la fase de quemado, pero cae rápidamente durante el periodo de operación normal, donde los dispositivos ocasionalmente fallan.

[2] La curva es el resultado de la combinación de tres diferentes ecuaciones. La fase inicial de quemado identificada como curva de aprendizaje, combinada con incrementos lineales y exponenciales en la segunda y tercera etapa de la vida de un componente. Para mayor referencia ver http://www.sys-ev.com/reliability01.htm

Conforme el tiempo pasa, la incidencia de falla de los dispositivos crece dramática y predeciblemente. La curva es el resultado de la combinación de tres diferentes ecuaciones. La fase inicial de quemado identificada como curva de aprendizaje, combinada con incrementos lineales y exponenciales en la segunda y tercera etapa de la vida de un componente.

La Información sobre el MTTF de los componentes de hardware puede ser usada para calcular el índice de falla de algunos dispositivos específicos y para entonces reemplazarlos antes de que ellos entren en el período o tiempo de falla. Esta estrategia es frecuentemente utilizada cuando el costo de un componente en falla puede ser catastrófico.

La Tabla 2.2-1 muestra los valores de MTTF para los componentes de hardware más comunes.

Componente	MTTF
Conectores y Cables	1,000 años
Tarjetas Lógicas	3 – 20 años
Discos	1 – 50 años
LAN	3 semanas
Alimentación Eléctrica (U.S.A.)	5.2 meses

Tabla 2.2-1 MTTF[3] de los componentes más comunes de hardware.

[3] Jim Gray y Andreas Reuter. Transaction Processing: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco, 1993.

Es importante hacer notar que la fiabilidad de diversos componentes de hardware del mismo tipo, puede variar considerablemente dependiendo de una gran variedad de influencias. Por ejemplo, áreas geográficas con tormentas eléctricas frecuentes u otras condiciones ambientales extremas, experimentan diferentes índices de falla. Estas condiciones estresan los componentes más allá de su diseño normal y de las especificaciones de un ambiente operativo.

Estadísticas como las mostradas en la Tabla 2.2-1 no están disponibles para defectos de software, ya que los errores de datos generados deberían ser predecibles para poder desarrollar fórmulas que anticipen de forma eficaz la falla de una aplicación.

Disponibilidad (Availability)

Disponibilidad es una medida de la cantidad de tiempo que un sistema o componente del mismo realiza su función específica. La disponibilidad se relaciona con la fiabilidad pero es diferente a ella. La fiabilidad mide que tan frecuente un sistema falla mientras que la disponibilidad determina el porcentaje de tiempo que el sistema esta en su estado operacional.

De esta manera, la meta final es entonces, mantener todos los recursos de cómputo, aplicaciones y servicios disponibles para el usuario. Esto puede lograrse de diversas formas, que pueden ir desde aquellas que utilizan hardware redundante y configurado a la medida para asegurar disponibilidad, hasta las que utilizan software en conjunto con componentes de hardware.

Para calcular disponibilidad se necesita conocer dos tiempos, el tiempo promedio de falla (Mean Time To Failure MTTF[4].) y el tiempo promedio de

[4] El MTTF es también conocido o referenciado como MTBF (Mean Time Between Failure.).

recuperación (Mean Time To Recovery MTTR[5].). El MTTR es una medida de que tanto toma en promedio recuperar el sistema a su estado operacional después de una falla. Si se conocen ambos tiempos es decir el MTTF y el MTTR se puede calcular la disponibilidad usando la siguiente fórmula:

$$Disponibilidad = \frac{MTTF}{MTTF + MTTR}$$

MTTF = Mean Time To Failure (Tiempo Promedio De Falla)
MTTR = Mean Time To Recover (Tiempo Promedio De Recuperación)

Por ejemplo, si a un centro de cómputo le toma un promedio de seis meses presentar una falla (MTTF = 6 meses) y le toma 20 minutos en promedio, regresar a su estado operacional (MTTR = 20 minutos), entonces su disponibilidad es:

$$Disponibilidad = \frac{6\ meses}{6\ meses + 20\ \text{min}} = 99.992\%$$

Nótese entonces que existen dos caminos para incrementar la disponibilidad de un sistema –incrementando el MTTF o reduciendo el MTTR. Un centro de cómputo eficiente generalmente intenta hacer ambas cosas. La selección cuidadosa de hardware y software puede ayudar a incrementar el MTTF de un

[5] El MTTR es también conocido o referenciado como MTTR (Mean Time To Repair)

sistema, mientras que mantener partes de refacción a la mano, servidores de respaldo y el uso de diversas tecnologías como puede ser la de cluster puede ayudar a reducir el MTTR del mismo.

La Figura 2.2-2[6] muestra que el tiempo de reparación o downtime para un servidor de aplicaciones, es el intervalo de tiempo requerido para restaurar la operación normal del negocio. Al final del periodo de reparación, las aplicaciones funcionales estarán disponibles para los usuarios. El tiempo de reparación incluye tanto la recuperación de la aplicación como la recuperación del servidor.

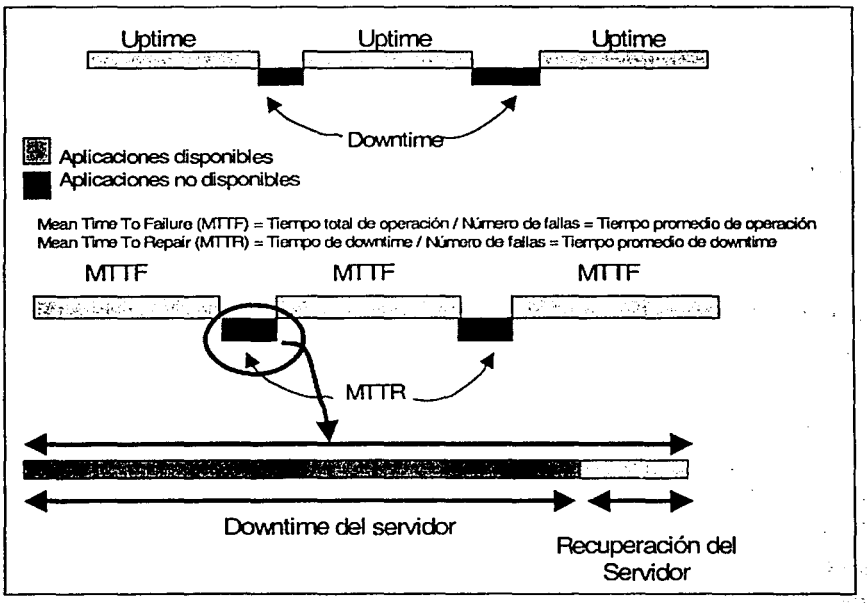


Figura 2.2-2 Anatomía de la falla de un servidor simple.

[6] Calculating Availability for Applications White Paper, UNISYS

La recuperación del servidor comprende el tiempo requerido para:

- Detectar y diagnosticar la falla
- Reparar el servidor
- Levantar el sistema operativo

La recuperación de la aplicación comprende el tiempo requerido para:

- Reiniciar y recuperar la base de datos
- Reiniciar el software de aplicación

Niveles de Disponibilidad[7]

Todas las compañías desearían sistemas de alta disponibilidad, términos como "garantizar el 99.9% de disponibilidad" son frecuentemente utilizados, pero ¿Qué significa realmente esto? Las garantías de disponibilidad están típicamente basadas en una operación continua de veinticuatro horas al día, siete días a la semana y trescientos sesenta y cinco días al año. Todos los gerentes de informática sueñan con los cinco nueves (99.999%) y resulta una meta donde llegar, solo cinco minutos y quince segundos sin servicio por año. Pero llegar ahí es costoso y se cumple la regla 80/20 (El primer ochenta por ciento del esfuerzo representará el veinte por ciento del costo total.).

Disponibilidad es típicamente definida como un porcentaje de uptime. Basándose en una operación de 7x24x365; la Tabla 2.2-2 relaciona los porcentajes de disponibilidad contra el monto de downtime por año que representan.

[7]Definidos por el Harvard Research Group en 1999.

% de Uptime	% de Downtime	Anual	Normalizado (segundos)
98%	2%	7.30 días	630,720
99%	1%	3.65 días	315,360
99.8%	0.2%	17 horas, 30 minutos	63,000
99.9%	0.1%	8 horas, 45 minutos	31,500
99.99%	0.01%	52 minutos, 30 segundos	3,150
99.999%	0.001%	5 minutos, 15 segundos	315
99.9999%	0.0001%	31.5 segundos	31.5

Tabla 2.2-2 Porcentajes de disponibilidad.

El Harvard Research Group ha definido la disponibilidad en términos del impacto que un sistema no disponible tiene en la actividad de los negocios y del consumo (usuario final) de un servicio, en lugar de evaluar las tecnologías usadas para alcanzarla. Los cinco ambientes de disponibilidad (Availability Environments AE) descritos más adelante, definen la disponibilidad en términos del impacto en los negocios y al usuario final o consumidor.

Nivel AE-0 Convencional: Las funciones de negocios pueden verse interrumpidas y la integridad de los datos no es esencial.
Disponibilidad: < 90% **Mecanismos:** Respaldo tradicional

Nivel AE-1 Medio: Las funciones de negocio pueden verse interrumpidas, pero se debe mantener la integridad de los datos.
Disponibilidad: <95% **Mecanismos:** Journaly[8]

[8] Ver glosario

Nivel AE-2 Alta Disponibilidad: Las funciones de negocios aceptan pequeñas interrupciones y al recuperar la operación se requiere re-procesar algunas transacciones.

Disponibilidad: <99% **Mecanismos:**Journaly, Clustering y Recuperación automática

Nivel AE-3 Resistencia a Fallas (Fault Resilient): Requiere de operación sin interrupción en horario laboral, la operación es recuperada de manera automática.

Disponibilidad: <99.9% **Mecanismos**: Clustering y Espejeo

Nivel AE-4 Tolerancia a fallas (Fault Tolerance): Capacidad de procesamiento continuo, ya que cualquier falla deberá ser transparente para el usuario.

Disponibilidad: <99.99% **Mecanismos:** Duplicidad y Redundancia

Nivel AE-5 Tolerancia a desastres: Disponibilidad en todo momento, incluyendo la capacidad para soportar desastres naturales y humanos.

Disponibilidad: <99.999% **Mecanismos:** Los anteriores más redundancia del centro de cómputo y de los procedimientos de recuperación

Costo del downtime

No muchos gerentes de tecnología de información (IT) toman en cuenta el costo del downtime. Los Presidentes Corporativos de Informática (CIO Corporate Information Officers) lo dimensionan de una forma mas adecuada. El downtime

no sólo implica el costo de la falla actual y su consecuente reparación, implica también muchos factores que no pueden ser rápidamente cuantificados.

Tales factores incluyen pérdida de ingresos, pérdida de productividad, pérdida de reputación, pérdida de clientes y hasta del mismo negocio. Vamos hablar con más detalle cada uno de estos factores.

La pérdida de ingresos impacta el mayor porcentaje del costo total del downtime. De acuerdo con una investigación realizada en 1998 por el Strategic Research Corporation (SRC) El porcentaje del costo de una hora de downtime de una compañía de ventas por catálogo es de noventa mil dólares, en cambio una agencia financiera pudiera perder hasta seis millones y medio de dólares por una hora de downtime. Cuando los empleados no pueden hacer su trabajo la compañía está perdiendo dinero. Actualmente, casi el total de las funciones de los empleados de una empresa tienen que ver en gran manera con aplicaciones de red y manejo de sistemas de cómputo. Hace cinco años, un correo electrónico podría no haber sido considerado una aplicación de misión crítica. Hoy en día esto es diferente. Solo pregunte a cualquiera que no haya tenido acceso a su correo durante un día o dos.

Si un cliente no puede entablar contacto con su proveedor de servicios, existe la posibilidad de que el pueda elegir otra empresa. Cuantas veces hemos estado en un banco o en otro negocio donde tenemos que esperar porque los procesos automatizados están caídos, hacer negocios puede ser frustrante bajo estas condiciones.

Entonces, ¿cómo podemos determinar el costo de downtime actualmente? Los métodos mostrados en la Tabla 2.2-3 pueden ser utilizados individualmente o en combinación para precisar de mejor manera el costo del downtime.

Métodos para calcular costo de downtime
1. Porcentaje de ingresos por hora = Costo de una hora de downtime.
2. Porcentaje del número de transacciones por hora x porcentaje del valor de la transacción = al costo de una hora de downtime.
3. Número de usuarios x el costo de usuario por hora = costo de una hora de downtime.

Tabla 2.2-3 Métodos para calcular costo de downtime.

La primer fórmula es simple y clara ¿cuál es el porcentaje de ganancia por hora? Ese será el costo promedio de una hora de downtime. La segunda fórmula toma en cuenta el número de transacciones en una hora promedio y el valor promedio de esas transacciones. Esto proporcionará un costo más preciso de downtime. La tercer fórmula toma en cuenta el costo de los empleados y clientes. El número de usuarios puede ser fácilmente comparado al número de empleados usando las aplicaciones, multiplicando ese número por el salario promedio podremos tener una buena idea del costo del tiempo ocioso de los empleados en una hora de downtime. Combinando el resultado de las tres fórmulas obtendremos una buena idea del costo total de una hora de downtime. Multiplicando este resultado por la duración del downtime, tendremos el costo del downtime de un periodo.

El costo del downtime es típicamente calculado, promediando la pérdida de la productividad, negocio y reputación por el tiempo específico de duración. Sin embargo, este tipo de medida descuenta una variable mayor que influencia el costo del downtime – horario de ocurrencia -. Por ejemplo, el costo del downtime durante horas no hábiles, es significativamente menor que durante horas hábiles, debido a que la actividad de la compañía es mucho menor y el tráfico de transacciones es dramáticamente menor también. Debido a ésto, la pérdida de productividad, negocio y reputación se reduce dramáticamente.

La Tabla 2.2-4 muestra el costo promedio del no servicio en que los diversos sectores del mercado incurren.

Sector	Ingresos / Hora
Energía	US $2,817.846
Telecomunicaciones	US $2,066.245
Financiero	US $1,495.134
Comercio	US $1,107.274
Químico	US $ 704.101
Salud	US $ 636.030
Entretenimiento	US $ 340.432

Tabla 2.2-4 Costo por hora del downtime en diversos sectores del mercado.

Causas del downtime

Diversos estudios basados en retroalimentación extensiva de parte de los usuarios, estiman que las fallas o interrupciones de un sistema se deben o se presentan por diversas causas tales como: virus informáticos, fallos de electricidad, errores de hardware y software, caídas de red, piratas informáticos, errores humanos, incendios, inundaciones, etc. Y aunque no se pueda asegurar que cada una de estas interrupciones nunca ocurrirá, las empresas sí pueden prepararse para evitar las consecuencias que éstas puedan tener sobre su negocio, al invertir en mejorar procedimientos de manejo de problemas,

controles de cambio, herramientas automatizadas, eliminación de puntos simples de fallas (Single Point Of Failure SPOF.).

La Figura 2.2-3 muestra el resultado de un estudio realizado por IBM a cerca de las causas más frecuentes de downtime no planeado, así como el porcentaje de incidencia de las mismas

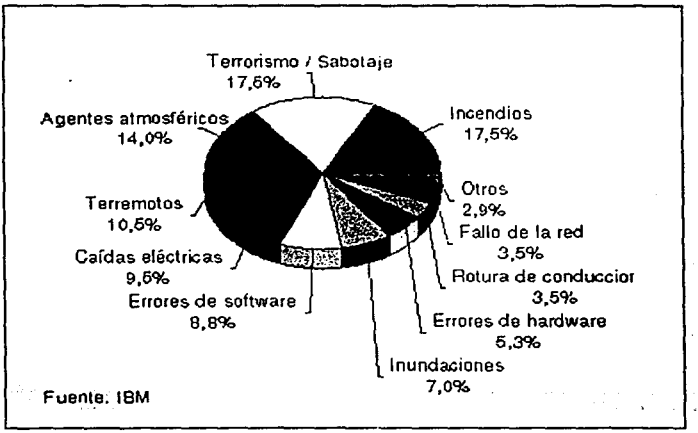


Figura 2.2-3 Causas del downtime no planeado.

2.3

REDUNDANCIA

Nos referimos a la redundancia de un sistema cuando uno o más de sus componentes se repiten para que de esta manera se mantenga el sistema activo. Esto significa que si los recursos del sistema sufren algún tipo de caída su duplicado retoma la actividad, de esta manera; el sistema se encuentra disponible en todo momento que se requiera. Existen técnicas de redundancia en el nivel de hardware y en el nivel de software.

Técnicas de redundancia en el nivel de hardware

Este tipo de redundancia se basa en la replicación de módulos para lograr una mayor garantía de funcionamiento. Existen tres tipos de redundancia:

- La redundancia pasiva, que se basa en el enmascaramiento de las fallas (F*ault-masking*)[9]

[9] El *fault-masking*, es una técnica que se utiliza para ocultar los efectos que pueda causar una falla a través de considerar que la información redundante tenga mayor peso que la información incorrecta, es decir, por votación mayoritaria.

- La redundancia activa, en donde existe un proceso de detección, aislamiento, reconfiguración y recuperación de la falla
- La redundancia híbrida que es una mezcla de las dos anteriores, pasiva y activa; aunando las ventajas de cada una de ellas

Redundancia pasiva

Una de las características de este tipo de redundancia es que se basa en la replicación de módulos con un proceso de votación, implicando con esto alta redundancia y alto costo, no se lleva a cabo la detección de errores ya que los mismos componentes impiden que se produzcan los errores (Visto desde fuera del módulo replicado), el inconveniente es que desde fuera (Del módulo) no se sabe si ha habido un error. La recuperación y reparación del sistema después de la falla se realiza por enmascaramiento (*Fault-masking.*). Este tipo de redundancia se utiliza en sistemas de alta fiabilidad.

Sistemas redundantes con N módulos y votación (NMR)

Los sistemas redundantes con *N* módulos son los más extendidos cuando se requiere una alta fiabilidad, se basan en la existencia de *N* módulos *MN* que realizan la misma función más un votador *V* que realiza el voto por mayoría. La salida es correcta mientras la mayoría de los módulos tengan un funcionamiento correcto. Para permitir *F* fallos, necesitaremos *N* módulos con $N = 2F + 1$. El sistema más utilizado es el Triple Modular Redundante (ver Figura 2.3-1.).

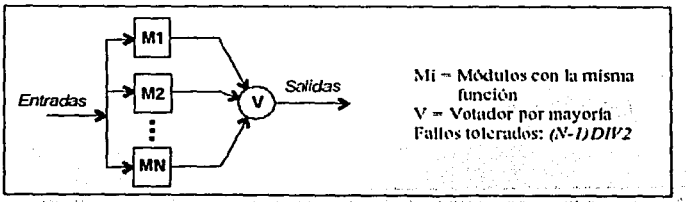


Figura 2.3-1 Sistema redundante con N módulos.

El diseño de votadores con entradas binarias se muestra en la Figura 2.3-2.

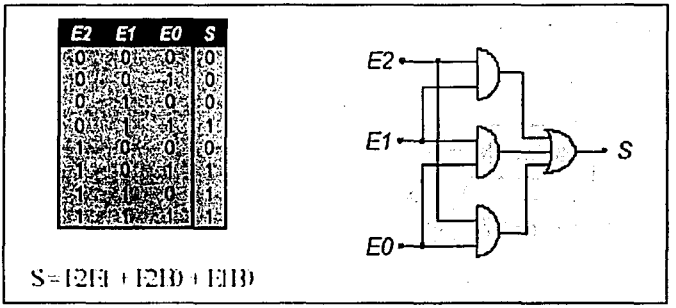


Figura 2.3-2 Diseño de votadores.

El problema que tiene el votador arriba mencionado es la sincronización[10], así que la manera de solucionar este problema se muestra en la Figura 2.3-3.

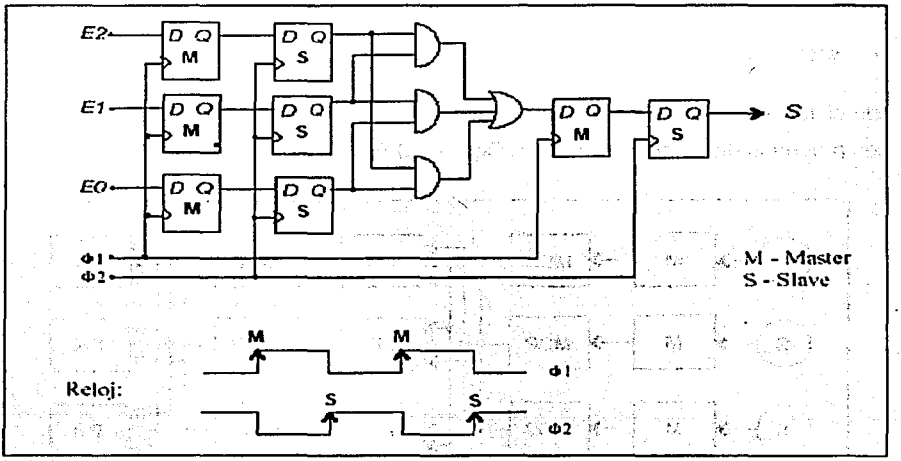


Figura 2.3-3 Votador sincronizado.

[10]Coordinar los elementos involucrados en el proceso a través de un reloj generador de pulsos eléctricos.

En el diseño de un sistema con tolerancia a fallas en los votadores, se establecen votadores triplicados para dos etapas; de esta forma el sistema tolera la falla de un módulo o un votador por etapa, véase la Figura 2.3-4.

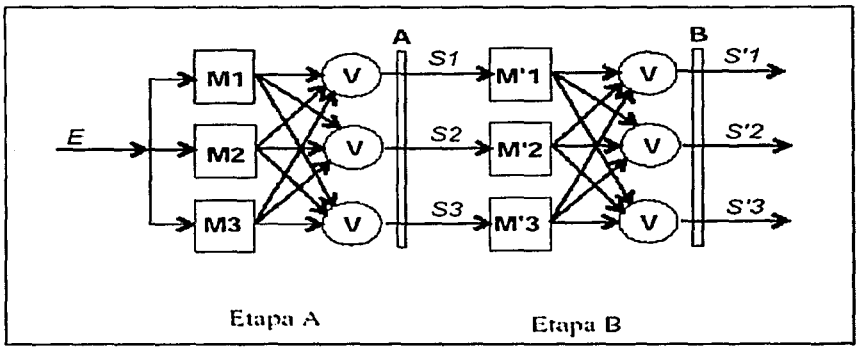


Figura 2.3-4 Sistema con tolerancia a fallas de los votadores.

Votación usando procesadores

Un sistema de control para varias etapas es un ejemplo que muestra claramente los procesos de votación, véase la Figura 2.3-5.

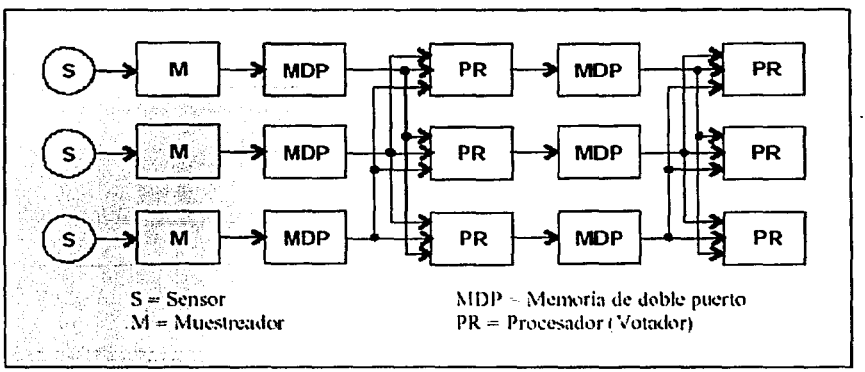


Figura 2.3-5 Votación en un sistema de control de varias etapas.

Redundancia activa

En comparación con la redundancia pasiva, en este tipo de redundancia se tiene como misión primaria la detección de los errores. Existen varios tipos de detección de los errores:

- **Detección en el nivel de señal:** Se utilizan técnicas en línea, su costo es alto, es de baja latencia[11], utiliza códigos detectores de errores y existe una duplicación en el nivel de componentes

- **Detección en el nivel de función:** Se utilizan sistemas duplicados en línea, pruebas de aceptación (fuera de línea) y circuitos comprobables en línea; es una técnica de bajo costo y de alta latencia

- **Detección periódica:** Se realiza en tres etapas:
  - **Detección inicial:** Se realiza fuera de línea cuando se pone en marcha un sistema, es una prueba exhaustiva y general
  - **Detección concurrente:** Es una prueba rápida y general realizada en línea
  - **Detección fuera de línea:** Es el típico diagnóstico realizado fuera de línea como mantenimiento preventivo o cuando ocurre una falla, es una prueba exhaustiva, general o particular

[11] La latencia describe el atraso de una transmisión, desde el momento en que entra en la red hasta el momento en que deja la misma. Una baja latencia significa pequeños atrasos y una elevada latencia significa grandes atrasos.

Una vez detectado el error; se aplica el proceso descrito a continuación con el objetivo de regresar el sistema al estado normal:

- Aislamiento de la falla
- Diagnóstico
- Re-configuración
- Recuperación
- Reparación
- Reintegración

El siguiente paso después de la etapa de detección es el aislamiento de la falla, el cual previene la contaminación de otras áreas del sistema limitando el efecto de la falla. Posteriormente, el diagnóstico es necesario sí y solo sí la detección de errores no informa sobre la localización y propiedades de la falla.

La reconfiguración tiene lugar cuando se detecta un error permanente, el sistema es reconfigurado sustituyendo el componente o bien aislándolo del resto. También en esta etapa se puede degradar el sistema.

En la etapa de la recuperación se utilizan técnicas para tolerar los efectos de la falla. Por ejemplo, el reintento, que tomando en cuenta la latencia del error; consiste en repetir la operación para tolerar fallas transitorias. O la inicialización, en donde, si no ocurren daños; se puede continuar con las tareas justo en donde se detectó el error (Recuperación de una falla transitoria.). A este efecto se le conoce como "*Hot*", el efecto "*Warm*" solo permite continuar con algunos procesos y el "*Cold*" implica una nueva carga de todo el sistema.

El penúltimo proceso llamado reparación, permite reemplazar los componentes que se han detectado en falla y además, el sistema en algunos casos, podrá ser reparado en línea utilizando repuestos. En este punto, se puede hacer uso de la degradación funcional (También fuera de línea.).

Finalmente la última etapa es la reintegración, que consiste en que el módulo reparado se reintegre al sistema. En la Figura 2.3-6 se ilustra el proceso completo.

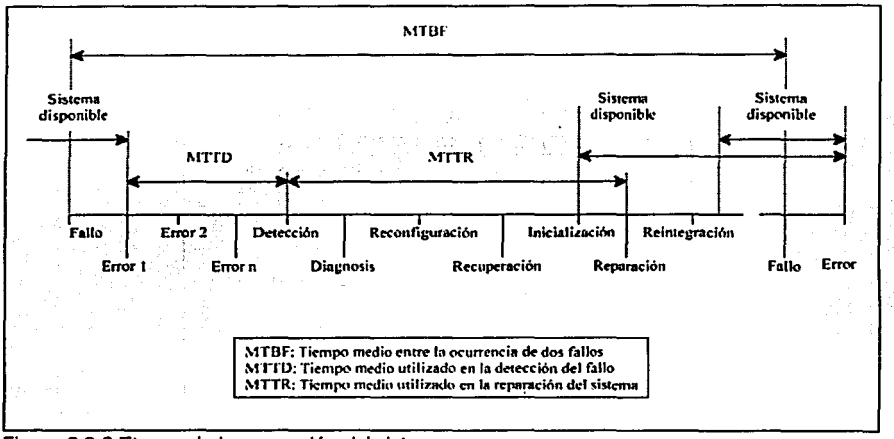


Figura 2.3-6 Etapas de la reparación del sistema.

Duplicación y comparación

En la redundancia activa como se mencionó arriba, es muy importante detectar el error, entonces, es importante duplicar los módulos en donde se realiza la comparación de la salida, ya que ello nos permitirá realizar dicha detección, ver Figura 2.3-7.

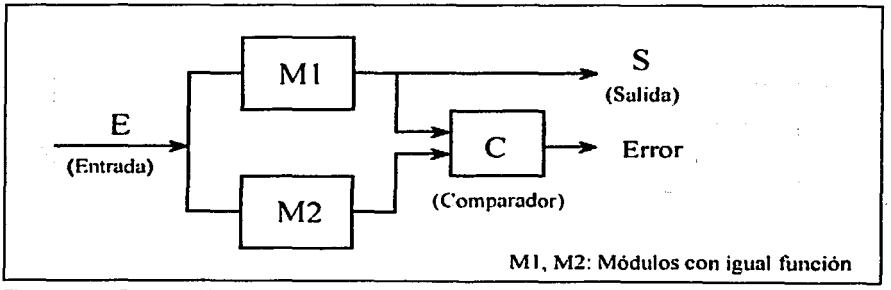


Figura 2.3-7 Duplicación y comparación.

Ejemplos muy claros sobre las fallas de modo común se presentan por ejemplo en memorias RAM y circuitos VLSI (Very Large Scale Integration), la mejora se da a través de la diversificación funcional para el caso de las memorias RAM. (Véase la Figura 2.3-8.).

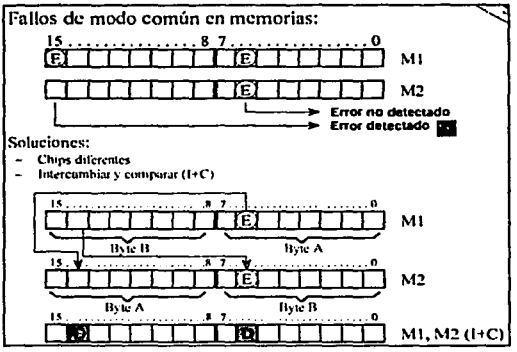


Figura 2.3-8 Detección de errores en una memoria RAM.

Para la falla en modo común en el caso de los chips VLSI, tenemos que la solución puede ser también la técnica de diversificación funcional mediante el uso de lógica complementaria en cada módulo. Como se muestra en la Figura 2.3-9.

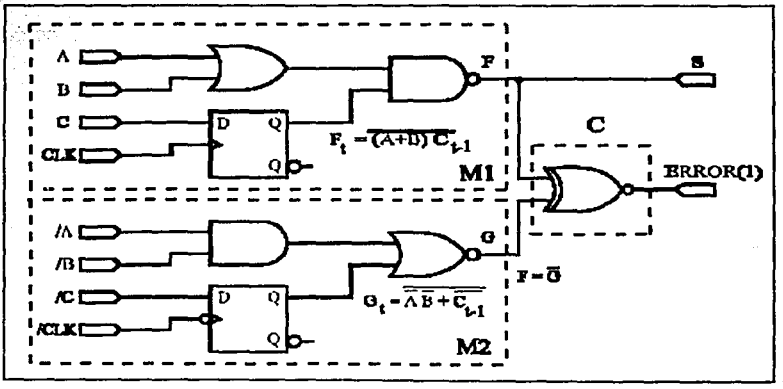


Figura 2.3-9 Detección de errores para chips VLSI.

Redundancia Híbrida

Este tipo de redundancia, se basa en sistemas con redundancia pasiva a los que se les añaden mecanismos de detección de errores, además fusiona las ventajas de la pasiva y la activa:

- Redundancia pasiva
  - Ventajas: Bajos tiempos de latencia y alta garantía de fiabilidad
  - Desventajas: Alto costo, alta probabilidad de falla total en algún módulo previamente en falla

- Redundancia activa
  - Ventajas: Bajo costo y baja complejidad
  - Desventajas: Menor garantía de fiabilidad

La Figura 2.3-10 muestra un sistema NMR con repuestos.

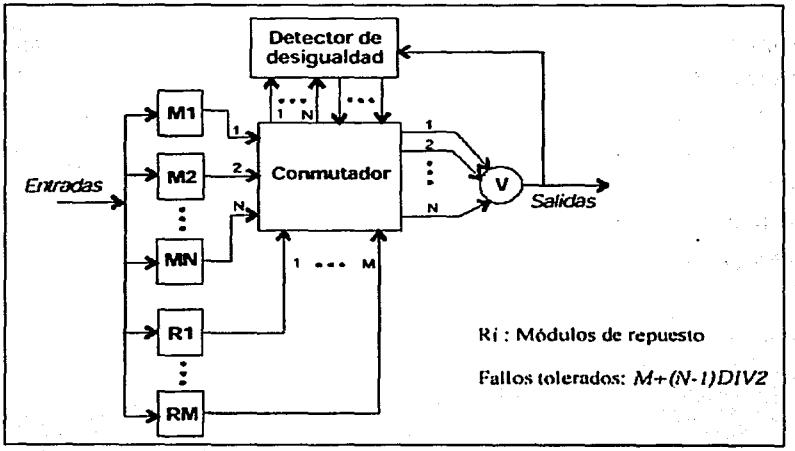


Figura 2.3-10 Sistema NMR con repuestos.

Si consideramos un solo repuesto, el sistema quedaría como se muestra en la Figura 2.3-11.

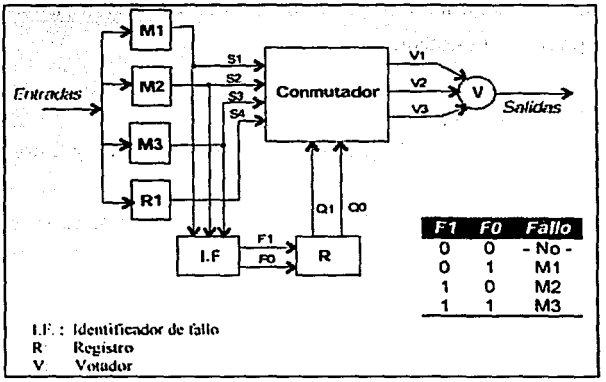


Figura 2.3-11 Sistema NMR con un solo repuesto.

Redundancia por autocomprobación

Este tipo de redundancia híbrida está integrado por un *comparador* que compara las salidas de los módulos 2 a 2. Para un 0, tiene igual salida y para un 1, tenemos diferente salida. El *detector* verifica si el módulo con falla tiene una salida minoritaria con respecto a las demás. Para un 0, es módulo con salida mayoritaria, y para un 1, tenemos módulo con salida minoritaria. Finalmente, el *colector* determina la salida en función de las salidas de los módulos y del detector (Ver Figura 2.3-12.).

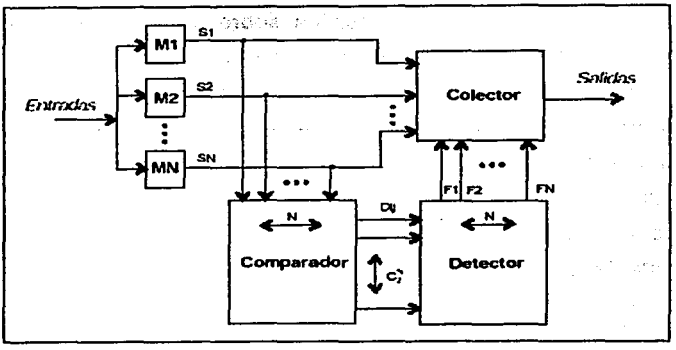


Figura 2.3-12 Redundancia por autocomprobación.

Técnicas de redundancia en software

Cuando se desarrolla un sistema tolerante a fallas, un aspecto de suma importancia que se debe tomar en cuenta, es el conjunto de propiedades que puede ofrecer el software para mejorar la garantía de funcionamiento de un sistema. El uso de técnicas de tolerancia a fallas por software, nos permitirá obtener una alta fiabilidad a partir de un conjunto de componentes de menor fiabilidad. Para decidir la aplicación de una técnica de tolerancia a fallas por software, es fundamental determinar cuales son los componentes más críticos del sistema, luego determinar la técnica más adecuada para el sistema concreto y estimar el costo que va a suponer el desarrollo de software tolerante a fallas.

Las técnicas más difundidas son: Bloques de recuperación (*Recovery Blocks*) y programación de N-versiones (*N-version programming*), en ambos casos se desarrollan distintas versiones del mismo algoritmo, la diferencia reside en la forma en la que se determina el resultado del cómputo. Los aspectos a determinar para la elección de una técnica son: Clases de fallas que se puede tolerar, la redundancia disponible en el sistema (Posiblemente requerida para

aplicar alguna técnica), el desarrollo de diferentes versiones del programa, diseño de los mecanismos de decisión sobre el resultado. Las tendencias actuales para este tipo de redundancia en el software son: La realización de extensiones y combinaciones de los métodos ya existentes, aplicación de software tolerante a fallas en sistemas distribuidos, verificación formal y validación, unificación de notaciones para el software tolerante a fallas y el estudio del software tolerante a fallas en sistemas de tiempo real. A continuación, describiremos brevemente algunas técnicas usadas para implementar la redundancia en el nivel de software.

Bloque de recuperación

Técnica desarrollada para que los programas secuenciales pudieran tolerar fallas cuya localización y efecto no se podía determinar, para su aplicación se supone que un programa se puede dividir en una serie de bloques independientes y que es posible detectar en cada uno de ellos, errores. Cada bloque de recuperación posee: Bloque primario, prueba de aceptación, bloques alternativos (Opcional.).

En la ejecución de un bloque de recuperación se realizan los siguientes procesos:

- Cuando se entra en un bloque se almacena el **estado** de un proceso
- Se ejecuta la rutina primaria
- Al terminar se pasa una prueba **de aceptación**, este tipo de prueba debe ser capaz de determinar la corrección o no de los resultados de cómputo
- Si la prueba se completa afirmativamente, se abandona ese bloque y se evoluciona al siguiente

- Si la prueba detecta un error, se recupera el estado original (antes de ejecutar el bloque), se ejecuta una rutina alternativa, se vuelve a pasar la prueba de aceptación al final (Observar la Figura 2.3-13.)

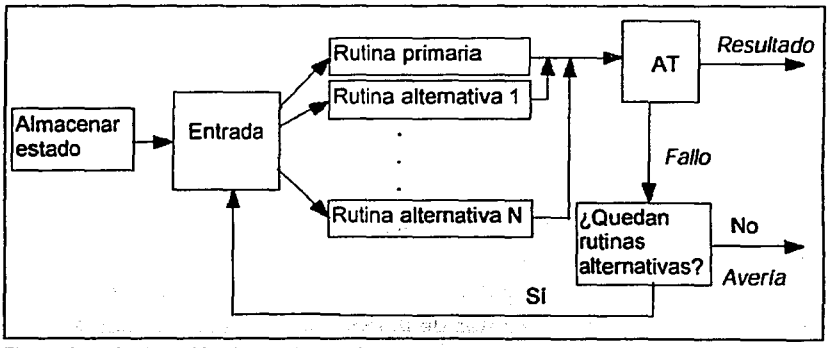


Figura 2.3-13 Ejecución de un bloque de recuperación.

En la Figura 2.3-14, se muestra una descripción de un bloque de recuperación.

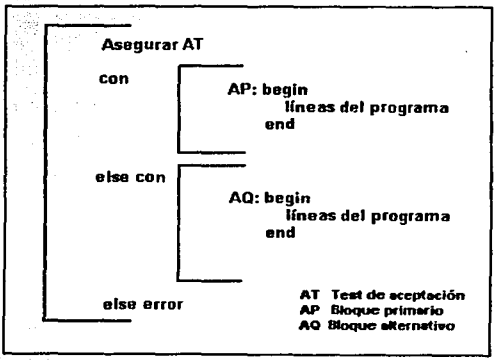


Figura 2.3-14 Descripción de un bloque de recuperación.

Los tipos de fallas que tolera esta técnica son, fallas de diseño de software y errores debidos a fallas transitorias o intermitentes en el hardware. Para el diseño del test de aceptación es importante verificar que valide absolutamente la corrección de los resultados y es aconsejable que el test no sea muy estricto, ya que debido a que las rutinas alternativas pueden ofrecer un *comportamiento degradado* o por su complejidad, pueden presentar fallas de diseño que podrán dar lugar a detección de errores cuando el resultado es correcto.

Para el desarrollo de los bloques primario(s) y alternativo(s), no es necesario un incremento de su complejidad. A partir de las especificaciones se diseñan distintas rutinas que las cumplan, es posible además, diseñar rutinas que ofrezcan un *comportamiento degradado.* Otro punto muy importante es el proveer puntos de recuperación que consiste en almacenar el estado cuando se entra en un nuevo bloque, (Estado, es el conjunto de variables que determinan cómo se encuentra el sistema antes de la ejecución de ese bloque). Por último tenemos la sobrecarga introducida que como mínimo ejecuta el test de aceptación y en caso de error, se ejecutan la o las rutinas alternativas y se almacena el estado del sistema.

Programación de N-versiones

Se define como un conjunto de versiones de un mismo programa a aquellas realizadas a partir de la misma especificación y que utilizan la misma réplica de los módulos de hardware o software. Todas las versiones se ejecutan en paralelo y se realiza una votación de los resultados al final.

La ejecución en paralelo se puede realizar en tres dominios: Tiempo (Repetición), espacio (Hardware) e información (Software). Así, la descripción de un sistema no tolerante a fallas se puede realizar de la siguiente manera: una ejecución (1T), de un único programa (1S), en un único hardware (1H.), o en

distinto hardware o software (Nd). Para la descripción de un sistema tolerante a fallas se toma en cuenta el incremento de la redundancia en algunos de los dominios, tal como se describen los siguientes esquemas:

1T/NH/NdS. 1T/NdH/NS. 1T/NdH/NdS generalmente la programación de N-versiones requiere redundancia de hardware, en este contraste estamos en un esquema similar a los bloques de recuperación. Su esquema clásico consiste en la ejecución de N copias idénticas de programa en N dispositivos de hardware poseé mecanismos de decisión tal como el voto por mayoría (por ejemplo el TMR). Se debe tener cuidado con las fallas de diseño ya sea hardware o software (Es aconsejable realizar distintas versiones para ambos). Es indispensable un "supervisor" que coordine el funcionamiento de las versiones, algunos mecanismos para efectuar la coordinación se presentan a continuación:

- *Vectores de comparación*. Es una estructura de datos que representa el estado del programa.
- *Indicadores de comparación de estado*. Se usan para representar los resultados obtenidos por comparación o votación de los vectores de comparación. Indican si el resultado de una versión difiere de otra o si no se llega a alcanzar un resultado aceptable debido a la falta de conclusión de las versiones.
- *Mecanismos de sincronización*. Usados para sincronizar la ejecución de las distintas versiones. Cada versión usa estos mecanismos para indicar que su vector de comparación ya está disponible. Sirve también, para impedir que se vote antes de que alguna versión termine.

Otros aspectos a tomar en cuenta son que las distintas versiones del programa pueden llegar a resultados similares pero no idénticos y el uso de votación inexacta puede causar problemas en la determinación del resultado.

Las ventajas de la programación de N-versiones son:

- Capacidad de tolerar fallas
- Mínima sobrecarga en el sistema (en cuanto al tiempo)
- La probabilidad de fallas en modo común se reduce
- Gran autonomía de los procesos

Inconvenientes de la programación de N-versiones:

- Alto costo de implementación
- Se pueden dar fallas similares
- El mecanismo de decisión puede ser muy complicado

Combinaciones y extensiones de los bloques de recuperación y programación de N-versiones

Bloques de recuperación con consenso

Es una mezcla de las dos técnicas arriba mencionadas. Se ejecutan en paralelo N versiones distintas del software, los resultados se votan para entonces determinar el resultado; si existe mayoría (Si no se llega al resultado por mayoría, se hace uso del test de aceptación sobre cada uno de los resultados), se reduce el papel crítico del test de aceptación. Las aplicaciones que pueden generar múltiples resultados son más sencillas, la fiabilidad global es superior a la de las técnicas anteriores. Sin embargo, tiene un alto costo de implementación. Ver Figura 2.3-15.

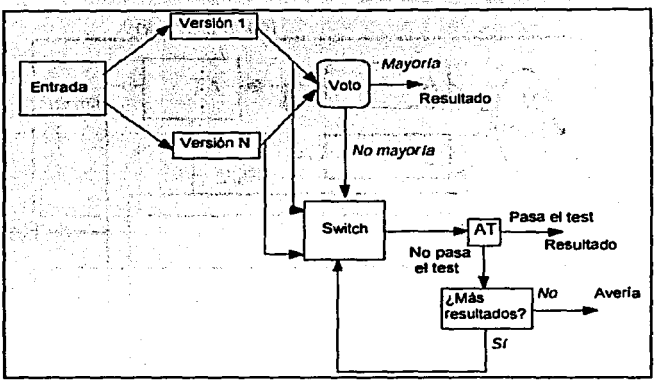


Figura 2.3-15 Bloques de recuperación por consenso.

Bloques de recuperación distribuidos

Dos bloques de hardware (principal y distribuido) trabajan en paralelo ejecutando el mismo proceso. Hay dos versiones de software en cada uno de los módulos de hardware, en uno de los módulos de hardware se ejecuta una de las versiones de software (en el otro se ejecuta la otra), si el resultado de la ejecución de la versión de software del bloque de hardware principal pasa el test de aceptación, ése es el resultado correcto; en caso contrario se conmuta al secundario, que ha ejecutado la segunda versión de software (el primario hace en paralelo regresa y recalcula), en caso de pasar el test, ese es el resultado correcto y para el caso contrario (escenario de falla) se repite-comprueba en el primario el resultado de la segunda rutina, en el último de los casos se volverá a evaluar la misma rutina ejecutada por segunda vez en el módulo secundario (Presentado en la Figura 2.3-16).

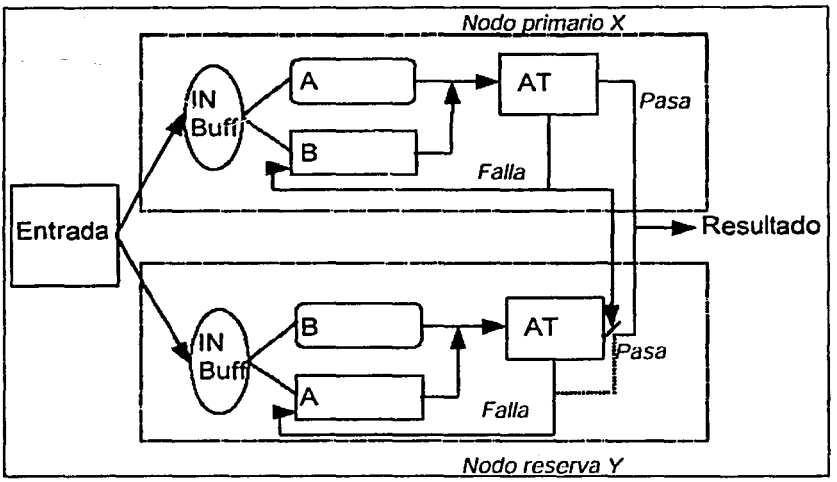


Figura 2.3-16 Bloques de recuperación distribuidos.

Finalmente, la redundancia en el nivel de hardware y software como se describió a lo largo de este capítulo, es fundamental para la disponibilidad de un sistema.

2.4

TECNOLOGÍA CLUSTER

Definición

Un cluster de computadoras se define simplemente como una interconexión de sistemas de cómputo, dispositivos de almacenamiento y periféricos que formando un ambiente integrado, pueden proveer lo que a continuación se describe:

- Que equipos y dispositivos de almacenamiento separados compartan datos, aplicaciones y recursos como si ellos fueran parte de un mismo sistema
- Alta disponibilidad para aplicaciones y datos
- Fácil crecimiento del sistema
- Incremento de la productividad, ya que múltiples equipos pueden ser administrados como un sistema simple además de proveer mayor poder de cómputo. Esto sin la necesidad de más personal.

Los clusters enfatizan la idea de compartir los recursos en por lo menos un par de sistemas independientes (Nodos). Éstos, no se comunican a través de memoria compartida; ellos típicamente se comunican enviando mensajes entre sí a través de un mecanismo llamado cluster interconnect. En contraste con los pares acoplados de sistemas también llamados sistemas de multiprocesamiento simétrico (Symmetric Multiprocessor Systems SMP), que permiten a múltiples CPU's cooperar entre sí a través del uso de memoria compartida. Un sistema SMP podría ser un miembro o nodo de un cluster.

Cada nodo en un cluster opera independientemente de los demás, excepto cuando utiliza un recurso compartido. Cada nodo opera y falla sin afectar o interrumpir el procesamiento que está tomando lugar en otros nodos del mismo cluster.

Un cluster requiere una interconexión entre nodos para ejecutar programas en paralelo. Si se provee un mecanismo de interconexión rápido, se podrá obtener eficiencia significativa. Las nuevas tecnologías de comunicaciones estándar pueden ofrecer un mecanismo de interconexión que provea mucho mayor velocidad en las transacciones.

Mientras los clusters no proveen las características de disponibilidad continua que los tradicionales equipos de tolerancia a fallas sí proveen (Recuperación extremadamente rápida sin interrupción de las aplicaciones del usuario), ellos sí ofrecen dos ventajas significantes sobre equipos tolerantes a fallas:

- Los clusters son más baratos porque:
  - Utilizan hardware y componentes estándar
  - El monto de disponibilidad y desempeño deseado puede ser obtenido fácilmente ya que puede ser comprado en incrementos

- Los clusters pueden proveer algún tipo de detección y tolerancia de fallas de software. Esto depende de las características de cada producto de software instalado en el cluster

El propósito de los clusters de alta disponibilidad es simplemente hacer que un sistema y sus aplicaciones estén altamente disponibles. En la mayoría de los equipos actuales podemos encontrar componentes redundantes. Cuando uno de ellos falla, el sistema puede permanecer disponible para los usuarios aún en ese estado. El objetivo de un servidor confiable es no tener puntos simples de falla. Los sistemas únicos sin embargo, siguen teniendo algunos puntos simples de falla.

Un cluster de alta disponibilidad toma el concepto de servidor confiable y entonces el servidor en sí mismo es duplicado, removiendo de esta forma, todos los puntos simples de falla. En una configuración de este tipo, si un componente no redundante falla en uno de los nodos causando una caída del mismo, la aplicación trabajando simplemente se va a otro servidor.

Por ejemplo, un cluster de dos nodos está corriendo una aplicación de misión crítica y la fuente de alimentación falla en el nodo A, el nodo A se cae; entonces la aplicación y cualquier otro recurso de cluster, serán automáticamente, movidos al nodo B para con esto continuar disponibles.

Una cualidad importante de los clusters es que permiten aumentar la escalabilidad, el desempeño, la fiabilidad y la disponibilidad de los sistemas y / o servicios.

Escalabilidad

La escalabilidad es la capacidad de un equipo para hacer frente a volúmenes de trabajo cada vez mayores sin, por ello, dejar de presentar un nivel de rendimiento aceptable.

Existen dos tipos de escalabilidad:

- Escalabilidad horizontal. Que permite que se puedan adicionar hardware (Servidores) para obtener una ampliación de la misma aplicación
- Escalamiento vertical. Que permite dividir una aplicación en diferentes piezas de manera que la misma permita el crecimiento en forma horizontal

Si un sistema fue diseñado para permitir su escalabilidad horizontal y vertical, esto significa que sumar servidores le permite atender la creciente demanda con un riesgo e impacto mínimo a los usuarios y / o clientes.

Disponibilidad y Fiabilidad

La disponibilidad y la fiabilidad son dos conceptos que, si bien se encuentran íntimamente relacionados, difieren ligeramente. La disponibilidad es la calidad de estar presente, listo para su uso, accesible, mientras que la fiabilidad es la probabilidad de un funcionamiento correcto.

Pero hasta el más fiable de los equipos puede terminar fallando. Los fabricantes de hardware intentan anticiparse a las fallas aplicando la redundancia en áreas clave como son las unidades de disco, las fuentes de alimentación, las controladoras de red y los ventiladores, pero dicha redundancia no protege a los usuarios de las fallas de las aplicaciones. De poco servirá por lo tanto, que un servidor sea fiable si el software de base de datos que se ejecuta en dicho servidor falla, ya que el resultado no será otro que la ausencia de disponibilidad. Esa es la razón de que un sólo equipo no pueda ofrecer los niveles de escalabilidad, disponibilidad y fiabilidad necesarios que sí ofrece un cluster, ya que un cluster puede ejecutar aplicaciones multi-instancia.

Clasificación

En un principio, el implantar un cluster sólo era posible para las grandes empresas en las que una interrupción de sus servicios representaba grandes pérdidas económicas, como es el caso de las instituciones bancarias o de comunicaciones, ya que el costo de los equipos con arquitectura de cluster es elevado. Pero en la actualidad, gracias al bajo costo de los equipos personales y las capacidades que actualmente presentan, se pueden implantar clusters con una inversión menor pudiendo ser una buena solución para las pequeñas empresas.

Teniendo en mente esta idea y de acuerdo al tipo de equipo que forma un cluster, podemos clasificarlos de la siguiente manera:

Clase I

Son sistemas compuestos por máquinas cuyos componentes cumplen con los estándares del mercado, lo que significa que sus elementos son de uso común y pueden ser adquiridos muy fácilmente en cualquier tienda distribuidora. De esta manera, estos clusters no están diseñados para ningún uso ni requerimiento en particular.

Clase II

Son sistemas compuestos por máquinas cuyos componentes no cumplen con los estándares del mercado (Tecnología propietaria), lo que significa que sus componentes no son de uso común y por tanto, no pueden encontrarse con la misma facilidad que los componentes de sistemas de la clase anterior. De tal manera que pueden estar diseñados para algún uso o requerimiento en particular. Las máquinas ubicadas en esta categoría presentan un nivel de prestaciones superior a las de la clase I.

También se tiene una clasificación de acuerdo a su forma de operar. Así, los clusters se pueden dividir en Activo/Activo, Activo/Stand by y Tolerante a Fallas.

Activo / Activo

Todos los nodos en el cluster realizan un trabajo significativo. Si algún nodo cae, el nodo restante (O nodos restantes) continúan realizando su propio trabajo además del trabajo del nodo que se ha caído. En esta situación, el desempeño del sistema se ve disminuido, ya que la carga de trabajo es la misma pero el número de nodos es menor. El tiempo de recuperación está entre 15 y 90 segundos.

Activo / Stand by

Un nodo (El nodo primario) realiza trabajo y el otro espera (Ocioso) a que suceda una caída de este nodo primario. Si el primer nodo falla, la solución de cluster transfiere el trabajo del nodo primario al nodo en espera (Stand by). El tiempo de recuperación está entre los 15 y los 90 segundos.

Tolerante a Fallas

Un cluster tolerante a fallas es un sistema completamente redundante (Disco, CPU, etc.) cuyo objetivo es estar disponible el 99.999% del tiempo. Este objetivo se traduce en menos de 6 minutos fuera de servicio, por año.

Ambos nodos del cluster tolerante a fallas realizan simultáneamente tareas idénticas. El trabajo de los nodos es redundante y el tiempo de recuperación es menor a un segundo.

De igual manera y relacionada con la clasificación anterior, las aplicaciones instaladas o ejecutándose en un cluster, se pueden dividir en tres tipos:

Aplicación de instancia simple

Este tipo de aplicaciones se ejecutan en un sólo miembro del cluster a la vez. Con la finalidad de mantener la alta disponibilidad de la aplicación, el cluster debe proporcionar un mecanismo para reiniciarla en otro de sus miembros en el caso de que el miembro actual no pueda seguir ejecutándola.

Aplicación de instancia múltiple

Este tipo de aplicaciones pueden ser ejecutadas en varios miembros del cluster al mismo tiempo ya que cada uno estará corriendo una imagen (Instancia) de la misma aplicación. Las aplicaciones de instancia múltiple son altamente disponibles, porque la falla de uno de los miembros no afecta las demás instancias de la aplicación ejecutándose en otros miembros del mismo cluster. En caso de que una de estas instancias falle, la aplicación es capaz de trasladar las operaciones pendientes (Operaciones que se estaban realizando al momento de presentarse la falla) por aplicar a las demás instancias activas para que las mismas puedan ser terminadas y no se pierdan.

Aplicación distribuida

Una aplicación distribuida está específicamente diseñada para ser ejecutada en un cluster, asignando a los diferentes miembros un trabajo específico.

En la práctica, la mejor arquitectura de cluster necesita proveer un amplio rango de funcionalidades, desde el nivel más bajo de protección de datos (Cluster de disponibilidad), hasta el nivel más alto de aplicaciones disponibles, dispersas en varios recursos de computo (Clusters de desempeño avanzado). Por otro lado, los cluster de escalabilidad, ofrecen la habilidad de crecer las aplicaciones de una empresa, dividiéndolas entre los nodos de un cluster. De esta forma las tres

características más importantes de una solución de cluster deben ser disponibilidad, escalabilidad y alto desempeño. Figura. 2.4-1.

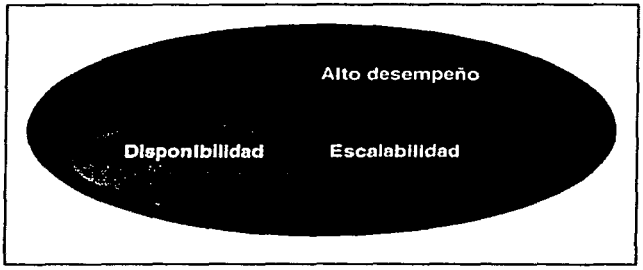


Figura. 2.4-1. Características importantes de Clusters.

Las ventajas y desventajas de cada tipo de diseño de cluster se muestran a continuación.

Clusters de Disponibilidad (Availability Cluster)

Este tipo de cluster es de los más usados, tanto en ambientes UNIX como Windows NT. La funcionalidad de estos clusters se podría definir a partir de a qué grado se pueden automatizar las tareas de recuperación de aplicaciones para minimizar la intervención del administrador del sistema, cuando se presenta una falla.

Ventajas:

- Implantación de bajo costo
- Basados en arquitecturas estándar
- Diseños sencillos

Desventajas:

- Administración duplicada (Para cada miembro del cluster)
- Los procedimientos de recuperación deben ser proporcionados por el administrador del sistema

TESIS CON
FALLA DE ORIGEN

Cluster de Desempeño Avanzado (Advance Performance Cluster.)

El cluster de desempeño avanzado es una tecnología que no sólo proporciona respuestas de entrada y salida de un alto desempeño, sino que también proporciona los beneficios de gran funcionalidad, facilidades de escalamiento mejoradas y mayores recursos de manejo del cluster.

Ventajas:

- Sistemas de archivos de cluster que usan un DLM (Distributed Lock Manager) para sincronizar el acceso a los archivos
- Marcadas características de cluster que hacen que todo el sistema parezca un sistema simple ante los usuarios, aplicaciones y administradores
- Se pueden hacer configuraciones muy grandes
- Una administración del sistema simplificada, esto es, que el administrador sólo tiene que modificar el sistema una vez

Desventajas:

- Costos elevados

Cluster de Escalabilidad (Scalability Cluster)

Un cluster de escalabilidad permite a las aplicaciones en múltiples nodos tener un acceso coordinado a un disco de datos compartido. Esencialmente la alta disponibilidad y la habilidad de escalar aplicaciones por medio de nodos, son las principales ventajas de este cluster. El ambiente UNIX es el que predomina en estos clusters.

Ventajas:

- Se pueden ampliar las aplicaciones al distribuirlas en más nodos

Desventajas:

- El ambiente de UNIX está muy orientado a aplicaciones de bases de datos
- Está limitado el número de nodos (Actualmente ocho) y de sistemas de almacenamiento que se les puede agregar

Efectos de un cluster en la disponibilidad

Existen situaciones en las que no se puede evitar la caída del sistema, esto es donde sólo se cuenta con un servidor. Si existe el riesgo de pérdidas significantes a causa de una caída del sistema, se debe usar un cluster de alta disponibilidad. Un cluster de dos nodos provee disponibilidad excepcional ofreciendo redundancia en el nivel de servidor y en el nivel de aplicación.

La Figura 2.4-2 compara el downtime de:

a) Un servidor
b) Un cluster de alta disponibilidad de dos nodos
c) Un cluster paralelo multi nodo

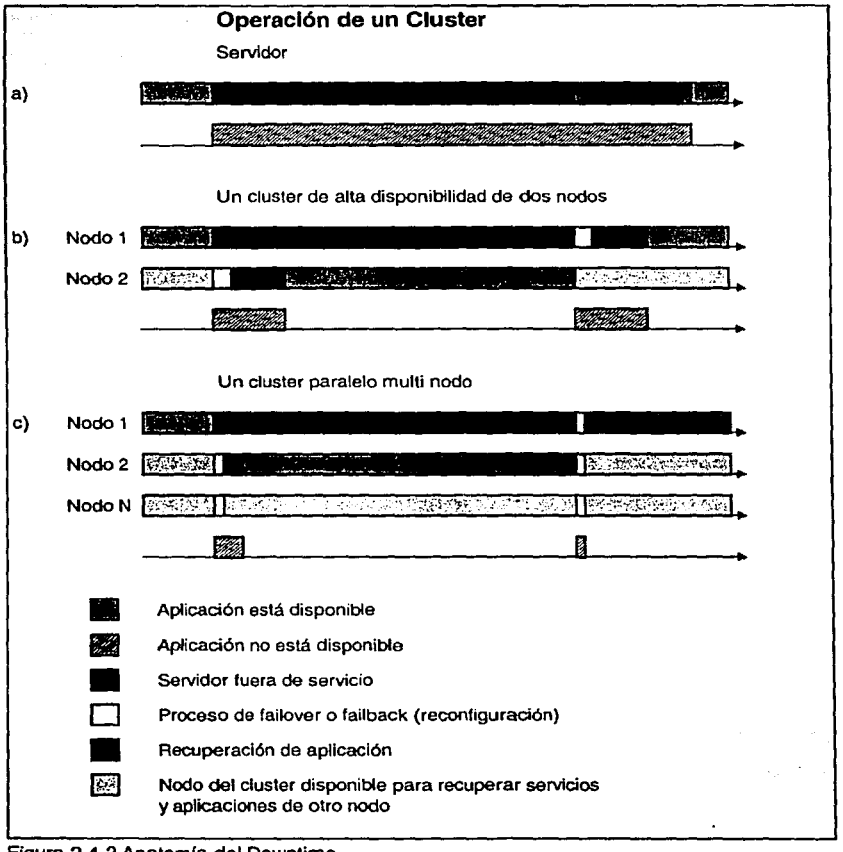


Figura 2.4-2 Anatomía del Downtime.

En un cluster de alta disponibilidad, si uno de los nodos falla la aplicación que se encontraba en este nodo se 'transfiere' al otro nodo como resultado del proceso de recuperación de la aplicación (Failover). Aunque se presenta la falla en el nodo, la comunicación de los clientes con la aplicación usualmente continúa con

una pequeña interrupción. En la mayoría de los casos, la interrupción del servicio es detectada en 5 segundos y los servicios se encuentran disponibles nuevamente en un promedio de 30 segundos (Dependiendo de cuanto tarda en reiniciar la aplicación).

Un cluster de dos nodos provee un downtime mucho más corto que un servidor simple. Si observamos las diferencias del downtime de las opciones "a" y "b" de la Figura 2.4-2, descubriremos que el "tiempo para reparar el servidor" más el "tiempo para levantar el sistema operativo", es usualmente mucho más grande que el "tiempo para transferir las aplicaciones al otro servidor disponible". Es importante mencionar que el "tiempo de recuperación de las aplicaciones", se incluye en el downtime en ambos casos (Caso del servidor simple y caso de un cluster de dos nodos).

Por otra parte, en un cluster paralelo, si uno de los nodos falla o se pone fuera de servicio por un mantenimiento programado, otro de los nodos del cluster continúa en operación normal. Al mismo tiempo, se provee un proceso de reconfiguración de las aplicaciones y los usuarios son transferidos automáticamente a una de las instancias de la aplicación que se encuentra en los nodos funcionales. Este proceso no requiere el reinicio de la base de datos o de la aplicación. En comparación con el cluster de dos nodos, un cluster de este tipo ofrece para el usuario final una interrupción mucho más pequeña en los servicios.

Más adelante, en el capítulo tres profundizaremos en los detalles técnicos y las funcionalidades específicas de un cluster de alta disponibilidad.

2.5

TOLERANCIA A FALLAS (FAULT - TOLERANCE)

Los sistemas tolerantes a fallas pueden proporcionar dos ventajas únicas:

- Procesamiento de transacciones en forma "Nonstop", donde el procesamiento continúa aún cuando un componente falla, o cuando un equipo está siendo reparado o reemplazado, o mientras nuevos procesadores o dispositivos periféricos están siendo agregados al sistema

- Multiprocesamiento independiente, con muchos procesadores corriendo simultáneamente pero independientemente. Aunque otras máquinas han incluido mas de una Unidad de Procesamiento Central (CPU), solamente los sistemas tolerantes a fallas proporcionan Unidades de Procesamiento Central (CPUs) en las que cada una, tiene su propia memoria y su propio software de sistema operativo; Y pueden así, operar como computadores individuales dentro del sistema. En estos sistemas, el

multiprocesamiento toma lugar concurrentemente con la multiprogramación en cada procesador

A través de los años, varios sistemas de cómputo han sido adaptados para correr aplicaciones de procesamiento de transacciones. Muchos de esos sistemas, por naturaleza, soportan servicios en línea para esas aplicaciones. Pero los sistemas de cómputo tolerantes a fallas están específicamente diseñados para adaptarse al procesamiento de transacciones en línea. Por lo tanto, estos sistemas concentran su atención en los requerimientos básicos de las aplicaciones de procesamiento de transacciones.

Uno de los requerimientos más importantes es que el sistema de cómputo debe permanecer disponible continuamente durante el tiempo en que las transacciones están siendo incorporadas al sistema. Para minimizar o eliminar estos problemas, los sistemas tolerantes a fallas procuran desarrollar sistemas que soporten la operación continua del sistema y hagan posible el "procesamiento de transacciones sin interrupción". Las principales funcionalidades proporcionadas por un sistema de este tipo son:

- Las aplicaciones deben continuar ejecutándose aún si un componente falla. El mal funcionamiento de un componente de hardware no deberá detener la operación del sistema, ni dañar la aplicación

- El sistema deberá continuar operando mientras el componente dañado, procesador, fuente de poder, controladores de entrada / salida o "buses" están siendo reparados o reemplazados. Y una vez que esto es efectuado, el sistema deberá reasumir el uso del componente reparado sin interrumpir el trabajo de la aplicación en línea

- El sistema debe continuar en procesamiento mientras nuevos componentes de hardware o programas de aplicación están siendo adicionados, o los existentes están siendo removidos. Además, estos cambios deberán ser posibles sin modificar los componentes que ya están presentes

Estas tres características son algunas veces referidas como "tolerancia a fallas", "reparación en línea" y "creciendo modular" respectivamente.

Trayectorias de datos y hardware múltiple para respaldo del sistema

La duplicación de los componentes de hardware a través del sistema, en una distribución que evita los puntos de control únicos, ayuda a prevenir las fallas de un solo componente que detienen el sistema. Estos componentes operan independientemente y concurrentemente con otros. Están distribuidos de tal manera que si uno falla, otro puede tomar su lugar como su respaldo. Por ejemplo, cada sistema tolerante a fallas cuenta con por lo menos dos módulos de procesador, controladores múltiples, trayectorias de datos múltiples entre los procesadores y los controladores de entrada / salida, y fuentes de poder múltiples. Los módulos de los procesadores están conectados uno con otro, a través de un "bus" ínter procesador que proporciona dos trayectorias de comunicación entre dos módulos de procesador. Cada una de esas trayectorias transmite datos en altas velocidades entre los módulos de procesador en forma independiente y simultánea. Cada controlador de dispositivos está conectado a través de sus dos puertos de entrada / salida, a dos módulos de procesador.

El sistema responde a fallas de componente asignando recursos para compensar la emergencia. Por ejemplo cuando un módulo de procesador o trayectoria de datos falla, un módulo de procesador de respaldo o una trayectoria alterna de datos asume automáticamente el control. A pesar de esta

duplicación de hardware, el sistema continuamente usa todos los módulos de procesador y todas las trayectorias de entrada / salida de datos para el procesamiento de las cargas de trabajo – no hay módulos de procesador de respaldo o trayectorias de datos de entrada / salida que funcionen solamente cuando una falla ocurre en el sistema. Todos los componentes del sistema operan completa y concurrentemente en todo tiempo. Un sistema tolerante a fallas se compone mínimo de una configuración con dos módulos de procesador, un "bus" ínter procesador, una unidad de disco, una unidad de cinta y una terminal de usuario, cada una conectada a un controlador de dispositivos de dos puertos.

En un sistema de este tipo, cada módulo de procesamiento a su vez, incluye los siguientes componentes: Unidad de Procesamiento Central, memoria, canal de entrada / salida e interfase del "bus" ínter procesador.

Estos componentes permiten a cada módulo de procesador operar independientemente de, pero concurrente con, todos los módulos de procesador en el sistema.

La Unidad de Procesamiento Central (CPU) ejecuta programas al traer instrucciones de la memoria y procesarlas. Efectúa todos los cálculos aritméticos. Requiere solamente una instrucción para enviar datos desde la memoria de un módulo de procesador a través del "bus" ínter procesador a la memoria de un módulo de procesador y un dispositivo de entrada / salida por medio del canal de entrada / salida.

La memoria Principal contiene las instrucciones ejecutadas por cada CPU así como los datos que le pertenecen y que pueden ser movidos hacia y desde memoria virtual en disco. Debido a que la comunicación entre módulos de procesador toma lugar sobre los "buses" ínter procesador, ningún CPU

comparte su memoria con otro CPU. Esta característica, a su vez, elimina un punto donde una sola falla podría detener el sistema.

Cada módulo de procesador está conectado con los demás por el "bus" ínter procesador, el cual a su vez es un par de "buses" de alta velocidad duplicados. Cada "bus" es totalmente autónomo, operando independiente y simultáneamente con el otro "bus".

El uso de los dos "buses" asegura que dos trayectorias de datos existen entre todos los CPU del sistema. Si un bus falla, toda la comunicación de ínter procesadores es automáticamente enrutada sobre el "bus" en operación.

Debido a que cada "bus" está controlado por su propio controlador y cada controlador es independiente de los circuitos lógicos dentro del CPU, ninguna falla de CPU puede interrumpir la transmisión del "bus".

Mientras los procesos de aplicación están siendo ejecutados, cada CPU periódicamente envía un mensaje sobre el bus ínter procesador. A su vez, cada CPU conectado al bus, periódicamente estará revisando la recepción de esos mensajes provenientes de los demás.

Si el sistema operativo en un CPU no recibe un mensaje desde alguno de los otros CPUs, asume que tal CPU tiene un mal funcionamiento. El sistema operativo entonces envía un mensaje de CPU caído a los programas pertinentes. De este modo un proceso corriendo en un CPU es informado si otro CPU falla.

Los datos son transferidos entre un CPU y un dispositivo de entrada / salida sobre un canal de entrada / salida. Los dispositivos están conectados al canal a través de controladores "dual-port". El término "dual-port" significa que cada controlador está conectado a los canales de entrada / salida de dos CPUs, de

modo que el sistema realmente tiene trayectorias de comunicación redundantes a cada dispositivo. El controlador es manejado por un solo CPU pero en el caso de que este CPU falle, el otro puede tomar el control inmediatamente.

El uso de controladores "dual ported" elimina la necesidad de conmutar de "bus" los dispositivos que podrían permitir que la falla en un solo punto detuviera el sistema.

Puesto que la información que reside en discos es crítica, cada unidad de disco tiene puertos duales que, en combinación con sus controladores, permiten configurar las unidades para tolerar cualquier nivel de falla. La más alta salvaguarda contra una falla de este tipo es suministrada implementando redundancia de volúmenes de disco a través de cualquier tecnología disponible para tal fin.

En los sistemas tolerantes a fallas, la energía eléctrica es distribuida para soportar la operación tolerante a fallas. Cada módulo de procesador se alimenta de su propia fuente de poder independiente. Además los controladores dual-ported reciben energía de dos fuentes de poder. Si una fuente de poder falla, dejando a un módulo de procesador inoperativo, el módulo de procesador alterno o de respaldo toma el control. En esta clase de falla de módulo de procesador, el módulo de procesador de respaldo consume la mitad de la potencia disponible de su fuente de poder, ofreciendo el remanente para ayudar a manejar los controladores de dispositivos.

En algunos casos, en efecto, la potencia disponible es suficiente para alimentar a todos los controladores de dispositivos. En otros casos, una fuente de poder suplementaria es requerida para entrada / salida únicamente. Aun, si la falla de energía ocurriera, los contenidos de memoria pueden ser mantenidos por periodos cortos de tiempo. El tiempo que los contenidos de memoria pueden ser mantenidos depende del número y tipo de módulos de memoria incluidos en el

procesador. La fuente de poder retiene suficiente energía para permitir que el sistema suspenda sus actividades en forma ordenada, de modo que ninguna transacción se pierda durante la falla de potencia o sea duplicada cuando la potencia retorne.

Sistema operativo para soportar operación continua

El software de soporte primario para todos los sistemas de cómputo tolerantes a fallas, es suministrado por un sistema operativo capaz de controlar y reaccionar ante las fallas. Una copia de los programas del sistema operativo reside en cada módulo de procesador, lista para responder a una falla en cualquier lugar del sistema. Esta característica no solamente contribuye a la operación continua del sistema, sino que permite a cada CPU actuar como computador individual que funciona independientemente de los otros dentro del sistema. Cada copia del sistema operativo es autónoma y no requiere software de control para coordinarse con las otras copias, además hace posible la "multiprogramación" al permitir que varios programas intercalados se ejecuten en el mismo CPU. Garantiza también que estos programas se ejecuten como unidades independientes que no interfieran una con otra, aun así, comparte el CPU eficientemente. El "Multiprocesamiento" puede también tomar lugar simultáneamente.

Las principales tareas que un sistema operativo de este tipo debe realizar son:

- Permitir a los procesos de aplicación ejecutándose sobre un CPU iniciar otros procesos, en el mismo CPU o en cualquier otro
- Permitir a los procesos comunicarse con otros, a pesar de los CPUs sobre los cuales, ellos estén ejecutándose

- Permitir a los procesos acceder todos los dispositivos físicos, a pesar de los CPUs en los cuales estos dispositivos están conectados
- Asignar recursos entre procesos en ejecución, de modo que cada proceso parece tener los recursos en el sistema disponibles para él
- Proporciona soporte de software para dos actividades que son las más vitales para la operación continua del sistema: las trayectorias alternas de entrada / salida y chequeo del estado de los CPUs

La operación continua en el nivel aplicación se logra a través de "pares de procesos": un proceso primario ejecuta la aplicación, mientras un proceso secundario en otro CPU, programado para recibir mensajes de "checkpoint" periódicos sobre el estado del primario, permanece listo para tomar control si el primario falla. En ese evento, el proceso de respaldo reasume el trabajo en el punto del último "checkpoint" válido. El miembro primario de un par de procesos es el miembro activo, siempre efectuando las funciones de la aplicación que se requieren. El proceso de respaldo es una copia pasiva del primario, que atiende a mensajes procedentes del primario. Antes de que el primario efectúe cualquier función crítica tal como actualizar una base de datos, el primario envía su mensaje de "checkpoint" al respaldo.

En resumen, la tolerancia a fallas en los sistemas de cómputo está basada en el uso selectivo de componentes duplicados de hardware y software. El sistema operativo, incluye procesos que interactúan a través de mensajes para llevar a cabo la revisión de todos los componentes del sistema y reaccionar de forma eficiente ante las fallas.

2.6

RECUPERACIÓN DE DESASTRES (DISASTER RECOVERY)

Llamamos disaster recovery a la capacidad de recuperación rápida de un desastre en el menor tiempo posible. Esto es mayormente alcanzado a través del uso de medios de respaldo en una segunda localidad físicamente separada del centro de cómputo primario El tiempo de recuperación puede variar entre un rango de minutos, días o hasta semanas. Cuando se habla de recuperación de desastres o de tolerancia al desastre es conveniente recordar que los mejores planes y la mejor tecnología del mundo no garantizan nada si no se implementan cuando son necesarias. Un ejemplo clásico es el caso en el que una compañía tiene un elaborado sistema para respaldar sus archivos en una forma regular, así como una política implementada para solicitarla cuando sea necesaria. Pero una falla al implernentar el procedimiento en un momento inadecuado puede causar pérdidas irreparables.

Tolerancia al desastre

La tolerancia al desastre es la práctica de duplicar las operaciones en un primer centro de cómputo – información, aplicaciones y todos los elementos de tecnología necesarios –en otra locación, permitiendo así operación continua o bien casi continua. Es un caso especial de alta disponibilidad en el cual el punto único de falla final en una infraestructura de tecnología de información – la pérdida del centro de cómputo por sí mismo – es eliminado.

Durante el período normal de procesamiento de datos en un sistema de tolerancia al desastre, la información es escrita de forma simultanea tanto en el centro de cómputo local como en el remoto. Idealmente un sistema tolerante al desastre debería enmascarar los mismos, de tal suerte que los clientes nunca percibieran alguna interrupción en la disponibilidad de su aplicación. En la práctica los clientes pueden esperar el restablecimiento del servicio de su aplicación después de un tiempo razonable, comúnmente minutos y con una mínima o ninguna pérdida de información.

El centro de cómputo secundario proporciona ya sea un respaldo de tipo "hot" o bien de tipo "warm". "Hot" significa que el segundo centro esta operando en tiempo real la información y la aplicación mientras se presenta una falla. "Warm" significa que esta listo y esperando, pero para ser activado podría requerir la puesta en marcha de otro proceso.

Aspectos Decisivos

Los conceptos de recuperación de desastres y tolerancia al desastre, tienen que ver con los aspectos decisivos que influyen en el nivel de continuidad de negocio que la organización puede necesitar. No todos los negocios requieren una total tolerancia al desastre. Si se aspira a incrementar o mantener la

continuidad en las soluciones del negocio, se debería entonces tomar decisiones basadas en factores tales como:

- ¿Tipo de negocio?
- ¿Cuales son los riesgos comparados con las consecuencias?
- ¿Cuál es el costo del downtime?
- ¿Que esta dispuesta a hacer su organización?
- ¿Quién es el responsable de cada actividad y como deberá ser ejecutada esa responsabilidad?

Existen varias soluciones a la recuperación de desastres que van del rango de simples respaldos hasta configuraciones de cluster y aplicaciones distribuidas.

Las soluciones pueden variar debido a:

- Costo
- Tiempo de recuperación
- Punto de recuperación
- Distancia

Costo

El costo de una solución individual debe ser comparado contra los efectos de downtime creados por un desastre. La mayoría de las organizaciones subestiman – o bien no tienen calculado - el costo de downtime de su negocio. De esta forma, el punto de inicio para cualquier discusión de disponibilidad debería ser el costo de la caída del sistema de información de la organización. Mientras más alto sea el costo del downtime, la solución deberá ser más robusta. Y mientras más efectivo sea el ambiente en entregar el nivel de disponibilidad requerido por la organización, más rápido será el retorno de la inversión.

Punto y tiempo de recuperación

El tiempo de recuperación es la medida de que tan rápido la operación se recupera después de una falla. En la planeación de un ambiente de tolerancia a desastres es importante hacerse las siguientes preguntas: ¿Cuantas transacciones pueden ser ignoradas y perdidas?, ¿Cuál es el impacto a largo plazo de la ausencia de disponibilidad del sistema?, etc. Si el costo de oportunidad de una interrupción es mayor que el costo de la solución, no cabe duda que es más prudente y menos costoso implementar la solución con el tiempo de respuesta a fallas más corto.

El punto de recuperación es la medida de la cantidad de información perdida o corrupta como resultado de un incidente. ¿Se pueden perder los últimos 10 minutos de información? ¿Que hay sobre la última hora o día? ¿Que tan precisa debe ser la base de datos? Un archivo por si solo debe ser estable – no en el proceso de ser cambiado - sino en el proceso de ser modificado. Debemos recordar que muchas de las aplicaciones modifican múltiples archivos a la vez. Para tener un punto de recuperación seguro, los archivos relacionados con una misma aplicación deberán ser respaldados al mismo tiempo (Sincronización de tiempos de backup).

Distancia

Un sistema tolerante a desastres esta formado por dos o más centros de cómputo separados por una distancia física. Esta distancia es generalmente determinada por el alcance que la solución tiene respecto del tipo de desastres para los cuales se desea estar prevenido. Puede ir desde cientos de metros en un mismo campus para prevenir un incendio hasta miles de kilómetros para prevenir un terremoto.

La distancia juega un rol muy importante ya que de eso depende la definición del tipo de interconexión del cluster (Cluster interconnect). Distancias grandes significan grandes retardos, los cuales pueden hacer que el procesamiento síncrono se vuelva un problema de desempeño. Aunque se incremente el ancho de banda ilimitadamente a una configuración, el incrementar la distancia en la cual los datos deben viajar siempre incrementará la latencia lo cual se traduce en serios retrasos de las operaciones de I/O que las aplicaciones realizan. A mayor distancia requerida, se necesitará implementar alguna forma de espejeo asíncrono para evitar el retardo.

La replicación síncrona mantiene los datos consistentes en términos de contenido, calidad y tiempo. Si los datos son actualizados, la actualización será aplicada inmediatamente a todas las demás copias. Este tipo de replicación es típicamente usado cuando la pérdida de datos no es tolerada después de un desastre. La replicación asíncrona permite que las múltiples copias de datos temporalmente se encuentren fuera de sincronización entre ellas. Si la copia original es actualizada, el cambio finalmente se propagará y todas las copias convergerán en la misma información pero esto puede ocurrir en segundos, minutos y hasta horas como un paso secundario.

La Figura 2.6-1 muestra la solución tolerante a desastres más frecuentemente usada.

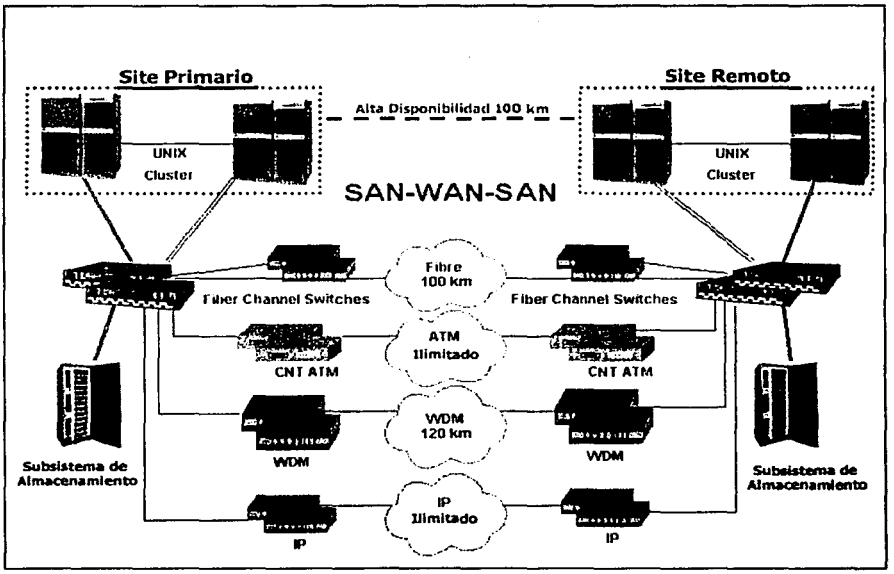


Figura 2.6-1 Solución Típica de Recuperación de Desastres

2.7

ANÁLISIS COMPARATIVO

Mucha gente cree que los conceptos de cluster, tolerancia a fallas y recuperación de desastres se refieren a lo mismo, sin embargo, no es así.

Cada uno, corresponde a una tecnología diferente, pero es probable que en algunos casos se lleguen a complementar entre ellas. Cuando se combinan correctamente puede surgir una solución empresarial robusta. Brevemente definidos, la tolerancia a fallas generalmente se refiere a las fallas que ocurren dentro del ambiente de control de un sistema o aplicación. Por otro lado, la recuperación de desastres implica un procedimiento para recuperar el servicio en un sistema después de una interrupción del mismo, de una aplicación o del ambiente en el cual ellos residen. Entendemos desastre cuando nos referimos a una catástrofe mayor que ocurre fuera del ambiente de control del sistema o aplicación.

Combinando la tecnología de tolerancia a fallas con la de cluster, la fiabilidad de cada nodo y subsistema de discos se incrementa, o bien, implementando una solución de cluster que espejee el subsistema de almacenamiento en lugar de

compartirlo; los nodos pueden estar geográficamente separados para crear una solución de recuperación de desastres.

Enseguida, en la Tabla 2.7-1 presentaremos un cuadro comparativo que permitirá identificar y contrastar las principales características de estas tecnologías.

Tecnología	Recuperación de Desastres	Tolerancia a Fallas	Cluster
Uptime	99.999%	99.99%	99.9%
Tiempo de recuperación	Minutos a horas	Minutos	Minutos
Manejo de la falla	Failover remoto	Redundancia	Failover
Redundancia	Sin puntos simples de falla	Posibles puntos de falla	Posibles puntos de falla
Desempeño	Dependiente de la plataforma	Dependiente de la plataforma	Dependiente de la plataforma
Intervención humana	Mínima o ninguna	Ninguna	Ninguna
Nivel de disponibilidad	AE-5	AE-4	AE-3
Número de centros de cómputo	Dos o más	Uno	Uno
Distancia entre nodos	Metros a miles de kilómetros	Metros	Metros
Política de recuperación	Lenta y manual o automática (Dependiente del costo)	Rápida y automática	Rápida y automática
Costo	Muy alto	Moderado a muy alto	Moderado

Tabla 2.7-1 Cuadro comparativo.

3

TECNOLOGÍA DE CLUSTER

3.1

HISTORIA DE LA TECNOLOGÍA CLUSTER

La idea de la tecnología cluster surge en 1960 en la empresa IBM como una manera de unir grandes mainframes para proporcionar un costo-efectivo (Cost-effective) del paralelismo comercial. En aquella época, los sistemas HASP (Houston Automatic Spooling Priority) y su sucesor, el JES (Job Entry System) proporcionaron una solución para el trabajo distribuido.

IBM estableció soporte con clusters de mainframes a través de su sistema *parallel Sysplex*, el cual permitió a equipos de tamaño mediano, al sistema operativo, y a las aplicaciones del sistema proporcionar un gran desempeño y un mejoramiento del costo, permitiéndole a los usuarios de los grandes mainframes dar continuidad en la ejecución de sus aplicaciones.

En la década de los 80's, la tecnología cluster se diversificó presentando tres tendencias: Procesamiento de alto desempeño, Computación distribuida y Computación de alta disponibilidad.

Proyectos como el HPVM (High Performance Virtual Machine) y el Beowulf[1] son consecuencia de estas diversificaciones.

A principios de la década, Digital Equipment Corporation (DEC) aplicó el concepto de la tecnología cluster a la alta disponibilidad inventando y desarrollando el producto llamado VAXCluster. Esto, provocado por la necesidad cada vez más creciente de mayor poder de cómputo en áreas como la investigación y las aplicaciones comerciales, al alto costo y al bajo acceso a las supercomputadoras tradicionales y a la necesidad de proporcionar mejores y más altos niveles de disponibilidad en el mercado. De esta forma el concepto de agrupar varios sistemas para obtener mayor desempeño fue también usado para obtener mayor disponibilidad.

Los recientes avances y la disponibilidad de componentes cada vez más baratos con los que se implementan los cluster y las redes de cómputo (PC's[2], estaciones de trabajo, SMP's[3]) ayudan a disminuir el costo-beneficio de la tecnología. Los clusters que pueden ser construidos usando componentes estándares de hardware y software, desempeñaron un papel muy importante en la redefinición del concepto del supercómputo.

Los grandes avances en la tecnología de software representaron de igual manera un motor importante para este fin. Permitieron el desarrollo y despliegue de aplicaciones utilizadas en la ciencia, ingeniería y necesidades comerciales.

Desde entonces el concepto ha sido mejorado y desarrollado por la mayoría de los proveedores de cómputo y servicios del mercado.

[1] El objetivo de estos proyectos fue entregar desempeño de tipo supercómputo en sistemas de bajo costo y construidos a partir de componentes estandar.
[2] Computadoras Personales.
[3] Symmetric Multiprocessors.

Como consecuencia del aumento de corporativos con aplicaciones de misión crítica la tecnología ha proporcionado una solución de fácil acceso, relativamente fácil de implementar, de costo aceptable y ampliamente utilizada en el mercado como una de las soluciones de la cual se puede echar mano para incrementar los niveles de disponibilidad de un sistema de misión crítica.

3.2

DESCRIPCIÓN TÉCNICA DE UN AMBIENTE DE ALTA DISPONIBILIDAD

El ambiente de servidores altamente disponibles fue diseñado para satisfacer la demanda de disponibilidad en las aplicaciones de misión crítica. En nuestro caso, nos referiremos indistintamente a un ambiente de servidores altamente disponibles o a un cluster.

El concepto de disponibilidad que provee un cluster es simple, si dos o más nodos pueden accesar los mismos datos y uno de ellos falla, entonces cualquier otro nodo deberá estar disponible para continuar el acceso, haciendo de esta manera que las aplicaciones que usan esos datos se encuentren disponibles. La alta disponibilidad que un cluster ofrece se encuentra en la mitad del espectro de soluciones, en algún punto entre los costosos sistemas tolerantes a fallas y un muy bien administrado y relativamente económico sistema de cómputo simple.

Eliminando los puntos simples de falla del hardware, el ambiente se torna más disponible. El objetivo de un ambiente de alta disponibilidad es el de detectar

fallas en torno a todos los componentes del mismo (Servidores, dispositivos de almacenamiento, red, etc.) para entonces dinámicamente reconfigurar el ambiente del sistema y llevarlo al estado normal de operación.

Por lo tanto un cluster reduce el riesgo por falla de hardware o software que una aplicación tiene de no estar disponible al ser ejecutada en un servidor. Un ASE[4] detecta fallas del sistema solamente, no puede detectar fallas de software; por ejemplo de una aplicación o de una corrupción de datos. Sin embargo y debido a la intercomunicación entre los servidores miembro de un cluster que se lleva a cabo al menos por una de las rutas redundantes que serán descritas más adelante, si se puede recuperar de fallas de software. En algunos casos, el servidor que ha sufrido la falla puede (Si no esta completamente caído) iniciar el proceso de recuperación por sí mismo. En otros casos, el servidor dañado no podrá hacer nada debido a que se encuentra caído completamente. Entonces, cualquiera otro servidor sobreviviente del ASE reaccionará apropiadamente realizando el failover de los servicios necesarios para continuar la operación lo más pronto posible.

Servicio

Un servicio es una aplicación provista a los clientes (Bases de datos, servidor de mail, etc.) por el cluster. Tales aplicaciones corren continuamente en un sistema de cómputo proveyendo beneficios a los clientes que comúnmente entran al servidor para usarlas. En un ASE cada servicio tiene un nombre único, mismo que el administrador puede usar para manejarlo dentro de la infraestructura del cluster. Los clientes pueden y por conveniencia deben usar este nombre cuando realizan peticiones a la aplicación ya que de esta manera ellos no necesitan saber cual servidor está actualmente proporcionando el servicio. Por ejemplo,

[4] ASE, por sus siglas en ingles **Available Server Environment.**

los mensajes que un usuario envía a la dirección *usuario@servidor1.site.com* serán retrasados cuando el servidor1 se encuentre caído por reparación o falla mientras que los enviados a la dirección *usurario@correo.site.com* no se verán afectados debido a que otro servidor del ASE estará proveyendo el servicio llamado correo.

Failover o Recuperación de un Servicio

El software del ASE responde a los eventos de falla, relocalizando los servicios de un nodo a otro. Una relocalización provocada por una falla de hardware es referida como *failover*. Hay otras razones además de las fallas para iniciar el proceso de relocalizar un servicio. Por ejemplo, el administrador del sistema puede relocalizar un servicio por razones de balanceo de carga de trabajo o al momento de dar de baja un nodo por razones de mantenimiento.

Política de Relocalización de Servicios

Siempre que un servicio debe ser relocalizado, el ASE utiliza políticas configurables para determinar cual de los nodos restantes es el más apto para hospedar y ejecutar el servicio. Las política es una función del evento ocurrido y de las preferencias definidas por la administración del sistema para cada servicio. Por ejemplo, un servicio debe ser relocalizado si el nodo en el cual está corriendo se cae o si el cable SCSI es desconectado. El administrador del sistema puede especificar el nodo hacia el cual el servicio debe ser relocalizado (Política llamada "balanced member", el servicio siempre tiene un nodo alterno en el cual puede ser ejecutado por razones de falla o de balanceo de carga de trabajo.). También es posible proporcionar preferencias específicas para obtener un comportamiento de recuperación definido. Por ejemplo, el encargado del sistema puede especificar que un servicio retorne siempre a su nodo

especificado si ese nodo vuelve a estar disponible (Política llamada "Restricted member", el servicio siempre será ejecutado en el nodo definido cuando el mismo se encuentre disponible.). Para servicios que toman largo tiempo en su inicialización, el administrador del sistema puede especificar que un servicio sea relocalizado solamente si su nodo falla (Política llamada "favored member", siempre existe un nodo favorito y a menos que el mismo no se encuentre disponible el servicio será ejecutado en otro nodo.).

Existen dos componentes importantes en un cluster, el administrador de distribución de discos remotos (DRD Distributed Remote Disk) y el administrador de acceso distribuido (DLM Distributed Lock Manager).

Subsistema de distribución de discos remotos

Este subsistema fue desarrollado con el objetivo de presentar una vista a nivel cluster de los discos conectados a él. El DRD provee un espacio de nombres para el cluster y un mecanismo de acceso físico y lógico hacia los volúmenes. Ofrece una forma confiable y transparente de accesar remotamente cualquier disco conectado al cluster desde cualquier miembro del mismo.

Administrador de acceso distribuido

El administrador de acceso distribuido provee servicios de sincronización apropiados para soportar acceso de forma paralela a los datos. Las aplicaciones pueden usar candados para controlar el acceso a copias distribuidas de datos o limitar el acceso concurrente a dispositivos compartidos tales como aquellos provistos por el subsistema de DRD. Cuando un proceso solicita aplicar un candado sobre cierto recurso, tal solicitud es negada o autorizada tomando como base los candados previamente impuestos a ese mismo recurso.

Elementos que intervienen en la operación de un cluster

En la operación del cluster intervienen varios elementos, los cuales pueden ser subsistemas, controladores, scripts y archivos de configuración que se integran para proporcionar la funcionalidad de los recursos del cluster. A continuación describimos brevemente cada uno de esos elementos.

1. Agente (Agent daemon), supervisa un servidor. Cada miembro del cluster ejecuta una instancia de este programa. El Agente mantiene una vista, casi en tiempo real, del estado actual del servidor en que se encuentra y lo comunica al proceso Director. Deduce el estado de disponibilidad de un servicio a partir de información recibida del Monitor de Estado del Servidor (Host Status Monitor). Inicia y detiene los servicios en un nodo bajo la dirección del Director.

2. Monitor de Estado del Servidor (Host Status Monitor daemon), Se encarga de monitorear el estado de los demás servidores así como el de las interfaces de la red local, el cual puede ser arriba o abajo (up / down) reportando cualquier cambio al proceso Agente. Cada nodo debe ejecutar una instancia del programa.

3. Director (Director daemon), tiene una visión global del estado de los servicios proporcionados por los nodos del cluster. Este se encarga de asignar los servicios a los nodos de acuerdo al estado de disponibilidad del mismo, respetando preferencia y manteniendo las políticas definidas por el administrador del sistema. Solo hay una instancia del Director ejecutándose en el cluster.

4. Logger (Logger daemon). Los mensajes provenientes de todos los subsistemas del cluster son dirigidos hacia este elemento, el cual se encarga de escribirlos en las diversas bitácoras de eventos del sistema.

Una instancia del Logger podrá ejecutarse en cada una de los nodos del cluster.

5. Manejador de disponibilidad (Availability manager), es un pseudo-controlador que se encuentra en una capa superior al controlador del sistema básico SCSI CAM del servidor. Implementa funciones requeridas por el Monitor de Estado del Servidor que no se encuentran en el sistema base para soportar el envío de mensajes servidor-a-servidor vía una ruta SCSI, además reporta las fallas ocurridas en los subsistemas de I/O a los demonios del cluster.

6. Una base de datos binaria en la cual se registran todas las características de cada uno de los elementos, servicios y miembros del cluster.

7. Los scripts de acción, que son la interfaz entre el cluster y los servicios definidos. Una aplicación bajo el control del cluster, requiere de por lo menos un script de alta y un script de baja de los procesos que la conforman.

La Figura 3.2-1 Nos muestra la relación existente entre los componentes antes mencionados.

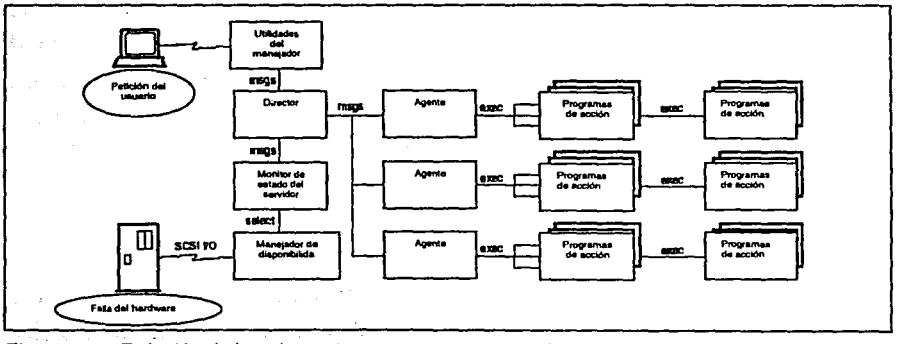


Figura 3.2-1 Relación de los elementos que componen un cluster

Monitoreo y tipos de falla

La parte más importante e interesante del código del software de cluster es aquella que se encarga de monitorear el correcto funcionamiento de los servidores miembro, además de determinar como reaccionar cuando el status de alguno de ellos parece cambiar. La lógica para la toma de decisiones depende del uso de dos rutas de comunicación redundantes entre los servidores. La red primaria es una de ellas y el o los buses SCSI compartidos permiten establecer un segundo camino de acceso

Cada miembro que integra al cluster esta conectado a todos los demás vía ambas rutas de acceso. El proceso llamado Host Status Monitor utiliza estas rutas de la siguiente forma: Periódicamente un servidor envía un comando SCSI "send" a otro servidor a través de sus respectivos adaptadores SCSI, entonces el servidor destino contesta a su vez con un comando "send" en sentido contrario. Esta operación permite el intercambio de información como puede ser la dirección IP de los servidores. Sobre la ruta formada por la red primaria, un paquete (Ping) es intercambiado entre los servidores, el cual también contiene información que sirve para diferenciarlo de otros iniciados por otras aplicaciones de software. La falta de respuesta de un servidor a cualquiera de estos chequeos periódicos (Con reintentos) puede finalmente causar que el servidor alertado invoque los procedimientos establecidos de recuperación (Failover).

Durante el resto de este trabajo, usaremos el termino ping indistintamente de la ruta que estemos referenciando. Es decir, relacionaremos de igual forma el intercambio de un paquete comúnmente usado en terminología de red con el intercambio de un ping de SCSI. De esta forma cada miembro intercambia un ping de red y un ping de SCSI cada tres segundos. En caso de existir más de un bus SCSI compartido , el ping de SCSI será direccionado a través de cada uno de ellos. Esto hasta que a través de uno de ellos se obtenga la respuesta o todos los buses fallen.

Se debe notar que la falla un bus SCSI no es una razón necesaria para invocar el procedimiento de failover. Considerando las dos funciones principales de un bus SCSI, llevar I/O de y hacia los dispositivos e intercambiar información entre los servidores; si otro bus esta disponible y funcionando, la información sobre el estado del servidor puede ser intercambiada. Si no hay dispositivos de almacenamiento activos sobre el bus en falla, entonces no hay una razón determinada que obligue a disparar los mecanismos de alerta y/o recuperación de los servicios asociados en ese momento.

Las condiciones suficientes y necesarias para iniciar una respuesta del cluster en el caso de una falla de tipo SCSI son:

- La falla de un dispositivo de I/O
  No es razón suficiente para activar los mecanismos de recuperación el que uno de los servidores falle al responder a los pings enviados sobre todos los buses SCSI del mismo. En caso de falla de un dispositivo el servicio puede ser relocalizado o dado de baja, todo depende de si por lo menos otro de los miembros es capaz de accesarlo o no y solo al determinar lo anterior el cluster sabra cual de las dos opciones aplicar.

Cuando se presenta una falla de red, el Host Status Monitor diagnostica y determina si la falla es local (Por ejemplo, la falla de la interfaz de red del servidor) o si la red se ha particionado. En el primer caso, el cluster iniciará el mecanismo de failover hacia uno de los servidores miembro cuya interfaz pueda seguir operando. En el segundo caso, no hace sentido realizar un failover puesto que los demás miembros (Ubicados en la misma red) están siendo afectados también por el particionamiento de red. Cuando el particionamiento de red es seguido de una falla de tipo SCSI, los servicios afectados serán dados de baja pero no relocalizados; los servicios no podrán ser dados de alta mientras se

encuentre presente un particionamiento de red, el mecanismo de monitoreo del cluster continuará enviando pings y manteniendo este estado hasta que la red haya sido reparada y la operación pueda continuar de manera normal. Resumiendo, la condición de falla en la red que puede disparar el mecanismo de failover es:

- No hay respuesta de un servidor al ping realizado debido a que su interfaz de red ha fallado.

Existe una tercera condición bien definida que dispara el mecanismo de recuperación:

- La caída de un servidor, diagnosticada por la falta de respuesta tanto a los pings realizados por la red como a los realizados por los buses SCSI

Modo de operación

El cluster monitorea cinco tipos de falla:

- *Fallas en la ruta de acceso* SCSI, cuando el bus SCSI se encuentra inoperable
- *Fallas de dispositivo*, cuando falla un disco SCSI al responder alguna petición de I/O
- *Caída de un servidor*, cuando un nodo no responde a cualquier petición de respuesta
- *Fallas en la interfaz de red*, cuando una conexión de red falla
- *Partición de red*: fallas de la red, en base a la conexión de ésta con los servidores.

Vamos a suponer que la conexión del adaptador del bus SCSI de un servidor A se desconecta. Entonces, el siguiente ping SCSI que hay entre el servidor A y el servidor B fallará. Si la conexión de la red esta funcionando, entonces el ping de red se completará con éxito. Para esta situación, se genera un mensaje descriptivo de la situación como este: "SCSI ping fails but network ping succeeds", el cual será escrito en la bitácora de registro de eventos, indicando una *falla de ruta (path failure)*. El cluster no realizará ninguna acción hasta que el servidor A intente realizar operaciones de I/O en un disco sobre el bus afectado (Si el servidor B intentara realizar alguna operación de I/O en un disco ubicado en el mismo bus, ésta deberá tener éxito porque la conexión de este servidor no está afectada. Esta es la razón del cuidado que se debe tener de hacer uso de terminadores externos en el bus SCSI compartido: El servidor A está desconectado, pero el terminador externo está aún conectado al cable, de esta manera el bus SCSI puede continuar operando). Ahora vamos a suponer que el servidor A intenta realizar operaciones de I/O en un disco asignado a un servicio o aplicación de cluster, si el disco se encuentra en el bus desconectado, la operación debe fallar. El subsistema SCSI CAM agota eventualmente su lógica de comprobación y reporta un error, el controlador de discos del servidor A notifica al Manejador de Disponibilidad la falla del dispositivo, y éste a su vez la notifica al Agente (Cabe mencionar que no todos los errores de I/O serán considerados como fallas por el software de cluster. Por ejemplo, intentar leer el bloque N+1 de un disco de N bloques no es una falla del dispositivo, por lo que el cluster no reaccionará a esta situación).

El Agente entonces checa la descripción del servicio en los archivos de configuración del cluster, si el disco afectado está espejeado usando LSM, el mismo Agente ejecuta scripts para verificar si la copia de respaldo esta disponible y funcionando, de ser así, el cluster no ejecuta acción alguna (solo se registra un mensaje de alerta). De otra forma, el Agente detiene al servicio afectado (usando los scripts de baja) y notifica al Director, el cual asigna el servicio al primer servidor elegible que pueda tener acceso a todos los discos

que sean requeridos por dicho servicio. Los servidores elegibles son seleccionados de acuerdo a criterios especificados por el administrador del cluster en los archivos de configuración del mismo. El Director escoge cada servidor elegible, y ordena al Agente en ese servidor que inicie el servicio si puede alcanzar a todos los discos. Si ningún servidor elegible puede acceder a todos ellos, el Director repite esta operación con los criterios de disponibilidad de disco menos rígidos. Si este segundo intento falla, el servicio es marcado como"sin asignar" (unassigned) y un mensaje de alerta es registrado en la bitácora.

Así como se detectan las fallas en la ruta del SCSI, los dos tipos de falla de red son detectados por el Host Status Monitor (HSM), el cual notifica al Agente y al Director en caso de que éste se encuentre corriendo en el servidor afectado, se usa el mismo esquema de alerta para una falla de interfaz de red o para un particionamiento de red. Las fallas de particionamiento son las más simples de manejar: El proceso Director sencillamente termina haciendo que el cluster sea incapaz de arrancar el servicio. Los servicios ejecutándose en ese momento se quedan en ejecución, pero no se hace un failover a otro servidor. Ya que el failover sería infructuoso, puesto que todos los miembros se ven afectados por esta falla. Los servicios no se detienen porque hay una posibilidad de que los clientes sigan teniendo acceso al servidor. Un nuevo director no puede ser iniciado hasta que se halla reparado la falla de partición de red.

Sin embargo, cuando una interfaz de red ha fallado si tiene sentido el hacer un failover de los servicios[5]. El HSM del servidor afectado notifica a su Agente (y al Director si se está ejecutando en ese servidor), el cual detiene todos sus servicios, y avisa al HSM cuando lo ha hecho. El HSM notifica a las instancias de HSM de los otros miembros que el servidor ha sufrido una falla de interfaz y ha detenido sus servicios a través de un paquete de datos enviado por medio de

un ping SCSI, así los otros HSM notifican a su respectivo Agente. Mientras tanto, si el Director se encontraba en el miembro afectado, termina su ejecución en el momento que el HSM le notifica la falla de la interfaz. Ahora los Agentes de los miembros restantes iniciarán otro Director en algún otro servidor miembro (En el servidor activo con la dirección IP más alta). El Director pregunta a los HSM por el estado de todos los servidores en el cluster (Hace esto en respuesta a un mensaje de alerta de un HSM, o al arranque, según sea el caso). Y como hizo para las fallas de ruta del SCSI, el Director busca en los archivos de configuración otros servidores donde pueda reiniciar el servicio, e indica a los respectivos Agentes que realicen dicha operación.

La respuesta a una "caída de servidor" (Host down), empieza cuando cada HSM que se encuentra activo notifica a su Agente que un servidor está fuera de servicio. Alguno de los HSM también notifica al Director, en caso de que el Director no pueda ser notificado, debido a que estaba siendo ejecutado en el servidor caído, se creará un nuevo Director el cual inmediatamente se conectará al HSM local y a todos los Agentes activos. En cualquier caso, el director obtiene el estado de todos los miembros del cluster a través HSM local. Revisa los archivos de configuración para determinar que servicios se estaban ejecutando en el servidor inoperante, para posteriormente reiniciarlos en algún otro miembro como se explicó anteriormente. Al desarrollar los scripts de acción, es importante recordar que en este caso, el servidor que falló pudo no tener oportunidad para ejecutar los scripts de baja de los servicios.

[5] Considerando que cada miembro del cluster solo tiene una interfaz de red. En caso de tener NetRAIN configurado, se hace un fallover de la interfaz de red.

4 COMPONENTES DE UN CLUSTER

4.1

INTRODUCCIÓN

Este capítulo, describe algunos componentes y estrategias que pueden ser efectivos para mejorar o incrementar la disponibilidad de un cluster. Estas estrategias son generalmente consideradas durante la planeación de un sistema e implementadas durante la puesta en operación del mismo. Pueden ir desde soluciones de uso común hasta soluciones muy costosas con capacidad de tolerancia a fallas.

Implementar una o varias estrategias bien planeadas ayuda a incrementar la disponibilidad de un cluster y al mismo tiempo permite reducir los costos de soporte y recuperación de fallas.

El énfasis de todas estas estrategias es el de utilizar hardware estándar de uso común y cuando es posible el de alcanzar el mejor precio/desempeño posible.

4.2

RED EMPRESARIAL

Uno de lo elementos importantes de un cluster es el uso de una RED, se le denomina Red Empresarial debido a que ésta, puede formar parte de una LAN e incluso a una red WAN. Cabe mencionar que en este apartado no se describirá la implementación de una red, únicamente mencionaremos el uso y el control que se debe tener como parte de un sistema cluster. La Figura 4.2-1 muestra como una red puede formar parte del cluster.

Pero, ¿que tiene que ver el uso de la red para tener Alta disponibilidad ?. Al incorporar a la red un cluster se debe verificar que se cumplan con las siguientes características:

- La seguridad en la red
- Control de acceso a la red
- Integración de los sistemas a un ambiente de red

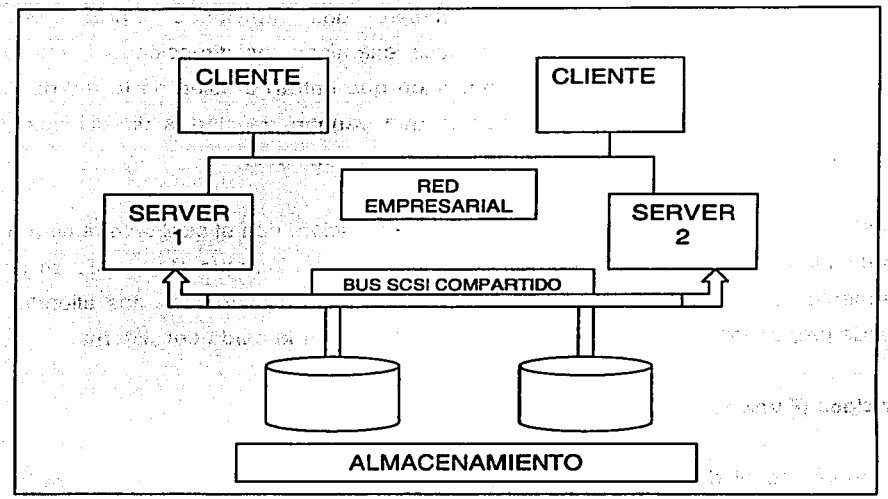


Figura 4.2-1 Componentes de un Cluster

La seguridad en la Red

Existen elementos físicos que son parte de la red que nos pueden auxiliar en mantener alta disponibilidad en el cluster.

Los Firewalls

Los firewalls son, en esencia, servidores que se anteponen entre Internet y la red local de la empresa. Todo usuario que pretenda acceder a la red privada deberá pasar por este servidor, que generalmente utilizará un núcleo seguro adaptado para llevar a cabo una única tarea, la específica de firewalls.

Estos servidores especializados cumplen dos funciones primordiales: primeramente filtran los paquetes, es decir examinan las direcciones fuente y destino de todos los paquetes de información que entran o salen de la red de la empresa. Y además , bloquea paquetes que vengan de ciertas direcciones y prevenir a los que intenten salir con destinos no autorizados.

Existe otro tipo de firewalls que es a nivel de aplicación, con el cual se examina el contenido de la información, así como sus direcciones. Aunque el firewalls es un elemento de filtrado lento (dependiendo del tráfico o de cuantos nodos atiende), puede proporcionar una mayor seguridad y poder evitar la caída del sistema.

Bridges (Puente)

El bridge es el dispositivo más simple para realizar la conexión de las redes locales. Este dispositivo se diseñó para unir redes LAN que usen el mismo protocolo físico y de acceso al medio. El Bridges proporciona un camino a la estación de una red para que difunda mensajes a las estaciones de otras redes.

El uso de Bridge esta determinado por las siguientes razones:

- Para ampliar la extensión de la red o el número de nodos que la constituyen

- Para reducir el cuello de botella del tráfico causado por un número excesivo de nodos unidos

Al cumplir con sus funciones el Bridge, se puede obtener una red altamente disponible debido a que reduce las fallas tanto de trafico de datos como la conexión de los elementos de hardware con la red misma.

Switch

Es un dispositivo de propósito especial diseñado para resolver problemas de rendimiento en la red, debido a anchos de banda pequeños y embotellamientos, lo que puede hacer él, es agregar mayor ancho de banda, acelerar la salida de paquetes, reducir tiempo de espera y bajar el costo por puerto.

Control de acceso a la red

Para un efectivo control de acceso a la red, se disponen de técnicas para lograr dicho objetivo:

- Un sistema con contraseña desde el teclado(Password)
- Un sistema con contraseña en el teclado apoyado con un software que le ponga restricciones (Seguridad de la red por medio de Software)
- Un sistema basado en "tokens", siendo un token un objeto físico único, que almacena la información necesaria para realizar el proceso de identificación mediante un protocolo determinado. De éstos, se pueden destacar:
- Las tarjetas de banda magnética en los que se almacena la identificación del usuario,
- Las tarjetas inteligentes que contiene un chip con la información del usuario y se accede mediante un PIN (número privado), para realizar la verificación que se trata de la persona que gestiona el servicio

Existe la posibilidad de una gestión con Single Sign-On[1] en la cual el usuario sólo tiene que autenticar una única vez para acceder a todos los recursos sobre los que tiene derecho. Esto facilita enormemente las tareas de administración, aunque puede llegar a suponer una merma importante en la garantía de dicha seguridad, debido a que con una sola contraseña mal gestionada y por lo tanto accesible, se puede desmembrar con gran facilidad la arquitectura de seguridad.
Una propuesta para un buen sistema de seguridad sería una combinación de métodos y técnicas de las siguientes características:

- Un sistema basado en token
- Contraseña dinámica
- Técnicas de gestión mediante Single Sign-On

Integración de los sistemas a un ambiente de Red

Una vez logradas las conexiones físicas de las red así como la configuración de la misma, es el momento de integrar los sistemas de la que se tendrán en el Cluster. Para realizar la migración de los sistemas es necesario realizar un estudio de las necesidades que se tienen para sugerir la adquisición y/o adecuación y/o implantación y/o desarrollo de sistemas que optimicen la operación, que generen la información necesaria de manera oportuna, se adapten a las políticas y procedimientos de la compañía y por supuesto, simplifiquen los procesos que permitan tener datos constantes y sobre todo un tráfico de datos dentro de la red sin tener fallas en la misma.

Adicionalmente a la instalación de los componentes de hardware y del control de acceso a la red , se debe contar con elementos como:

[1] Single Sign-on es la definición que se da para especificar que los usuarios no tienen la necesidad de recordar todas sus claves para poder acceder a todos los recursos de la red.

Software para detectar intrusos: Existen soluciones que se ocupan de generar reportes de registro sobre el tráfico de la red. También monitorean el sistema para detectar cualquier **paquete de datos sospechoso**, y ante una eventualidad están en condiciones de desbaratar los ataques potenciales o bloquear la conexión en caso de localizar alguna actividad anormal en la red.

Filtro de contenidos: La misión de los filtros de contenido es **agilizar la productividad**. En una gran empresa la comunicación interna se realiza a través del correo electrónico. Muchos de los e-mail, viajan con archivos de audio o video que, por su peso, forman un cuello de botella cuando los datos circulan por la red. Los filtros se encargan de detener todos aquellos contenidos superficiales: hoax (amenazas falsas de virus), spam (cadenas de e-mail, promociones), programas de back office (que permiten controlar una computadora a distancia) y cualquier intento de **violación o sabotaje interno**.

A estos programas se les utiliza también para restringir el acceso a Internet en determinadas horas del día; **evitar que se bajen programas**, que se visiten sitios con contenidos pornográficos, racistas o violentos, o que se haga chat desde el lugar de trabajo que hagan el trafico de la red más lento que podría ocasionar que la alta disponibilidad no existiera.

REDUNDANCIA DE RED

Para tener una red de alta disponibilidad es necesario contemplar el uso de redundancia, sí algún elemento falla, la red deberá por sí misma seguir operando. Un sistema tolerante a fallas debe estar diseñado en la red, de tal manera que, si un servidor falla, un segundo servidor de respaldo entrará a operar inmediatamente. La redundancia también se aplica para los enlaces externos que aseguran que la red siga funcionando en caso de que un equipo de comunicaciones falle o el medio de transmisión en cuestión. Es común que compañías tengan enlaces redundantes, si el enlace terrestre falla (por ejemplo,

una línea privada), entra en operación el enlace vía satélite o vía microondas. Al instalar una red redundante es necesario considerar que el costo de implementación es grande pero el beneficio que se obtiene es de que el usuario final no se percatará de fallas en el uso del sistema.

Para la implementación de la redundancia en la red es necesario primero detectar los puntos de fallo, estos pueden ser desde una tarjeta hasta un elemento como los switch. A continuación mencionaremos algunas configuraciones de red en donde se incluyen los elementos de hardware en los puntos críticos de falla.

Múltiples redes de protección

Esta configuración de firewall proporciona mayor redundancia y mayor privacidad en las conexiones al exterior. Realmente, lo que se está haciendo es emplear varios firewalls en el mismo sistema. Si el sistema dispone de conexiones a varias redes externas y una de ellas necesita de confidencialidad, puede dedicársele una red periférica y un router externo. A pesar de todo, el hecho de tener que mantener varios routers interiores tiene gran dificultad, y puede presentar potenciales problemas de seguridad. Por ejemplo, dos routers que dan acceso a la misma red externa deben mantener la misma política de seguridad. La Figura 4.2-3 muestra como se puede realizar esta conexión.

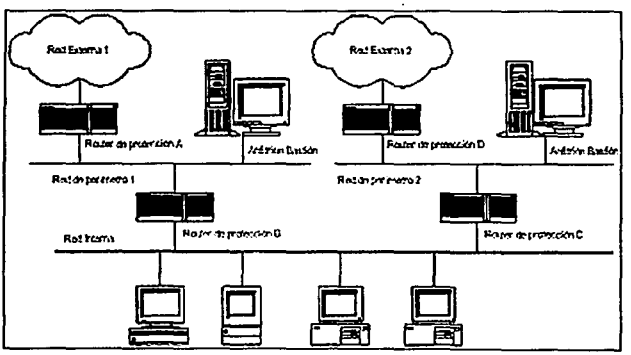


Figura 4.2-3 Múltiples redes de protección

Arquitectura de subred de protección

Este tipo de arquitectura proporciona una capa adicional de seguridad a la arquitectura de anfitrión de protección. Esta capa adicional se consigue añadiendo una red de perímetro[2] que aísla aún más la red interna a proteger de la externa. En la red de perímetro debe situarse el host de entrada puesto que se trata de las máquinas más vulnerables de toda la red y las que cuentan con mayor probabilidad de ser atacadas.

De este modo, se protege mejor al anfitrión y, en consecuencia, a las máquinas de la red interna frente a la arquitectura de anfitrión de conexión.

La arquitectura de subred de protección clásica se basa en dos routers de protección (con sus correspondientes filtros de paquetes) conectados a la red de perímetro. La Figura 4.2-4 muestra como se encuentra conectado a la red interna y a la de perímetro mientras que el otro se conecta a la red externa y a la de perímetro. De este modo, un atacante tendría que violar la seguridad de los dos routers antes de poder acceder a la red interna.

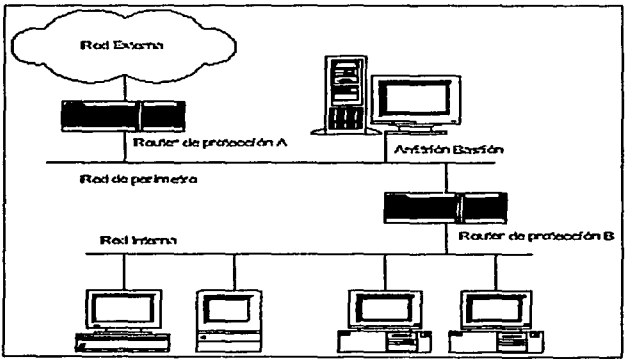


Figura 4.2-4. Arquitectura de subred de protección

[2] Se trata de una red adicional entre una red protegida y una red externa a fin de proporcionar una seguridad adicional.

TESIS CON
FALLA DE ORIGEN

En algunas instalaciones puede crearse una serie de capas de redes de perímetro entre la red externa y la interna. Los servicios menos confiables y más vulnerables se colocan en las redes de perímetro exteriores, más lejos de la red interior. De este modo, al atacante que logre acceder a una máquina de perímetro exterior, le costará trabajo atacar con éxito las máquinas internas debido a las capas adicionales de seguridad entre perímetro exterior y la red interna, siempre y cuando, las capas de filtrado en los routers sean las adecuadas.

Conexión de múltiples servidores

Solución válida por varias razones:

- Rendimiento : Mejora las prestaciones del sistema si se dedica un servidor a servicios que suponen una carga para el sistema, o bien ofreciendo los mismos servicios pero logrando una distribución de la carga entre los múltiples servidores
- Redundancia : Si uno de los servidores falla, los servicios podrán ser soportados por otro. No todos los servicios permiten esta posibilidad
- Separación de servicios : Por razones de seguridad o de rendimiento puede ser conveniente separar servicios en diferentes servidores. Por ejemplo un servidor podría albergar el servicio HTTP (Hipertext Transference Protocol) y otro el de FTP (File Transference Protocol) anónimo para evitar que uno pudiese ser empleado para comprometer al otro

La Figura 4.2-5 muestra como se podría realizar la conexión de los servidores en una misma red proporcionando a dicha red un servicio interrumpido.

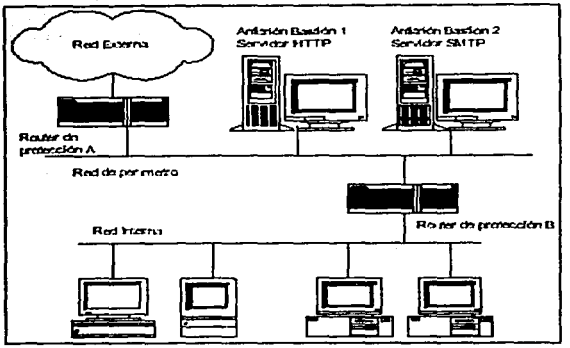


Figura 4.2-5 Conexión de múltiples servidores

Pero en este sentido no solo se pude tener redundancia en los componentes tales como firewalls, bridge, servidores, etc. Para tener una menor tiempo o una falla casi nula, existen componentes que ayudan a no tener fallas. La Figura 4.2-6 muestra como podemos tener la conexión redundantes en la RED.

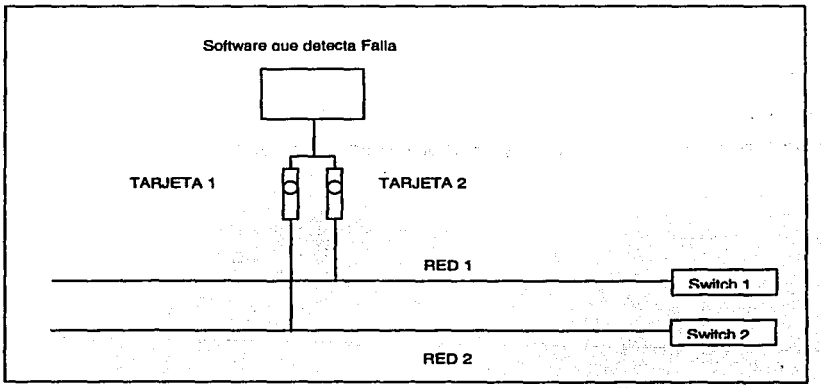


Figura 4.2-6 Conexión de componentes redundantes

La Figura 4.2-6 muestra la conexión de algunos componentes redundantes, inicialmente se cuenta con un equipo al cual se le asigna una sola dirección IP, este servidor cuenta con dos tarjetas de red, inicialmente la tarjeta 1 es la que se encuentra activa y que nos proporciona la salida a la red exterior. Cuando se produce una falla de la tarjeta 1, el software instalado en el servidor detecta la falla y activa a la tarjeta 2 asignándole la misma dirección IP del equipo. Como se muestra en la figura, las tarjetas se encuentran conectadas independientemente una a cada red privada (Red 1 o red 2).

Otro punto de falla se puede encontrar en el medio físico, si la red 1 deja de funcionar el mismo software que se encuentra en el servidor, detecta la falla y realiza el cambio a la red 2 por medio de la tarjeta 2. Con ese tipo de conexiones los servicios siempre se encuentran activos dando tiempo al administrador de la red para poder reparar la falla. Este tipo de solución es usada en el mercado y se le conoce como arreglo redundante de adaptadores de red o NetRAIN (Redundant Array of Independent Network Adapters).

Específicamente NetRAIN es una configuración de múltiples interfaces en un mismo segmento de red, la forma en que trabaja NetRAIN es muy simple uno de los adaptadores siempre está activo mientras que el otro permanece ocioso. Si el adaptador activo presenta una falla, el otro entra en línea. El tiempo de failover dependerá de la configuración y funcionamiento de nuestra red.

4.3

RED PRIVADA

Los miembros del cluster deben contar con una forma de comunicación eficiente entre ellos, principalmente para saber el estado de funcionamiento de cada uno de los miembros, el propósito de ésto, es determinar cuando se presenta una falla en alguno de los nodos y poder tomar acciones para mantener disponible la aplicación.

4.3.1 Cluster Interconnect

Una forma de interconectar los nodos de un cluster es por medio de interfaces de red, donde generalmente se pone una conexión redundante con comunicación serial. Esta es una de las razones por la que se llama red privada a este tipo de conexión, ya que prácticamente funciona como una subred en la que sólo figuran los miembros del cluster. Esta forma de conexión es rápida, dependiendo de la interfaz de red que se use.

4.3.2 Canal de Memoria (Memory Channel)

El Canal de Memoria es un sistema de interconexión que se utiliza para el envío y recepción de mensajes entre los nodos de un cluster. Esta conexión, que se basa en el bus PCI, proporcionando una red de alta velocidad, que brinda a las aplicaciones un espacio de direcciones de memoria común para todo el cluster. Las aplicaciones instaladas, pueden leer y escribir páginas de 8 KB a este espacio de direcciones como a cualquier memoria normal. Además cuenta con un sistema de verificación de errores, lo cual elimina la necesidad de tener software para este propósito.

El Canal de Memoria proporciona las siguientes características:

- Comunicación de alto desempeño entre el cluster
- Rápida detección de la falla de un nodo
- Interconexiones redundantes del cluster
- Comunicación directa de aplicación a aplicación
- Poco overhead en el sistema
- Sin protocolos de comunicación
- No necesita programación especializada

Funciones del Canal de Memoria

El Canal de Memoria actúa como una compuerta de escritura a la memoria de otro sistema (nodo). La petición de escritura pasa al adaptador del Canal de Memoria, éste coloca el dato en esta interconexión del cluster para una dirección en particular. El nodo que reconoce la dirección de memoria toma el dato y lo transfiere del Canal de Memoria hacia su propia memoria.

La implementación del canal de memoria se realiza por medio de adaptadores los cuales se instalan en ranuras de expansión PCI. Cada miembro del cluster debe tener un adaptador para el Canal de Memoria. Cuando hay solamente dos nodos

en el clusters, estos se pueden conectar directamente con un cable de enlace (Figura 4.3.2-1).

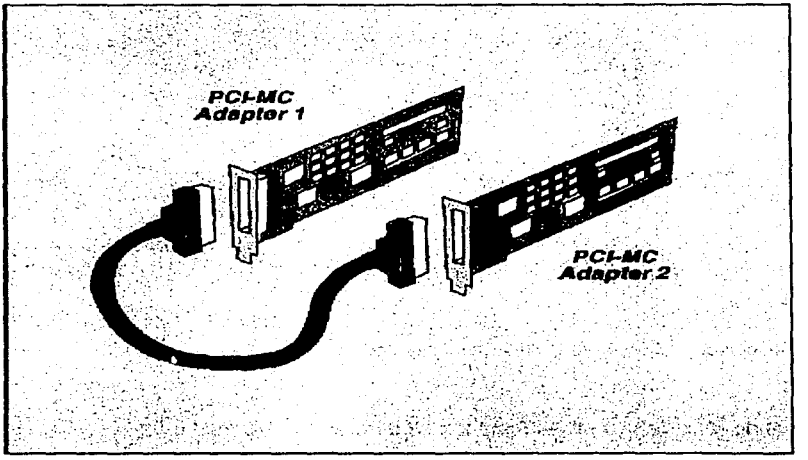


Figura 4.3.2-1 Conexión de dos adaptadores de Canal de Memoria.

Cuando hay más de dos nodos en el cluster, se pueden conectar por medio de un concentrador (HUB) de Canal de Memoria tal y como lo muestra la Figura 4.3.2-2

La interconexión de Canal de Memoria proporciona una ancho de banda de 100+ MB/s con un ciclo de 30 ns y una latencia en los mensajes de 4 μs (en comparación con la latencia en los mensajes que presentan las redes Ethernet, Fast Ethernet , y FDDI usando protocolo TCP/IP que es de 400 μs).

Se puede configurar un Canal de Memoria dual donde, el segundo canal permanece inactivo hasta que se detecta un error en el primer canal. En este caso, el cluster automáticamente detecta la falla y realiza el failover al segundo canal para continuar con su operación normal.

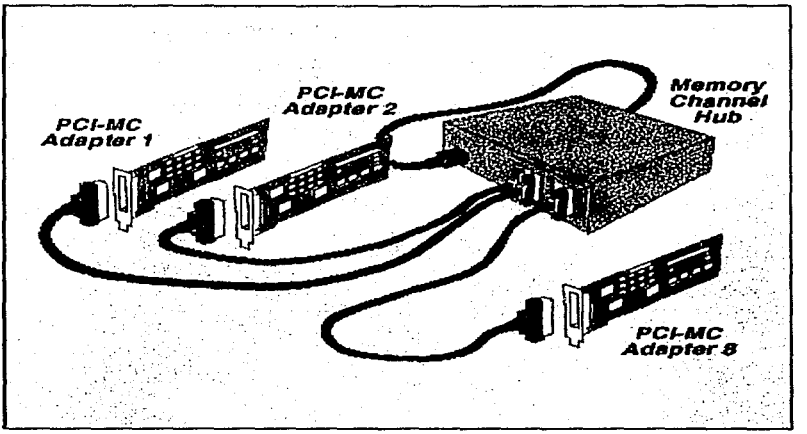


Figura 4.3.2-2 Concentrador de Canal de Memoria.

El Canal de Memoria también tiene las siguientes funciones:

- Transporta tráfico de comunicación entre los nodos del cluster.
- Transporta tráfico IP en el cluster.
- Soporta aplicaciones de cluster tales como:
  - DLM (Distributed Lock Manager), que sincroniza el acceso.
  - DRD (Distributed Raw Disk), hace que los discos estén disponibles para los nodos del cluster.
  - Monitoreo de la configuración del cluster.
  - Registro de errores dentro del cluster.

Failover automático

La interconexión de Canal de Memoria automáticamente redirige la comunicación cuando falla un miembro del sistema.

- La información (datos) son servidos por el miembro del cluster al que le pertenecen los discos.
- Cuando un miembro del sistema falla, otro de los miembros retoma el servicio. El miembro fallido se elimina del cluster, y a los miembros restantes se les informa que un miembro ha sido removido.
- Cuando hay un Canal de Memoria dual, el segundo canal permanece inactivo hasta que se detecta un error en el primer canal. En este caso el cluster automáticamente cambia al segundo canal para continuar con su operación normal. El Canal de Memoria dual comparten una misma dirección de red.

La Figura 4.3.2-3 muestra como se redirige una petición hecha por el nodo C. Si el nodo A falla, el nodo B recibe la notificación y acepta la petición de acceso a los datos del disco compartido.

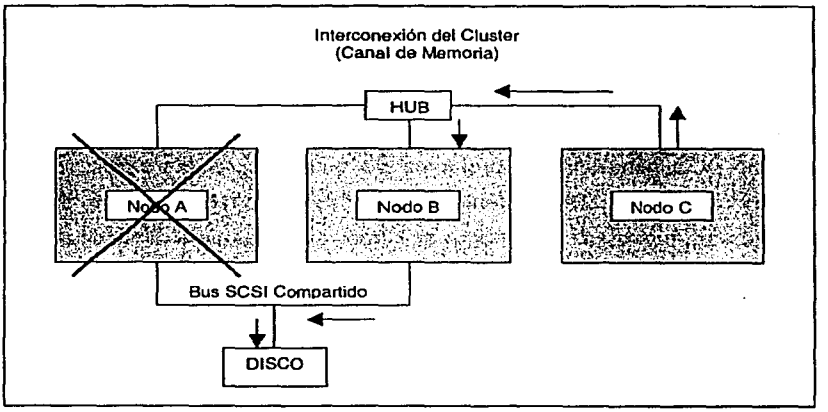


Figura 4.3.2-3 Redirección de información entre los miembros activos de un cluster

4.4

BUS DE ALMACENAMIENTO COMPARTIDO

Anteriormente se tenía la costumbre de utilizar adaptadores SCSI conectados directamente a un servidor. El almacenamiento compartido se realizaba cuando se conectaban de cada uno de los elementos de almacenamiento por un mismo elemento de BUS físico compartido. Este tipo de conexión tenía sus limitaciones; ésto era debido a que los elementos de almacenamiento se encontraban unidos por un mismo BUS generando el problema de que no se podía tener múltiples servidores conectados simultáneamente. Este problema es relativamente simple, y solo se limitaba al número de servidores que podían compartir el almacenamiento y el BUS de almacenamiento comenzaba a tener problemas de lentitud.

Actualmente los dispositivos de almacenamiento pueden estar conectados a la red privada directamente. Con esta idea, el almacenamiento dejó de estar atado a un solo BUS, y se implementa un ambiente de redes de área de almacenamiento (SAN´s[3]) donde los beneficios que se obtienen son muy grandes; éstos beneficios son:

[3] Storage Area Network

- Uso de un BUS privado dedicado a compartir datos
- Uso de una red privada
- Transferencia de datos

El BUS de almacenamiento compartido según se muestra en la Figura 4.4.-1, es el encargado de realizar la conexión física de los elementos de almacenamiento utilizando un canal de fibra (Fibre Channel) el cual permite transferir datos a una gran velocidad, además de que permite la conexión de sistemas de almacenamiento punto a punto o combinaciones de topologías.

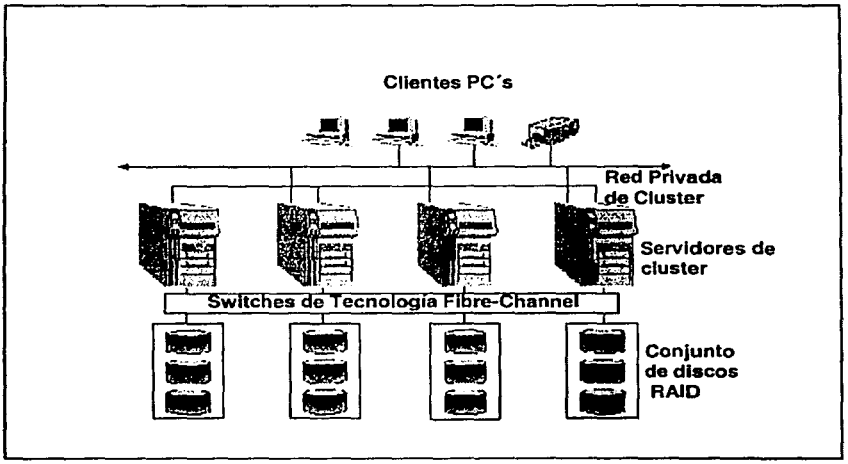


Figura 4.4-1 Cluster de cuatro nodos, ubicación del BUS de almacenamiento compartido

Cuando un elemento de almacenamiento tiene fallas, el software del cluster busca mediante el BUS los elementos de almacenamiento que se encuentran operando en el cluster y realiza el failover, para poder seguir realizando las operaciones del sistema.

Para una operación optima del BUS de almacenamiento compartido se debe contar con los siguientes elementos de apoyo :

- Drivers Básicos : Para poder realizar la conexión de los adaptadores del BUS con el ambiente de redes de almacenamiento.
- Soporte para el canal de fibra- switches: Que permita la conexión continua de uno o mas nodos de almacenamiento (Véase Figura 4.4-1).
- Cascada: Tener la habilidad de que los sistemas operativos puedan acceder a múltiples niveles del BUS compartido con el uso de los switches
- Zonas: Dividiendo el BUS compartido por zonas se puede estar protegido de una falla en el sistema. Si se realiza una división por zonas, cada una de ellas tendrá su propia seguridad de almacenamiento, es decir se crea una zona de almacenamiento para finanzas, otra para crédito, etc.

4.5

ALMACENAMIENTO DE DATOS

Diseñar una estrategia de almacenamiento de datos es una de las tareas más importantes en la implementación de un sistema de misión crítica. Esto es verdad por varias razones:

- Preservar la disponibilidad y la integridad de los datos es un requerimiento crítico
- Proveer un flujo óptimo de I/O entre el subsistema de almacenamiento y el subsistema de procesamiento ayuda a maximizar el desempeño de la aplicación

Realizar las tareas de mantenimiento de manera eficiente permite disminuir la carga de trabajo por administración del subsistema de almacenamiento.

Todos sabemos que los sistemas de almacenamiento y en su forma más sencilla: los discos duros están expuestos a fallas debido a variaciones de temperatura, ruptura de las cabezas de grabación/lectura con la correspondiente destrucción de los discos, y cambios en las condiciones de suministro de energía. Asimismo, fallas a nivel del sistema operativo, virus y/o la gran cantidad de tráfico en la interfaz de entrada y salida afectan su confiabilidad.

Cualquiera que sea el caso de fallase debe contar con un sistema de almacenamiento confiable, por lo cual hablaremos un poco acerca de las diferentes tecnologías para incrementar la disponibilidad de una solución de almacenamiento de datos.

4.5.1 Tecnología RAID

La tecnología RAID (Redundant Array of Independent Disks) permite que los datos sean almacenados de forma distribuida sobre series de discos, en lugar de seguir la forma secuencial y tradicional sobre discos individuales. Los datos de un archivo son divididos en segmentos (grupos de bloques) los cuales pueden ser escritos a través de múltiples discos. Al usar más de un disco, la tecnología RAID puede proveer tres beneficios principales:

- Incrementa la disponibilidad de los datos a través del uso de configuraciones redundantes
- Incrementa el performance de I/O y los índices de transferencia de datos comparados con los de discos individuales
- Incrementa la escalabilidad del subsistema de almacenamiento y como consecuencia la del propio sistema

La alta disponibilidad se logra creando duplicados de los datos originales (espejeo) o usando algoritmos de paridad los cuales permiten re-crear los datos contenidos

en cualquier disco dañado a partir de los datos contenidos en los discos que se encuentran en buen estado y que continúan operando.

El alto desempeño se puede alcanzar dividiendo la información en segmentos y distribuyendo los mismos en varios discos (distribución). Esto puede llegar a permitir la ejecución de múltiples operaciones de I/O en un archivo simple al mismo tiempo.

Combinando las técnicas de disponibilidad y desempeño mencionadas anteriormente, es posible obtener lo que se conoce como "niveles RAID", los cuales de acuerdo con sus características particulares pueden cubrir diferentes necesidades.
Al día de hoy, la industria a definido ocho niveles estándar de RAID (nivel 0 al nivel 7) cuyas características básicas de funcionamiento son mostradas en la Tabla 4.5.1-1.

Nivel RAID	Almacenamiento de datos	Redundancia
0	Distribuido	Ninguna
1	Normal	Espejeo
0 + 1	Distribuido	Espejeo
3	Distribuido	XOR[4] en un solo disco
5	Distribuido	XOR Distribuida
6	Distribuido	El resultado de dos ecuaciones en dos discos
7		Implementación dependiente del proveedor

Tabla 4.5.1-1 Sumario de las características de los niveles RAID

[4] La operación binaria XOR

La tecnología RAID puede ser implementada en los subsistemas de almacenamiento a través del uso de hardware específico o a través del uso de software. En una implementación de hardware, los algoritmos se ejecutan en el controlador de discos, mientras que en una implementación de software los algoritmos se ejecutan consumiendo recursos del servidor donde reside el sistema operativo.

La tecnología tiene tres atributos principales:

- Es un conjunto de discos físicos reales vistos y manipulados por el usuario como un dispositivo lógico simple
- Los datos de usuario son distribuidos a través de los discos pertenecientes a un conjunto de una manera definida
- Adiciona la capacidad de redundancia de discos en un sistema, ya que los datos pueden ser recuperados aun en caso de la falla de un disco

El RAID Advisory Board define un arreglo de discos como una colección de discos con algún software de administración del mismo. El software, que puede ejecutarse en el subsistema de discos o en el servidor controla la operación del arreglo y lo presenta como un disco virtual más.

Varios niveles de RAID soportan el reemplazo en caliente (hot swapping), que es una sustitución manual de un disco en falla mientras la operación normal continua siendo ejecutada.

Los tres requerimientos más importantes en el diseño de subsistemas de almacenamiento de datos generalmente son los siguientes:

- Bajo costo
- Alta disponibilidad
- Alto desempeño

La interrelación de estos tres factores se ilustra en el triángulo mostrado en la Figura 4.5.1-1. Para algunos usuarios un sistema tiene tres principales atributos: bajo costo, bajo costo, bajo costo. Estos usuarios ocupan la porción izquierda inferior del triángulo. Otros, pueden perder millones de dólares en su negocio si sus sistemas caen por solamente un pequeño lapso de tiempo; para ellos gastar dólares adicionales para proveer alta disponibilidad es una inversión bien realizada. Ellos ocupan la porción derecha inferior del triángulo. Para otros hacer el trabajo de manera rápida sobrepasa cualquier otra consideración. Ellos ocupan la porción superior del triángulo.

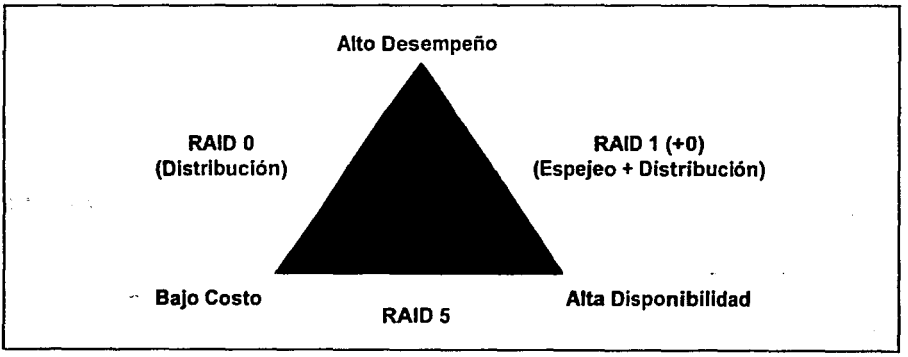


Figura 4.5.1-1 Interrelación entre Desempeño, Disponibilidad y Costo en la tecnología RAID

Diversas configuraciones RAID satisfacen diferentes combinaciones de estos requerimientos, y pueden ser visualizadas ocupando diferentes puntos del triángulo.

Debido a que la disponibilidad y el desempeño de los datos son ambos importantes en diferentes grados, el concepto RAID fue concebido para proporcionar diversas configuraciones o niveles que permiten obtener diversos rangos de fiabilidad y desempeño mismos que podrán ser evaluados para seleccionar el nivel más adecuado para cierto tipo de aplicaciones y ambientes de operación.

A continuación se describen brevemente los diversos niveles RAID existentes en el mercado.

Descripción de los niveles RAID

La tecnología RAID puede ser configurada de diversas formas. Estas configuraciones son llamadas niveles. Los niveles se diferencian por la forma en la cual:

- La información es distribuida
- La capacidad redundante es implementada

Los niveles del 0 al 7 han sido definidos por la industria. Los niveles 0,1,3 y 5 están comercialmente disponibles.

Nivel RAID 0

El nivel RAID 0 proporciona distribución de datos sin redundancia.

La Figura 4.5.1-2 muestra la operación de este nivel, donde el segmento A es físicamente almacenado en el primer disco, el segundo segmento, B, es almacenado en el segundo disco. De la misma forma el tercer segmento C, es colocado en el tercer disco y el cuarto segmento es colocado en el cuarto disco. El quinto segmento, el E, es entonces colocado sobre el primer disco siguiendo al segmento A, el sexto, el F, sobre el segundo disco siguiendo al segmento B, y así sucesivamente hasta que la capacidad del dispositivo lógico es agotada.

Para leer, la solicitud de un usuario es mapeada contra la localización física de la información. Los datos son entonces leídos de esa localidad. Para escribir, la localización es mapeada y la información escrita a la localización física. La carga de trabajo se distribuye a través de los discos y el tamaño del segmento puede ser configurado para proporcionar el mejor desempeño.

El tamaño del segmento es elegido para ser grande con respecto al tamaño de la solicitud del usuario. Con esa relación, cada uno de los discos del arreglo estará habilitado para ejecutar una petición diferente. Por el contrario, si la petición es grande con respecto al tamaño de segmento, cada disco dentro del arreglo ejecutará solo una porción de la petición a la vez.

Resumiendo, el nivel cero de RAID es el mejor en términos de costo ya que el usuario no pagará por capacidades extra de redundancia. Sin embargo, la integridad de los datos es pobre y el desempeño es bueno. La carga de trabajo se balancea a través de los discos, el tamaño del segmento es configurable para proveer el mejor desempeño posible y no se realiza trabajo extra para escribir redundancia.

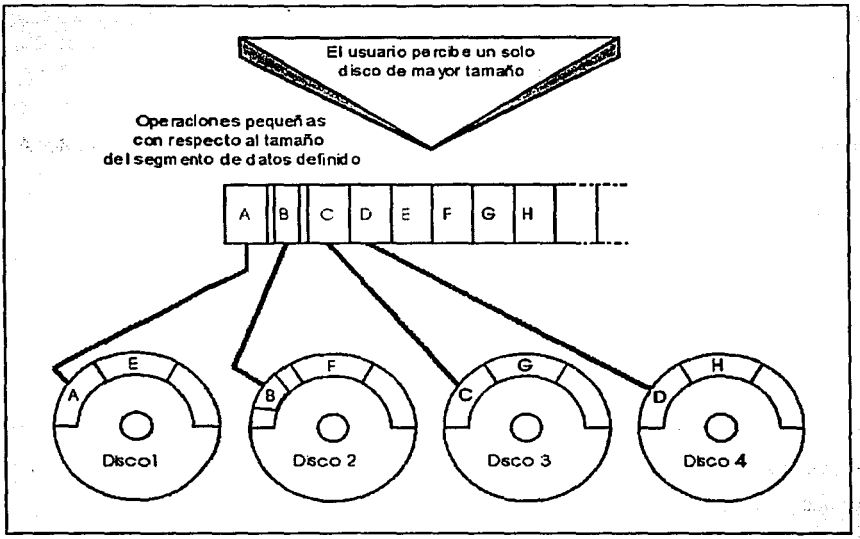


Figura 4.5.1-2 Nivel RAID 0.

Nivel RAID 1

El nivel uno de RAID provee la capacidad de redundancia.

Como muestra la Figura 4.5.1-3 los segmentos de datos son almacenados de forma tradicional. Esto es, los segmentos A, B, C se almacenan en orden secuencial en el primer disco y al mismo tiempo y de la misma forma en el segundo disco.

Al leer, la petición del usuario es mapeada contra la localización física de los datos, entonces la información puede ser leída de cualquiera de los dos discos donde reside pero solo uno ellos realiza el trabajo.

Al escribir, la localización es mapeada y los datos son escritos en ambos discos miembros del espejo. De tal manera que cada uno de los discos realiza las operaciones de escritura.

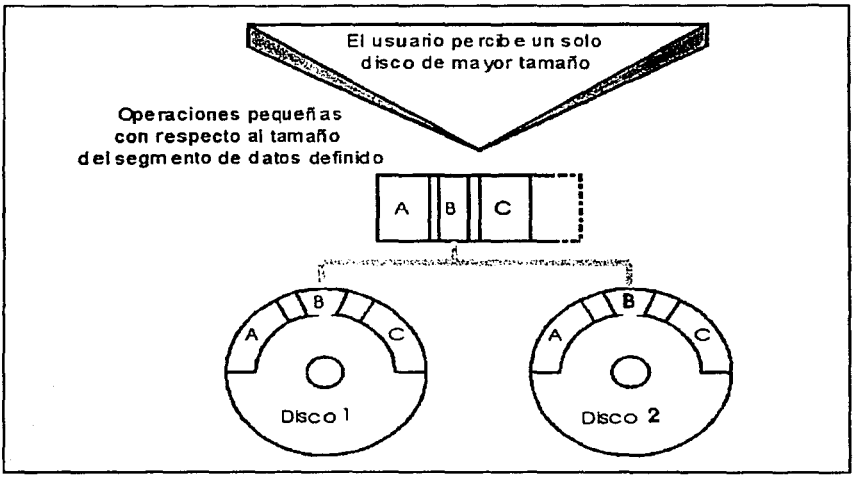


Figura 4.5.1-3 Nivel RAID 1.

En el nivel uno de RAID, la carga de trabajo no se balancea entre los discos, el tamaño del segmento es configurable para proveer el mejor desempeño y la integridad de los datos es excelente ya que existe total redundancia. El RAID uno es la solución más cara en términos de costo ya que requiere que cada disco tenga su correspondiente espejo.

Nivel RAID 0 + 1

El nivel RAID 0 + 1 proporciona capacidades de distribución y redundancia de datos ya que ofrece en una misma solución las bondades de los niveles cero y uno de RAID.

La Figura 4.5.1-4 muestra que la información es dividida en segmentos, el segmento de datos A es físicamente guardado en el primero y segundo discos, el segundo segmento, B, es guardado en el tercero y cuarto discos en la misma localidad física en la cual A fue almacenada. Los segmentos C y D son guardados siguiendo a los dos segmentos previos A y B respectivamente y así sucesivamente.

Para leer los datos, estos son mapeados contra la localidad física del disco. La información es entonces leída de cualquiera de los dos discos que la contienen, de tal manera que cada par de discos espejeados ejecutan solo la mitad del trabajo de lectura. Para escribir, la localidad física es ubicada y los datos entonces escritos en ambos miembros del par o espejo.

En este caso, cada disco del arreglo estará en posibilidad de ejecutar una operación de lectura mientras que solo se puede ejecutar una operación de escritura por cada par de discos. La carga de trabajo estará balanceada entre todos los discos pertenecientes al arreglo además de que el tamaño del segmento es configurable para poder obtener el mejor desempeño posible.

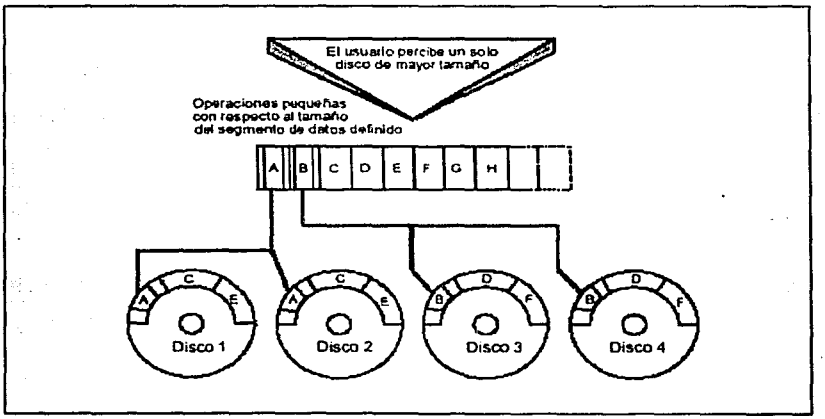


Figura 4.5.1-4 Nivel RAID 0 + 1.

Nivel RAID 3

El nivel RAID 3 proporciona capacidades de distribución de datos y redundancia.

La Figura 4.5.1-5 muestra que el segmento de datos A es almacenado en el primer disco. En el segmento B se almacena en el segundo disco, el segmento C en el tercero, el cuarto segmento D, es colocado siguiendo a A en el primer disco; el quinto segmento E se coloca en el segundo disco siguiendo al segmento B y así sucesivamente.

La redundancia es provista por el cuarto disco. El primer segmento de este disco contiene la operación XOR exclusiva de los primeros segmentos almacenados en los discos uno, dos y tres. El segundo segmento del cuarto disco contendrá la XOR de los segundos segmentos de datos de los primeros tres discos y así sucesivamente.
Para leer, la petición del usuario se mapea contra la localidad física del disco donde los datos residen. Los datos son entonces leídos simultáneamente de todos los discos. Para escribir, la localidad física se mapea y los datos son escritos simultáneamente a todos los discos. Mientras la operación de escritura está siendo ejecutada, la operación XOR es calculada para los datos y entonces escrita simultáneamente.

Si un disco falla, los datos del mismo podrán ser reconstruidos leyendo el disco que contiene las operaciones XOR, así como los demás discos y realizando la operación XOR entre esos datos para obtener los datos faltantes.

En el nivel de RAID 3 el tamaño del segmento debe ser pequeño con respecto al tamaño de la petición de tal manera que todos los segmentos envueltos en la operación XOR sean transferidos por la petición. Se ejecuta una operación de lectura / escritura a la vez de tal manera que todos los discos miembro, se vean involucrados equitativamente.

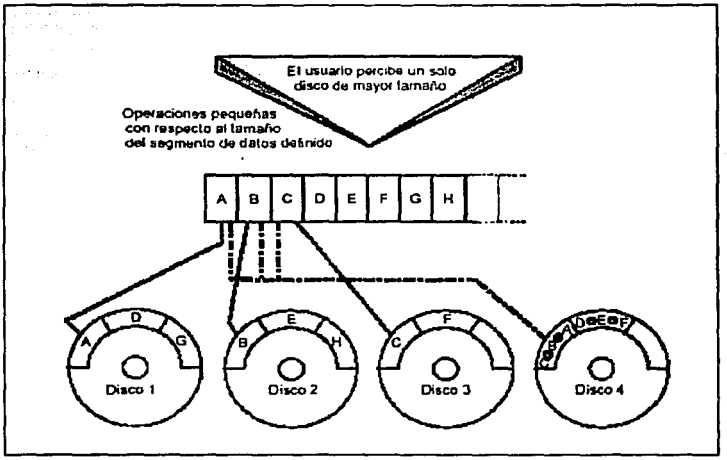


Figura 4.5.1-5 RAID Nivel 3.

Nivel RAID 2 y 4

El nivel RAID dos, opera de manera muy similar al nivel tres, excepto porque se tiene un número de discos redundantes formando un código Hamming en lugar del disco simple que almacena el resultado de las operaciones XOR. De esta manera, la redundancia es más costosa en este nivel.

El nivel RAID cuatro, es también similar al nivel tres. La diferencia primaria es que en este nivel los discos, miembro de un arreglo son administrados como discos independientes, mientras que en el nivel tres los mismos son administrados en conjunto.

Nivel RAID 5

El nivel de RAID cinco, proporciona las capacidades de distribución y redundancia de datos.

Como se muestra en la Figura 4.5.1-6 la redundancia se provee realizando la operación XOR pero a diferencia del nivel tres de RAID, en este caso el resultado de la operación se distribuye en todos los discos miembros del arreglo.

El segmento de datos A es físicamente almacenado en el primer disco. El segmento B se almacena en el segundo disco en la misma localidad física que el segmento A, el segmento C en el tercero, el primer segmento guardado en el cuarto disco contendrá la operación A Xor B Xor C. El cuarto segmento de datos D es colocado en el primer disco siguiendo a A; el quinto segmento, E, en el segundo disco siguiendo a B, El segundo segmento colocado en el tercer disco siguiendo a C contendrá la operación D Xor E Xor F y el sexto segmento, F, se coloca en el cuarto disco. Para el siguiente conjunto de segmentos, G, H, I la operación Xor será colocada en el segundo disco. Este patrón se repetirá hasta que toda la información del usuario pueda ser distribuida a lo largo de los discos. Para leer, la petición del usuario es mapeada contra la localidad física donde ella reside y entonces es leída.

Para escribir, la localidad es mapeada y entonces los datos previamente guardados en esa localidad y su Xor asociada son leídos. Se realiza entonces un Xor entre los datos que se encontraban previamente guardados y su propia Xor, de esta forma los datos anteriores son eliminados de su Xor asociada. Se realiza una operación Xor entre los nuevos datos y la Xor previa para obtener la nueva Xor. Entonces los nuevos datos y su correspondiente operación Xor son escritos en los dos discos envueltos a un solo tiempo. Si un disco falla, todos los demás son leídos, una operación Xor será realizada entre todos los segmentos existentes para reconstruir los datos perdidos.

Ya que casi todas las peticiones podrán ser cumplidas con un solo segmento, cada disco en el arreglo podrá ejecutar una operación de lectura simultáneamente. Las operaciones de escritura envolveran generalmente dos discos pero los discos restantes estarán disponibles para ejecutar otras peticiones. Debido a que las operaciones Xor se encuentran distribuidas en todos los discos, es posible ejecutar escrituras simultáneas.

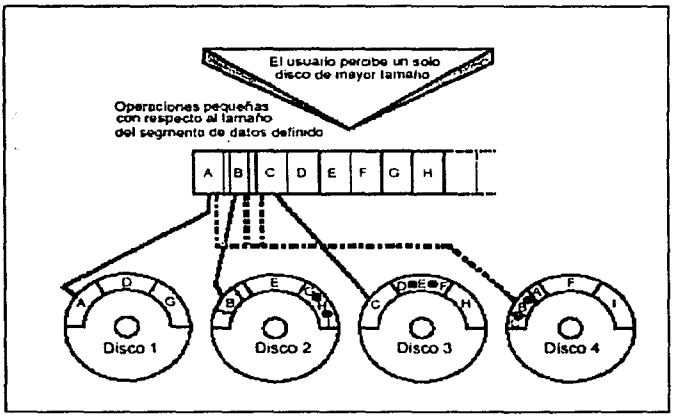


Figura 4.5.1-6 RAID Nivel 5.

Nivel RAID 6

El nivel de RAID seis provee capacidades de distribución de datos y redundancia.

El código de redundancia, el cual varía dependiendo de la implementación, se crea a partir de dos ecuaciones independientes (P y Q) Los segmentos redundantes son entonces almacenados de forma distribuida a través de todos los discos del arreglo.

La Figura 4.5.1-7 muestra que el primer segmento, A, es físicamente almacenado en el primer disco. El segundo segmento, B, es colocado en el segundo disco en la misma localidad física en la que A fue colocado en el primer disco. La primera localidad en el tercer disco contendrá la ecuación P(A, B) y el cuarto disco contendrá la ecuación Q(A, B). El tercer segmento, C, es colocado en el primer disco, siguiendo a A.

La segunda localidad en el segundo disco contendrá P (C,D) y el tercer disco contendrá Q(C,D). El cuarto segmento ,D, es colocado en el cuarto disco. La tercera localidad en el primer disco, contendrá P (E, F); la tercera localidad en el segundo disco contendrá Q (E, F). El quinto segmento, E, es entonces colocado en el tercer disco y F en el cuarto..Este patrón se repetirá sucesivamente hasta que la información se haya almacenado por completo.

Para leer, la petición del usuario es mapeada contra la localidad física donde ella reside y entonces es leída.

Para escribir, la localidad se mapea y los datos previamente almacenados en esa localidad y sus ecuaciones P y Q asociadas son leídas. Los datos anteriores son removidos de las anteriores P y Q. Los nuevos datos a ser escritos son entonces usados para calcular nuevas ecuaciones P y Q. Los nuevos datos y las nuevas P y Q, son entonces escritas en los tres discos involucrados.

En el evento de una falla de disco, todos los segmentos que permanecen más P y Q , son leídos y los datos faltantes, son entonces computados a partir de los datos conocidos. En el evento de la falla de dos discos, todos los discos que permanecen en buen estado, serán leídos y los datos faltantes computados a partir de ellos.

Ya que todas las peticiones podrán ser cumplidas manipulando un solo segmento, cada disco en el arreglo será capaz de ejecutar diferentes operaciones de lectura

simultáneamente. Una operación de escritura envolverá tres discos pero los discos restantes podrán ejecutar otras peticiones. Además, debido a que P y Q se distribuyen en todos los discos, se podrán ejecutar operaciones de escritura simultaneas si el arreglo es lo suficientemente grande.

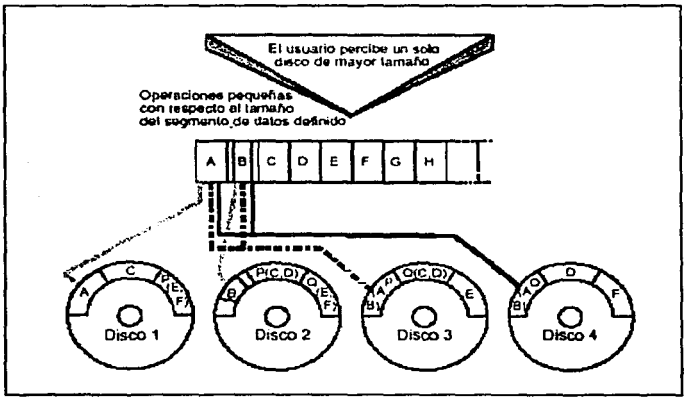


Figura 4.5.1-7 RAID Nivel 6.

Nivel RAID 7

Algunos proveedores definen un nivel 7 de RAID que utiliza un diseño asíncrono, en el cual, todos los discos operan de manera independiente entre ellos. Los usuarios pueden entonces configurar de uno a tres discos de paridad. Los beneficios potenciales son el alto desempeño y la tolerancia a fallas. Sin embargo, esta no es una definición ampliamente aceptada.

La Tabla 4.5.1-2 presenta un sumario del costo relativo, la disponibilidad y el desempeño que ofrece cada uno de los niveles RAID.

Nivel RAID	Costo	Disponibilidad	Desempeño
0	El mejor	Pobre	Bueno
1	Alto	Excelente	Bueno para lectura, pobre para escritura
0 + 1	Alto	Excelente	El mejor
3	Bueno	Bueno	Bueno
5	Bueno	Bueno	Moderado
6	Moderado	El mejor	Moderado
7			Depende de el proveedor

Tabla 4.5.1-2 Evaluación de los factores más importantes de los niveles RAID

4.5.2 Administrador Lógico de Almacenamiento (LSM)

En el tema anterior, describimos una de las soluciones más ampliamente usada para lograr la redundancia de almacenamiento en el nivel de hardware. En este tema presentaremos otro mecanismo útil para incrementar el nivel de disponibilidad del almacenamiento de datos.

Casi todos los sistemas UNIX actualmente cuentan con un manejador o administrador lógico de almacenamiento, en algunos casos se le conoce como Logical Storage Manager (LSM) y en otros como Logical Volume Manager (LVM). Para el caso es lo mismo y como su nombre lo indica, es un subsistema en el nivel

de software que nos permitirá manejar y/o administrar el almacenamiento de datos.

El administrador lógico de almacenamiento ayuda a reducir el downtime ya que utilizando la tecnología RAID en el nivel de software permite crear redundancia, reconstruir los datos de un disco dañado y ejecutar cambios en la configuración de almacenamiento sin necesidad de bajar el sistema.

El administrador lógico de almacenamiento generalmente provee las siguientes características:

- Concatenación (Spanning)

  Concatenación de los datos a través de los discos físicos, esta característica permite la creación de grandes file systems cuyo tamaño puede ir más allá del de un solo disco.

- Espejeo (Mirroring RAID1)

  Espejeo de datos críticos para protegerlos contra la pérdida de los mismos por corrupción o falla de hardware. Esta característica también ayuda a incrementar el performance de I/O.

- Distribución (Striping RAID0)

  Incrementa el performance de I/O distribuyendo el acceso a través de múltiples discos.

- RAID 0 + 1

  Como ya mencionamos, esta característica ayuda a proteger los datos y a incrementar el performance al mismo tiempo.

➢ RAID 5

Redundancia de datos usando paridad que permite la reconstrucción de los mismos en el caso de una falla de disco.

➢ Además sin necesidad de downtime se pueden realizar actividades como:

- Adicionar y borrar discos del sistema
- Crear o cambiar la configuración RAID
- Migración de datos por reparación de disco o balance de carga
- Separación de uno de los espejos para backup

➢ Disco de Respaldo (Hot Swap Spare)

Permite la re-localización y reconstrucción de datos de manera automática a un disco de respaldo previamente definido después de la falla de un disco de un arreglo RAID.

➢ Rapidez en la sincronización de datos

Mediante un mecanismo llamado Dirty Region Logging la velocidad de la sincronización de los datos en un arreglo RAID se incrementa ya que solamente son actualizados aquellos que cambiaron durante la falla.

Arquitectura

El Administrador Lógico de Almacenamiento esta formado por varios componentes que cooperan para proveer el manejo eficiente de los servicios de almacenamiento a las capas más altas del sistema operativo. El administrador se encuentra ubicado entre los sistemas de archivos (Si es que existen) y los dispositivos de almacenamiento físico. La Figura 4.5.2-1 muestra una pequeña descripción de la arquitectura del administrador. La misma incluye componentes tales como demonios, un manejador de kernel y un conjunto de archivos especiales.

Ya que nuestro objetivo no implica la descripción a detalle del administrador lógico de almacenamiento, no ahondaremos más en la explicación de cada uno de los componentes de la arquitectura.

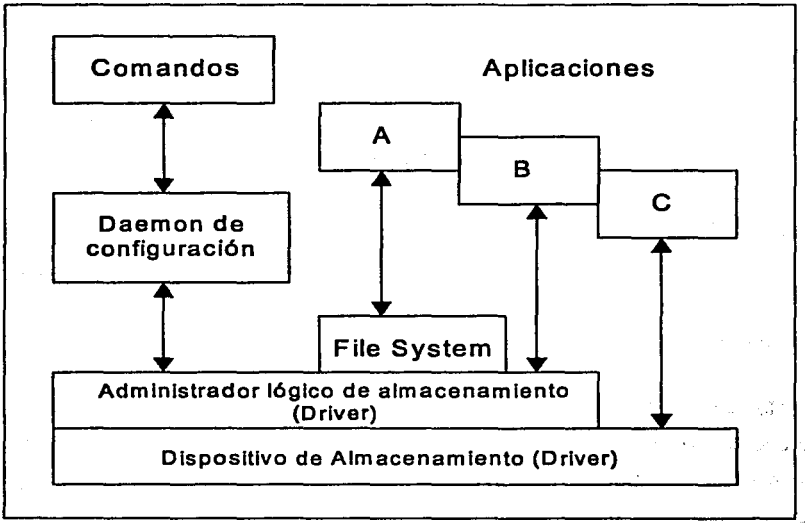


Figura 4.5.2-1 Arquitectura del Administrador Lógico de Almacenamiento.

Ventajas contra el RAID de Hardware

Ahora que conocemos un poco más acerca del administrador lógico de almacenamiento podemos pensar que es una implementación en el nivel de software de la tecnología RAID.

Y eso, es cierto!. Sin embargo, existen varias ventajas de esta implementación contra la implementación en el nivel de hardware:

- Provee la capacidad de espejeo a través de diferentes buses SCSI

- Provee elementos para realizar respaldos con mínimo downtime
- Provee flexibilidad para proveer espacios grandes y pequeños de manera muy sencilla

Finalmente, nuestra recomendación es utilizar esta solución de administración de almacenamiento como un complemento a la solución de RAID por hardware.

4.5.3 Sistema de Archivo de Cluster (CFS)

La clave para armar un cluster robusto es la idea de tener una sola imagen del sistema, esto es, todos los servidores en el cluster se ven como uno solo. Y haciendo posible que automáticamente se reconozcan los servidores que se agregan al cluster, además de balancear la carga de trabajo a través del mismo.

Al proporcionar una sola imagen del sistema, tanto el sistema operativo y los programas de la aplicación sólo se instalan una vez en todo el cluster. Además, de esta forma no es necesario configurar manualmente los privilegios de seguridad o algún otro atributo que deba mantenerse en sincronía con todos los miembros del cluster. Así el cluster se accede por medio de una sola dirección IP y los clientes no tienen que conectarse a un servidor en especifico.

La parte fundamental para una imagen única del sistema es un sistema de archivos orientado a clusters (Cluster File System – CFS) que permite que todos los nodos que lo componen compartan todos sus sistemas de archivos, es decir, la raíz común (common root), /var y /usr, así como también los datos de la aplicación. El CFS es la base del sistema de archivos local para cluster, como lo son el UFS y el CDFS en otros sistemas.

CFS se basa en muchos principios bien establecidos de los sistemas de archivos de UNIX, pero supera debilidades inherentes, y brinda la característica de alta disponibilidad, que es necesaria para los sistemas de misión crítica.

Sistemas de archivos de UNIX

Los sistemas operativos UNIX generalmente soportan varios tipos de sistemas de archivos, dando al administrador del sistema mucha flexibilidad para el manejo de recursos. AdvFS (Advanced File System) es un sistema de archivos para cluster que proporciona flexibilidad, compatibilidad, disponibilidad de datos, alto desempeño, y una administración simplificada, aunado a la capacidad de manejar y conjuntos de archivos de hasta 16 TB de longitud.

La configuración del AdvFS difiere de los de UNIX en que la capa física de almacenamiento se maneja de forma independiente de la capa de directorio. Por consiguiente, los administradores del sistema pueden agregar y quitar espacio de almacenamiento sin desmontar el sistema de archivos o tener que detener el sistema operativo. Además el AdvFS permite hacer defragmentación en línea, y algunas otras técnicas de almacenamiento avanzado. Desde el punto de vista del usuario, el AdvFS luce como cualquier otro sistema UNIX, pero con alto desempeño y alto grado de disponibilidad.

Aunque el AdvFS proporciona muchas ventajas, el cluster también soporta otros sistemas de archivos comunes de UNIX, como son:

UFS (UNIX File System) – Este es el primer sistema de archivos que usó UNIX, soporta archivos que exceden los 2 GB.
NFS (Network File System) – Es el sistema de archivos por medio del cual se comparten archivos en un ambiente heterogéneo de procesadores, sistemas operativos y entornos de red. Se puede montar un sistema remoto en un sistema local y hacer operaciones de lectura y escritura como si se trabajara localmente.
CDFS (Compact Disk File System) – Es un sistema de archivos local el cual se utiliza en los CD-ROM con formato ISO 9660.
DVD File System (DVDFS) – Se utiliza en los medios DVD.

MFS (Memory File System) – Es básicamente un sistema de archivos de UNIX que reside en memoria. Ninguna estructura de archivos es escrita a disco, por lo que el contenido del MFS se pierde cuando se reinicia o se apaga el sistema. Sin embargo, es un sistema de archivos muy rápido.

Características de los sistemas UNIX stand-alone

Los sistemas UNIX generalmente engloban los múltiples sistemas de archivos usando VFS (Virtual File System), El VFS presenta una interfaz uniforme a los usuarios y aplicaciones, permitiendo un acceso común a los archivos sin importar el sistema de archivos al que pertenezcan. Para el usuario, el acceso a través de diferentes sistemas de archivos es transparente.

La integridad de los datos también es confiable en los sistemas UNIX. Por ejemplo, el AdvFS se basa en el registro de sucesos (log-based), lo que permite una recuperación estable y, de esta forma, asegura la integridad de la información. El UFS usa un analizador del sistema de archivos para garantizar la integridad de los datos. Pero protocolos como el NFS proporcionan una baja integridad de los datos, debido a características propias del diseño.

Descripción del NFS

El NFS es el método más común en UNIX para compartir sistemas de archivos. Extiende un número de atributos de los sistemas de archivos locales a un ambiente distribuido, como lo es la transparencia y un intento por mantener el mismo desempeño cuando se acceden archivos remotos. La versión 3 de NFS también cuenta con mejoras como son el incremento del desempeño en operaciones de escritura por parte de un cliente, reducción de la carga de trabajo del servidor, soporte mejorado para sistemas con Listas de Control de Acceso (ACL), soporte para manejar archivos grandes (Gigabytes) en el servidor NFS.

A pesar de las mejoras que ha tenido el NFS, no es un sistema de archivos aceptable para usarse en ambientes cluster de alta disponibilidad. Por ejemplo, NFS se desvía de la semántica del sistema de archivos local porque tiene que re-localizar atributos a través de múltiples sistemas de archivos. En un sistema local, si un proceso escribe datos en un archivo y un segundo proceso lee la información de ese mismo archivo, en el segundo proceso se garantiza que leerá la información más reciente, esto es, mantiene coherencia en el cache del sistema de archivos; característica que NFS no tiene. Ya que si un proceso de un nodo escribe datos, y otro proceso de otro nodo intenta leer esa misma información, existe un lapso de tiempo considerable antes de que la nueva información sea actualizada en el disco, lo que implica que los usuarios pueden estar trabajando con información inexacta.

Este problema puede resolverse con el bloque de archivos, pero el desempeño del NFS se ve degradado de un 50 a un 90 por ciento.

CFS, una mejor solución

Para solucionar el problema de compartir sistemas de archivos en un ambiente de alta disponibilidad, fue que surgió el Sistema de Archivos de Cluster, CFS, el cual es un sistema de archivos que extiende todas las características de un sistema de archivos local a un ambiente de clusters, manteniendo la transparencia de operación, el desempeño, la integridad de los datos, y el fácil manejo, permitiendo además la escalabilidad y dando niveles altos de disponibilidad.

La parte fundamental para conservar las características de un sistema de archivos local es tener una sola imagen del sistema. CFS conserva la semántica de X/Open y POSIX tanto en el acceso al sistema de archivos, en interfaces de manejo de archivos, y en demás herramientas, haciendo que todo trabaje como en un sistema stand-alone.

Al proporcionar una imagen única del sistema, sin importar cuantos miembros existen en el cluster, los archivos son visibles y accesibles por todos ellos como si estuvieran en su sistema de archivos local. A diferencia del NFS, CFS mantiene una coherencia del cache, así si un proceso en uno de los nodos del cluster escribe información, no existen intervalos de tiempo en los que operaciones de lectura puedan leer datos sin actualizar. Gracias a esta coherencia del cache, todos los miembros del cluster perciben la misma información todo le tiempo. Además, cada miembro puede ser servidor y cliente de sistemas de archivos, lo que puede ser muy útil para hacer un balance de carga de trabajo al re-localizar sistemas de archivos entre dichos miembros.

Con CFS, aplicaciones y usuarios ganan acceso transparente a los archivos que se encuentren en cualquier sistema de archivos que soporte, sin importar en que lugar (nodo) se encuentre físicamente. Esto implica que no es necesario convertir los sistemas de archivos existentes a algún formato en específico para poder incorporarlos en el cluster. CFS tiene soporte para varios tipos de sistemas de archivos, incluyendo AdvFS, NFS (servidor y cliente), UFS, CDFS, DVDFS y PC-NFS (servidor).

La raíz común compartida y el espacio de nombres global del CFS dan al cluster acceso transparente a todos los dispositivos de almacenamiento, incluyendo dispositivos SCSI, cintas magnéticas, DVD y CDROM

CFS permite una administración sencilla ya que se tiene una partición de root compartida, además de poder compartir de igual forma la partición de /usr. Esto implica que el sistema operativo sólo tiene que ser cargado una vez y todos los archivos del sistema y de configuración son automáticamente compartidos con otros miembros del cluster. Las principales ventajas que nos da esto, es que agregar un sistema adicional al cluster es una operación muy rápida; y si se hace un cambio en la configuración, sólo se hace una vez, no es necesario hacerlo en todos los miembros del cluster.

Por otro lado CFS hace que el cluster tome todas las ventajas del AdvFS, es decir, nos permite agregar y quitar espacio de almacenamiento sin necesidad de desmontar el sistema de archivos o detener el sistema operativo en algún miembro del cluster. Esto, no solo ahorra tiempo, sino que le da al administrador una flexibilidad de asignar tareas sin interrumpir la operación habitual de la aplicación, lo cual podría afectar al usuario.

4.6

REDUNDANCIA DE OTROS COMPONENTES

Para obtener alta disponibilidad es necesario contar con elementos redundantes, además de los elementos ya mencionados. La importancia que se tenga de los elementos que a continuación se describen, dependerá de que tanta disponibilidad se requiera tener en nuestra aplicación.

Los elementos importantes a considera se pueden mencionar de acuerdo a la importancia que se tiene: Primeramente tenemos la instalación eléctrica, sistema de energía ininterrumpida un sistema de protección.

La instalación eléctrica interior y línea informática

Los problemas que se pueden suceder en una instalación eléctrica del interior de un edificio, son factores que pueden ocasionar grandes efectos. Causas como la conexión o desconexión de cargas inductivas como maquinaria, motores, ascensores, equipos de soldadura, compresores y el entorno de zonas industriales o industrias en particular, provoca este tipo de problemas, al cual se le añade

frecuentemente las dificultades de regulación, por parte de la compañía suministradora, debido al alto grado de variación en consumo de los mismos.

El criterio básico a tener en cuenta, en una instalación informática, es la instalación de una línea de suministro único al Sistema Informático, denominada comúnmente Línea Dedicada, y que alimenta al sistema desde la acometida de la red eléctrica (contadores). El conductor de Tierra, debe formar parte de esta línea dedicada. Al final de ésta línea y en función de la posibilidad de ejecución de la misma, los problemas descritos tendrán mayor o menor magnitud, la solución o atenuación de los mismos se puede realizar mediante diversos equipos, como son transformadores de aislamiento, estabilizadores, acondicionadores de red o SAI[5].

Sistemas de alimentación ininterrumpida (SAI's - UPS)

Son equipos que por su concepción autónoma, permiten realizar suministro aún cuando no exista suministro de red. Para ello incorporan baterías, cargador de baterías y ondulador, la finalidad de este último, es convertir la corriente continua procedente de los acumuladores, en corriente alterna, de iguales características que la red, pero exenta de los problemas de ruidos y variaciones que la afectan. Las prestaciones más generales que deben aportar dichos equipos son: Aislar la carga que se alimenta de la red, estabilizar el voltaje y la frecuencia de salida, evitar picos y efectos parásitos de la red eléctrica. Almacenar energía en las baterías, las cuales la suministrarán por un periodo fijo de tiempo, cuando haya un corte de corriente. Ésta energía almacenada permitirá llevar a cabo la salvaguarda de la información y el cierre normal del ordenador.

[5] Sistema de alimentación ininterrumpida

Diseño de un sistema de protección integral

Para el diseño e instalación de elementos ó equipos de protección integral, deberemos conocer en primera instancia la vulnerabilidad de los equipos a proteger. Recordemos que los factores que solían afectar al correcto comportamiento de sistemas electrónicos, eran: Regulación, Transitorios, Ruidos, Armónicos, Tierra y Cortes de suministro. En nuestro caso, el primero, cuarto y sexto, quedarán resueltos mediante la aplicación de un Sai. El quinto factor o Tierra, dependerá de la construcción de la misma, es recomendable utilizar un Tierra exclusivo para informática ó equipos críticos y otro para maquinaria. En función de la calidad del suelo, se instalarán las piquetas suficientes para asegurar una muy baja impedancia. Factor ruidos: Depende en gran medida de una correcta instalación de los buses de datos y comunicación, en ningún caso deben discurrir a líneas paralelas de suministro.

Además de los sistemas complejos que se tienen que implementar, es necesario contar con elementos redundantes menos complejos, pero igual de eficaces en caso de fallas.

1 Conectores múltiples de energía eléctrica
2 Conectores múltiples de nodos de red
3 Lectoras de cintas para respaldos
4 Hardware de uso común (Mouse, teclados, etc) propenso a fallas

4.7

SOFTWARE

Desde que fue desarrollada la idea del cluster como solución a la disponibilidad de información de un sistema, ha surgido la necesidad de programas especiales para su funcionamiento (software) del mismo cluster. El software que utilizan actualmente los clusters modernos varía conforme al diseño del mismo cluster ya que las compañías encargadas de crear el hardware por lo regular también se encargan de suministrar el software que requiere su equipo.

En muchas ramas de las ciencias, la complejidad de los problemas que se estudian requieren contar acceso a una supercomputadora, siendo éstas máquinas poderosas que pueden desarrollar varios miles de millones de operaciones por segundo. Las supercomputadoras tradicionales emplean procesamiento en paralelo, contienen arreglos de microprocesadores ultrarrápidos que trabajan en sincronía para resolver problemas complejos como pronósticos numéricos del estado del tiempo, o modelar estructuras complejas de la materia.

Sin lugar a duda un cluster presenta una alternativa importante para varios problemas particulares, no solo por su economía, sino también porque pueden ser diseñados y ajustados para aplicaciones específicas que requiera la empresa en cuestión. No obstante el uso de software para el implementación de un cluster es muy reducido y no existen muchas opciones en el mercado. Por mencionar algunos de los tantos programas que existen en el mercado se mencionaran algunos de los más usados.

Mosix: Mosix es una herramienta desarrollada para sistemas tipo UNIX, cuya característica resaltante es el uso de algoritmos compartidos, los cuales están diseñados para responder al instante a las variaciones en los recursos disponibles, realizando el balanceo efectivo de la carga en el cluster mediante la migración automática de procesos o programas de un nodo a otro en forma sencilla y transparente.

El uso de Mosix en un cluster de PC's hace que éste trabaje de manera tal, que los nodos funcionan como partes de un solo computador. El principal objetivo de esta herramienta es distribuir la carga generada por aplicaciones secuenciales o paralelizadas.

Una aproximación de balanceo de carga es realizada por los usuarios a la hora de asignar los diferentes procesos de un trabajo paralelo a cada nodo, habiendo hecho una revisión previa de forma manual del estado de dichos nodos.

HPF (High performance fortran) : Es un conjunto de extensiones para Fortran 90 que permite a los programadores especificar como los datos son distribuidos a través de múltiples procesadores en un ambiente de programación paralela. La construcción del HPF permite a los programadores utilizar el potencial de paralelismo a un relativamente alto, sin entrar dentro de los detalles de bajo nivel del pase de mensajes y sincronización. Cuando un programa HPF es compilado,

el compilador asume la responsabilidad de organizar las operaciones paralelas en una máquina física, reduciendo enormemente el tiempo y esfuerzo para el desarrollo de programas paralelos. Para aplicaciones, los programas paralelos pueden ejecutarse dramáticamente más rápido que los programas Fortran ordinarios.

MPI (Message Passing Interface): El objetivo principal de **MPI** es lograr la portabilidad a través de diferentes máquinas, tratando de obtener un grado de portabilidad comparable al de un lenguaje de programación que permita ejecutar de manera transparente, aplicaciones sobre sistemas heterogéneos.

PVM (Parallel Virtual Machine): es un paquete de software que permite a una colección heterogénea de computadoras, con sistema operativo UNIX, que estén conectadas a través de una red, ser usadas como una sola máquina paralela.

Para inicializar y ejecutar **PVM** deben configurarse dos variables de ambiente. La primera PVM_ROOT indica el lugar donde esta instalado **PVM**. La otra variable PVM_ARCH indica la arquitectura del servidor desde donde se está invocando **PVM** y de esta manera seleccionar los archivos ejecutables más apropiados.

El software de TruCluster nos da la solución para la administración del cluster a implementar, cabe destacar que el uso más común de este software es en maquinas Alpha, que además corre en Tru64 Unix, el sistema operativo de 64 bits más potente y probado del mercado y en sistemas Alpha server basados en la tecnología de proceso más potente de la industria como es Alpha. El propietario de está tecnología es Compaq.

Adicionalmente al software que existe para la administración y el control del funcionamiento del cluster existe específicamente un software para el control del almacenamiento de bases de datos y esta tiene el nombre de Oracle Parallel Server (OPS).

Oracle Parallel Server esta diseñado para dividir el trabajo en varios nodos. Para aprovechar todas la ventajas de OPS se divide distribución de datos, transacciones distribuidas y mantener el estado de los distintos nodos. El sistema debe cuidar la integridad de los datos almacenados en memoria cache y ser bloqueados en el momento que un usuario requiera acceder a ellos. Esta ventaja que presenta OPS es significativa, ya que si solo se tuviera un manejador de base de datos de una sola instancia no se estuviera aprovechando el cluster al cien porciento.

Esto se vuelve complicado, ya que en el sentido de usar varios nodos de trabajo el servidor de base de datos debe controlar todas las peticiones de datos teniendo cuidado en la integridad de la información. Debido a que la información es accedida en un arreglo de discos se debe contar con una interconexión a cada uno de los nodos de manera eficiente para adquirir una base de datos más robusta y con un desempeño alto.

Cuando un nodo del cluster tiene una falla, la operación del manejador de base datos, OPS, seguirá funcionando normal, debido a que OPS se encuentra corriendo en cada uno de los nodos del cluster y en el momento que un usuario requiera una petición simplemente se conectará al servidor que se encuentra activo. Si se instala un servidor de base de datos normal, digamos en una sola instancia, el proceso de petición de conexión a la base de datos no sería transparente y esto se debe a que, si el nodo donde se encuentra corriendo el manejador de base de datos falla se debe correr un proceso de relocalización al nodo que será activo.

4.8

SERVICIOS O APLICACIONES ALTAMENTE DISPONIBLES

Servicios

El término *servicio* es utilizado para describir los programas que hace altamente disponible, El modelo servicio proporciona acceso de red al almacenamiento disponible a través de su propio protocolo cliente-servidor. Ejemplos de los servicios ASE (Ambiente de servicios disponibles, por sus siglas en inglés) son el NFS (Network File System) y las bases de datos. Usualmente, un conjunto de programas o pasos de procesamiento necesitan ser ejecutados secuencialmente para iniciar o detener un servicio. Si cualquiera de los pasos no puede ser ejecutado con éxito, entonces el servicio no se puede proporcionar o no se puede detener, en el caso de que estuviera corriendo. Obviamente, si no podemos acceder al almacenamiento disponible, no se podrá inicializar el servicio. El servicio ASE proporciona una infraestructura general para especificar los pasos de procesamiento y las dependencias de almacenamiento de cada servicio.

Eventos y modos de falla

El servicio ASE monitorea el hardware y software para determinar el estado del ambiente. Un cambio del estado es reportado como una notificación de evento al software ASE. Ejemplos de eventos, son la falla y recuperación de un servidor, alguna falla de red o de discos, o algún comando de las utilidades del ASE.

Servicios de failover

El software ASE responde a los eventos relocalizando los servicios de un nodo a otro. Una relocalización debido a la falla del hardware es referido como servicio de failover. Existen otras razones para relocalizar un servicio. Por ejemplo, un manejador de sistema puede relocalizar un servicio con el fin de balanceo de cargas o dar de baja un nodo para darle mantenimiento.

Políticas en la relocalización de servicios

Si un servicio debe de ser relocalizado, el sistema ASE utiliza políticas configurables para determinar cual es el mejor nodo para correr el servicio. Las políticas están en función de los eventos y del manejador del sistema instalado preferente para cada servicio. Por ejemplo, un servicio debe ser relocalizado si el nodo en el cual está corriendo se cae o si el cable SCSI es desconectado. El manejador del sistema puede especificar hacia que nodo será relocalizado el servicio. La preferencia puede además ser proporcionada por el nodo. Por ejemplo, el manejador del sistema puede especificar que un servicio retorne siempre a un nodo específico, si éste se encuentra en operación. Para los servicios que toman largo tiempo en inicializarse, el manejador del sistema puede especificar que un servicio sea relocalizado solamente si su nodo tuvo una falla.

4.9

SCRIPTS

Para poder entender que es un script es necesario definir primeramente al shell (Interpretador de comandos en Unix).

En primera instancia Unix es un sistema operativo, éste tiene la función de actuar como mediador con el CPU y sus diversos componentes tanto de software como de hardware.

Cuando se abre una sesión de trabajo en Unix, la pantalla que vemos es en realidad un programa que corre en sistema operativo Unix y que esta monitoreando y responde a las acciones del teclado, a este programa se le llama shell de conexión (login shell). El shell de conexión sirve de interfaz entre el usuario y el sistema operativo Unix.

Es importante entender que cuando se introducen comandos por el teclado el shell los interpreta. Por ejemplo :

- Digamos que vamos a ejecutar un programa que muestra la estadística de procesos (Programa ps) , el programa ps se ejecuta dentro del programa shell, se podría visualizar mejor esto si se ve al shell como una esfera dentro de la cual se ejecutaran diversos programas

Los programas shell pueden ejecutar otros programas dentro de ellos mismos incluso pueden ejecutar otros shell dentro de ellos, ha este tipo de shell se les denomina shell hijos.

Un archivo script consiste en un archivo de texto ordinario que contiene una serie de comandos shell. Los comandos del shell son de forma libre y con solo unas pocas reglas de sintaxis.

La programación shell utiliza a los scripts casi siempre para la automatización de tareas. El shell posee muy pocas herramientas para poder interactuar con los usuarios, a lo más puede hacer una pregunta al usuario y recibir respuesta en forma de texto.

Los scripts son comúnmente utilizados para personalizar instrucciones o programas shell normales. Los scripts también son útiles para:

- Verificar la utilización del disco
- Mantener los logs del sistema
- Monitorear la actividad del usuario
- Y cientos de tareas más

Dos de los lenguajes más utilizados para la programación de scripts son Bash Perl y Tcl.

El Bash es lo que se considera comúnmente como shell, en tanto que Perl y Tcl son considerados lenguajes de programación, los tres pueden utilizarse para la creación de programas.

Un script tiene al igual que otros lenguajes de programación posee variables y ordenes de control de flujo (for, if , while, etc).

Los script como ya se mencionó, sirven para automatizar tareas dentro del sistema, su objetivo principal es la no intervención del usuario ya que se pretende la automatización del mismo sistema por medio de la toma de decisiones de los scripts.

Un scripts puede ser relativamente muy sencillo hasta llegar a ser realmente complejo todo depende de las necesidades del sistema o de los usuarios. Los scipts son hechos específicamente para el sistema donde se está trabajando ya que cada script se personaliza de acuerdo a las necesidades del sistema y puede ir siendo modificado durante el transcurso del tiempo si así se requiriera.

Una forma muy sencilla de script seria la siguiente :

- Use el comando **vi** y asigne un nombre en seguida **vi hola**.
- Pulse la letra **i** para entrar al modo de inserción.
- Agregue la primera línea de su script, la cual como ya se sabe será alguna instrucción.
- Pulse **scape** y teclee **wq** para guardar su script.
- Ejemplo :
    - #!/bin/bash
    - #"Hola Mundo"

para poder ejecutar este script o cualquier otro script se usa los siguientes comandos:

- % chmod +x hola
- % ./hola

La primera línea del script le dice al shell que programa debe emplear cuando ejecute el script. En este caso es el archivo /bin/bash.

La instrucción chmod modifica los permisos del archivo hola de forma que el shell pueda ejecutarlo empleando simplemente el nombre del archivo en la línea de comandos.

Existen formas más complejas para la creación de scripts. Claro ésta fue una de las formas más sencillas para que el ejemplo fuera fácilmente comprensible. En sí un script es traducido como escritura, esto nosotros lo entendemos como las instrucciones que van ir dentro del shell, dicho de otra manera los script son tan solo las instrucciones que van a ser ejecutadas dentro del shell, como ya se había mencionado, pueden ser relativamente sencillos como lo mostró el ejemplo anterior y pueden llegar a ser extremamente complejos dependiendo de las necesidades del sistema o de los usuarios.

Existen diversos comandos para el control de la ejecución de los scripts, como son:

- **break** : El comando **break** abandona el bloque de comandos actual
- **case** : el comando **case** es una alternativa al comando if, se emplea normalmente cuando se necesita tomar alguna decisión con base en una variable que tiene muchos valores posibles
- **continue** : El comando **continue** salta los comandos restantes de un bloque de comandos, se emplea en conjunto con los comandos **for** , **while** y **until**
- **exit** : El comando **exit** fuerza la terminación del script, este es muy util cuando se detectan condiciones de error

- **for** : el comando **for** lista sobre un conjunto de valores ejecutando un bloque específico de enunciados una vez por cada valor de la lista
- **If** : El comando if nos permite decidir entre uno o más cursos de acción
- **trap** : El comando **trap** le permite a su script interpretar señales y actuar sobre ellas. Si su script esta realizando algo particularmente vital, tal vez desee atrapar ciertas señales para evitar interrupciones. El comando trap también es utilizado para ejecutar comandos de limpieza, exactamente antes de que el script termine
- **until** : el comando **until** ejecutará repetidamente un bloque de comandos hasta que sea afirmativa la prueba (regrese un cero)
- **While** : el comando while, al igual que el comando until, repite un bloque de comandos. Sin embargo en lugar de parar cuando la prueba tiene éxito, se detiene cuando la prueba falla es decir cuando regresa un valor distinto a cero

Básicamente para el caso que a nosotros interesa el cual es la alta disponibilidad en sistemas, los scripts juegan un papel crucial como herramientas en la automatización de los procesos.

Básicamente existen cuatro tipos de scripts, para los clusters en alta disponibilidad.

- Script stop
- Script star
- Script de monitoreo
- Script de errores

A estos tipos de scripts se les conocen como Action Scripts, cada uno de estos scripts tiene una función especifica asignada dentro del sistema y entrarán en acción cada vez que el mismo sistema lo requiera.

La forma básica en que funcionan este tipo de scripts es la siguiente : por primera instancia el script ejecuta todos los comandos necesarios para comenzar la aplicación y debe volver 0 (cero) para el éxito y un valor no-cero para el incidente.

- Parar un script la forma en que trabaja este script es similar a la general, primero ejecuta todos los comandos necesarios de la aplicación, se debe devolver un cero para el buen funcionamiento de la aplicación y un no cero para el incidente o falla (Si el script no encuentra nada que parar devolverá un cero).

Para poder crear un script de acción es necesario que cumpla ciertos requisitos como son:

- La aplicación debe ejecutarse solamente en un sistema a la vez
- La aplicación debe poder ser inicializado o parado por un conjunto de comandos en una orden especifica

Cuando se instala un servicio, todos sus comandos se incluyen en un conjunto de programas el cual tiene por nombre Scripts de acción (Action scripts).

El software que utiliza un cluster, usa a los scripts de acción para detectar fallas en los servicios, esto se logra con el monitoreo de cada paso o instrucción del script y revisa que esta se halla concluido correctamente antes de seguir verificando la instrucción siguiente, este proceso se llevara acabo en todas las líneas del script hasta verificarlo por completo.

Hay cinco tipos de script de acción: agregar, suprimir o parar, comienzo o inicio alto o parar , y los de control. Además, hay dos versiones de cada tipo de script de la acción: script internos los cuales son definidos por el usuario de la acción. Estos scripts de acción se ejecutan en los momentos específicos y realizan tareas específicas.

Las descripciones de estos cinco tipos de scripts de acción son como sigue:

- **Agregar :** Después de instalar una nueva aplicación o servicio, el software del cluster ejecuta cada una de las instrucciones y agrega el script de la acción en todos los sistemas del miembro para configurar el servicio en los miembros. El software se ejecuta y agrega los script de acción en todos los miembros porque cada miembro del sistema debe poder ejecutar cada servicio. Un script de acción contiene todos los comandos que se necesitan para instalar el ambiente del sistema para permitir al servicio ejecutarse. Por ejemplo, un script de acción de la adición podía corregir ficheros del sistema.
- **Cancelar :** un script de cancelación se utiliza para suspender un servicio. El funcionamiento de este script es el siguiente: primero se ejecuta en todos los miembros y se verifica que servicio se requiere cancelar. También este script suprime las acciones de los script de acción de cualquier tipo de adición que se realice, por ejemplo : Si un script de acción de adición realizó cambios en un fichero, el script de acción de cancelación corregirá el mismo fichero y quitara los cambios hechos en él
- **Inicio (Start):** Los scripts de acción de inicio se usan cuando el software comienza un servicio en el sistema de algún miembro, este se ejecuta solamente en ese miembro del sistema. Este tipo de script contiene todos los comandos necesarios para empezar la aplicación en algún miembro del sistema, por ejemplo : Este script puede invocar a la aplicación y hacerla altamente disponible.
- **Parar (Stop):** El script de acción que sirve para parar una aplicación lo usan los miembros del sistema para poner fin a las aplicaciones de un determinado servicio de algún miembro. Por ejemplo, si algún script de acción desea modificar alguna aplicación, es necesario que primero se ejecute el script de paro; ya que si éste no para, todos los

procesos a los que tiene acceso tendrían problemas y no se ejecutarían los cambios realizados.

- **Monitoreo** : Los script de acción de monitoreo tienen la función de revisar si un servicio se está ejecutando. Cuando un proceso empieza, lo verifica y una vez revisado continua su ejecución. Los servicios son quienes invocan a este tipo de script. Es importante aclarar que cuando un servicio es suprimido también se invoca a este script para verificar su estado.
- Script de monitoreo
- Script de errores

La forma básica del funcionamiento de este tipo de scripts es la siguiente:

Por ejemplo el script de stop solo entra en acción cuando un componente del sistema esta dañado ya sea hardware o software, su función básicamente es la de apagar el sistema y redireccionar las carga de trabajo a los demás nodos. Está acción preventiva la toma el script para evitar mayores problemas en caso de avería o falla de software o hardware.

La contraparte del script de stop es el script start, este tiene como función principal el reestablecer el sistema cuando las fallas hayan sido corregidas o sea, cuando el sistema no detecta fallas dentro del mismo y que por alguna razón haya tenido que entrar el script de stop (Podría ser una falla eléctrica) el script de start evalúa el sistema y si no encuentra problema alguno, se dispone a reestablecerlo automáticamente.

Por otra parte los script de monitoreo continuamente se encuentran verificando los componentes del sistema tanto de hardware como de software, estos tienen la función de notificar al mismo sistema cuando algún componente está fallando.

Por último, un script de manda los mensajes de falla cuando algún componente del sistema ya sea de hardware o de software, ha fallado y necesita una revisión para evitar fallas posteriores.

5

CASO DE ESTUDIO: "CONFIGURACIÓN DE UNA APLICACIÓN ALTAMENTE DISPONIBLE EN UN CLUSTER SOBRE PLATAFORMA UNIX"

5.1

ANÁLISIS

5.1.1 Planteamiento del problema

Una empresa de reciente creación ha visto incrementada la venta de productos a través de su portal en Internet. Sin embargo, en los últimos meses debido a diversas fallas tanto de hardware como de software en el equipo donde reside el servidor web, el portal ha estado fuera de producción o no ha estado disponible en varias ocasiones por lo que la venta a través del mismo se ha visto impactada de manera considerable.

Debido al auge de éste tipo de sistemas, a que las operaciones de compra-venta en Internet a diferencia de las realizadas tradicionalmente causan un costo menor y a la necesidad de tener presencia y capacidad de venta en un esquema de 24 horas al día, es de vital importancia para la empresa encontrar una solución de cómputo que le permita mantener e incrementar el ingreso que tiene a través de la venta por Internet.

El acceso continuo al portal y la confiabilidad que se tenga acerca del registro de la transacción en su base de datos, le permitirá a esta empresa incrementar el margen de ventas en el segmento en línea (*On line*) del mercado y proporcionar un servicio eficiente a sus clientes.

El objetivo es entonces incrementar la disponibilidad del sistema de cómputo en el cual reside tanto el servidor web (Portal) como la base de datos de transacciones realizadas por los clientes (Registro de compras en línea) de la empresa, a través del diseño e implementación de una solución probada, robusta y con capacidad de proveer un nivel aceptable de disponibilidad.

5.1.2 Análisis del Problema

Como se mencionó en el planteamiento del problema, el procesamiento que realiza el sistema en cuestión, es de misión crítica, pues la información que opera a través de la base de datos y el portal web es fundamental para el buen funcionamiento de la compañía. Es necesario entonces, implementarlo sobre una solución de misión crítica ya que la misma nos proporciona grandes ventajas tales como: Tiempo muy corto de downtime, cuenta con mecanismos de seguridad que permiten integridad y disponibilidad, así como auditoria y control de acceso a la información.

Las alternativas en el proceso de selección de una solución son variadas. La solución de Recuperación de Desastres ofrece el más alto nivel de disponibilidad E5, que se traduce en menos de 6 minutos fuera de servicio al año (99.999% disponibilidad). Centra su operación en contrarrestar los efectos que provocan desastres de mayor magnitud sobre los sistemas de cómputo a través de procesar la información de forma simultánea, en un site primario y en un site remoto. Evidentemente el costo de esta opción se dispara considerablemente.

Los equipos Tolerantes a Fallas proporcionan un nivel de disponibilidad E4, es decir menos de 60 minutos de sistema caído anualmente (99.99% de disponibilidad), la disponibilidad que proporciona a los sistemas es significativa, aún cuando exista cualquier falla en los componentes de hardware la ejecución no se detiene. Cada componente del equipo se encuentra duplicado, esto implica también un costo importante.

El siguiente nivel de disponibilidad E3, representa menos de 9 horas de sistema inhabilitado (Con un 99.9%) y puede ser proporcionado por la tecnología cluster; Ambiente integrado que proporciona alta disponibilidad, fácil crecimiento del sistema, desempeño y escalabilidad por medio de compartir los recursos en por lo menos un par de nodos.

De las soluciones mencionadas anteriormente, mismas que ofrecen altos niveles de disponibilidad podemos concluir que la solución más viable respecto de los recursos y necesidades de la compañía actualmente, es la tecnología cluster. Creemos que representa una solución satisfactoria a los requerimientos. Implica un menor costo de implementación y no se necesita personal altamente especializado para su operación. Al igual que las soluciones de Recuperación de Desastres y Tolerancia a Fallas, el tiempo de disponibilidad que proporciona es satisfactorio (Menos de 9 horas de sistema caído al año), los componentes que lo integran son estándares, se necesita un único site para su ubicación y por ende menor costo en su operación y administración.

La tecnología cluster que usaremos para solucionar la problemática planteada es necesaria pero no suficiente, y para elevar aún mas la disponibilidad es importante considerar otros factores, tales como un ambiente controlado en el centro de cómputo, procesos de administración eficientes, personal capacitado, documentación, seguridad y otros mas.

Para ello instalaremos y configuraremos un cluster de dos miembros usando dos servidores tipo RISC operando sobre plataforma UNIX. Además se minimizarán al máximo los puntos simples de falla del cluster utilizando algunas soluciones de hardware y software existentes para tal efecto (RAID, NetRAIN, Multibus, etc.).

El tipo de cluster que estaremos configurando será un Activo/Activo ya que con ello lograremos el balanceo de cargas de trabajo y de esta manera evitaremos tener ocioso alguno de los nodos. El mismo proporciona un tiempo de recuperación de 15 a 90 segundos y al mismo tiempo nos proporciona un nivel de disponibilidad E3 necesario para las operaciones ininterrumpidas con recuperación automática. Algo muy práctico de esta solución es que proporciona niveles de escalabilidad muy altos. Lo que permitiría ampliaciones del sistema a futuro.

Aprovechando la migración del sistema, del servidor simple hacia la nueva solución de cluster y dado que conocemos el espacio en disco utilizado actualmente así como el promedio de crecimiento mensual del mismo; realizaremos la planeación de la capacidad del espacio en disco que se requiere para almacenar la base de datos en los próximos dos años.

5.2

DISEÑO

5.2.1 Planeación de la configuración de hardware

Es necesario llevar acabo la planeación de cada uno de los componentes de hardware que se verán involucrados en la implementación del cluster. La cual se realizará en el siguiente orden:

- Miembros del cluster
- Bus de almacenamiento compartido
- Subsistema de almacenamiento
- Red empresarial
- Cluster interconnect

Miembros del cluster

Debido a que en esta implementación se estará usando la versión 5.1 de Trucluster, es necesario que cada sistema miembro del cluster tenga instalada como mínimo la versión 6.1 de firmware. Esto debido a que la versión 5.1 solo esta soportada a partir de la versión de firmware mencionada.

Las características de cada uno de los nodos se muestran en la Tabla 5.2.1-1.

Componentes	Descripción	Cantidad
Servidor	AlphaServer ES40	2
CPU	Procesador RISC tipo Alpha a 500 MHz	2
Memoria	Memoria RAM	512 MB c/u
Adaptador SCSI	Interfaz tipo F.C.para conectividad con el bus de almacenamiento compartido	4
Adaptadores de Red	Tarjeta Ethernet 100Base-TX para conectividad con red empresarial	4
	Tarjeta Ethernet 100Base-TX para conectividad con el cluster interconnect	4

Tabla 5.2.1-1 Características de los nodos del cluster.

Para el caso de los dos nodos se debe tomar en cuenta la conexión de energía eléctrica, los cables de conexión a la red privada y el uso de otros componentes que se requieran para su implementación.

Bus de almacenamiento compartido

Para realizar la instalación del bus de almacenamiento compartido se debe contar con los siguientes elementos físicos.

El adaptador SCSI que conectará cada uno de los nodos del cluster con el bus de almacenamiento compartido que en este caso usa tecnología Fiber Channel. Por lo cual cada equipo deberá contar con un módulo PCI.

Cable de fibra óptica, el cual nos permite una longitud máxima de 500 metros entre el adaptador PCI SCSI y el Fiber Channel switch.

Fiber Channel switch, donde es recomendable que cada switch tenga un puerto de tipo RJ45 para facilitar su configuración y posteriormente el monitoreo de la red de almacenamiento (Storage Area Network - SAN)

El subsistema de almacenamiento, que en este caso contará con controladores modelo HSG80 y el cual albergará los arreglos de disco a utilizar (Para más referencia véase discos de almacenamiento).

Memoria cache, para cada controlador la cual por motivos de performance es recomendable sea superior a 512 MB.

Se debe tomar en cuenta que el bus de SCSI compartido deberá ser un bus externo a cada equipo.

Si realizáramos la interconexión de cada uno de los elementos mencionados, utilizando un solo switch, como lo muestra la Figura 5.2.1-1, no se estaría consiguiendo el objetivo de tener una configuración de alta disponibilidad.

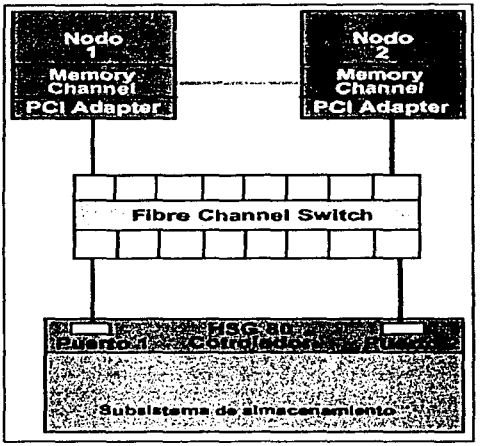


Figura 5.2.1-1 Configuración con puntos simples de falla.

La Figura 5.2.1-1 muestra una de las configuraciones validas para la interconexión de los elementos de un cluster. Sin embargo, la misma presenta varios puntos simples de falla los cuales pueden llegar a generar los siguientes problemas:

- Falla en adaptador PCI – SCSI
- El medio físico por donde se transmiten los datos, el cable de fibra óptica puede tener fallas
- Como se esta haciendo uso de un solo switch, éste puede fallar e interrumpir la conexión que mantienen los nodos y el controlador de discos HSG80
- Incluso el controlador de discos HSG80 puede tener fallas.

Para obtener mayor disponibilidad disminuyendo al máximo la posibilidad de interrupción por falla de un componente único en el bus SCSI compartido, utilizaremos el concepto de redundancia para eliminar los puntos críticos de falla anteriormente mencionados. La Figura 5.2.1-2 muestra la configuración resultante

del bus de almacenamiento compartido usando una configuración sin puntos simples de falla.

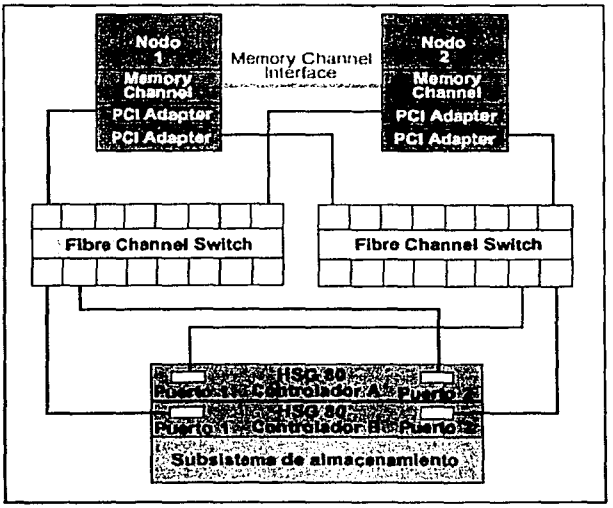


Figura 5.2.1-2 Configuración sin puntos simples de falla.

Si bien se observa una conexión mas complicada, lo cierto es que este tipo de conexión provee alta disponibilidad debido a que se tuvo el cuidado de adicionar redundancia en aquellos elementos que en cualquier momento pueden tener fallas.

Si una falla se llega a presentar en un adaptador PCI - SCSI de uno de los nodos, existiría un breve momento de interrupción en la comunicación solo mientras el mismo sistema operativo realiza una operación de failover entre adaptadores y entonces el segundo adaptador retomará la operación normal. De igual manera sucedería con los elementos considerados como puntos críticos, tales como los cables de fibra óptica, los switches de conexión y los controladores HSG80.

La configuración del subsistema de almacenamiento será tratada un poco más adelante.

En conclusión, es importante contar con el hardware descrito en la Tabla 5.2.1-2 para que la interconexión del bus de SCSI compartido sea la más eficiente y la que provea mayor disponibilidad al cluster en el cual será implementado el caso de estudio.

Cantidad	Descripción
4	Adaptadores PCI-to-Fiber Channel, dos por cada nodo
8	Cables de fibra óptica
2	Fiber channel switch
2	Controladores HSG80
2	Módulos de memoria cache para el controlador HSG80

Tabla 5.2.1-2 Elementos necesarios para conexión del bus de almacenamiento

Subsistema de almacenamiento

La disponibilidad de los datos es importante, por lo que entonces también es importante la correcta elección de la estrategia de almacenamiento de los mismos. Ésta dependerá de la capacidad de almacenamiento que el cluster requiera además, de dos factores muy importantes en cualquier sistema de cómputo:

- Disponibilidad
- Desempeño

Pero, ¿qué se debe hacer para obtener una adecuada disponibilidad de los datos?. Para lograr una buena disponibilidad en el almacenamiento de los datos es necesario primero definir la estrategia de configuración de los discos.

La solución que existe actualmente, es la de proveer redundancia de discos mediante el uso de un arreglo tipo RAID. Sin embargo, como se explicó en el tema 4.5.1, existen varios niveles de arreglos RAID que se pueden utilizar, cada uno ofreciendo diferentes características de redundancia y desempeño.

¿Cual será el nivel RAID de arreglo de discos más adecuado a implementar?

Para responder dicha pregunta nos apoyaremos en el tema 4.5.1 y específicamente en la Tabla 4.5.1-2. En ella se describen los diferentes niveles de arreglos de disco existentes así como el análisis de la disponibilidad y el desempeño que ofrece cada uno de ellos.

Nuestra implementación evaluará los niveles RAID 0+1 y 5. La selección de estos dos niveles se debe a que los niveles RAID 0, 1, 6 y 7 no cumplen con las necesidades de disponibilidad y desempeño. Ahora, el nivel 3, sería una buena elección, sin embargo no ofrece la mejor seguridad de almacenamiento y esto se debe a que la redundancia que ofrece se encuentra en un solo disco a diferencia del nivel 5 en donde la redundancia se encuentra distribuida en varios discos. Además, no es fácil encontrar soluciones en el mercado que permitan implementar el nivel RAID 3.

Ahora, existen dos niveles de arreglos el RAID nivel 0 + 1 y el RAID nivel 5 los cuales ofrecen la misma disponibilidad y casi el mismo desempeño, pero el costo de usar el nivel 0 + 1 es muy alto ya que por cada byte usado se requiere un byte de respaldo. Por lo tanto, nuestra implementación hará uso del arreglo de discos de nivel 5.

El nivel RAID 5 consume aproximadamente el 30% del espacio en disco para el almacenamiento de la redundancia de los datos, además la cantidad de discos que pueden ser adicionados a un arreglo de nivel 5 varia entre tres y ocho. Por tal motivo es importante determinar primero la cantidad de espacio útil que se

necesitará y por lo tanto la cantidad y tamaño de los discos que formaran parte del o de los arreglos de nivel 5 que se deben generar.

Por ejemplo, supongamos que instalaremos un cluster que sumando la cantidad de espacio en disco que necesita el sistema operativo, el propio software de cluster y las aplicaciones siendo ejecutadas en él; requiere un total de 50GB. de espacio útil; entonces ¿Cuantos arreglos deberán crearse? ¿Cuantos discos formarán parte de cada arreglo?.

Ya que contamos con discos de 9 GB. y el espacio útil que se necesita es de aproximadamente 50 GB., se crearán tres arreglos RAID de nivel 5 formados por tres discos cada uno. De esta forma obtendremos un espacio útil de 54 GB. Más adelante en el Tema 5.2.2 se describirá la distribución y tamaño de cada sistema de archivos a utilizar.

El subsistema de almacenamiento que estaremos usando cuenta con cuatro Shelfs[1] de discos, estos a su vez están interconectados a través de 6 canales independientes de I/O que son manejados por el controlador de discos HSG80. Además cada shelf es alimentado por dos fuentes de alimentación eléctrica independientes para efectos de soporte a fallas mediante el uso de redundancia.

Un factor importante a la hora de crear los arreglos de disco es la distribución de los mismos ya que de ella dependerá la obtención de un índice más alto de disponibilidad y desempeño.

Para ejemplificar la correcta instalación, configuración y creación de los arreglos RAID analizaremos los siguientes casos:

[1] Ver el glosario para una descripción más detallada

Conexión de los tres discos en un mismo shelf.

Si insertamos los 3 discos en un mismo shelf, como muestra la Figura 5.2.1-3, en caso de una falla eléctrica de las dos fuentes de alimentación del shelf todos los discos fallarán y por lo tanto todo el arreglo RAID se perderá.

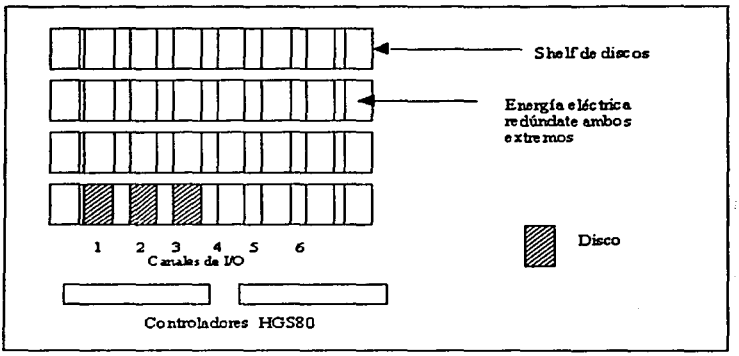


Figura 5.2.1-3 Conexión de los tres discos en un mismo shelf.

Conexión de los tres discos a un mismo canal de I/O.

Si todos los discos de un mismo arreglo son conectados a un solo canal de I/O, como lo muestra la Figura 5.2.1-4, en caso de una falla del canal todos los discos fallarán y por lo tanto todo el arreglo RAID se perderá ya que las operaciones de lectura / escritura no podrán ser efectuadas.

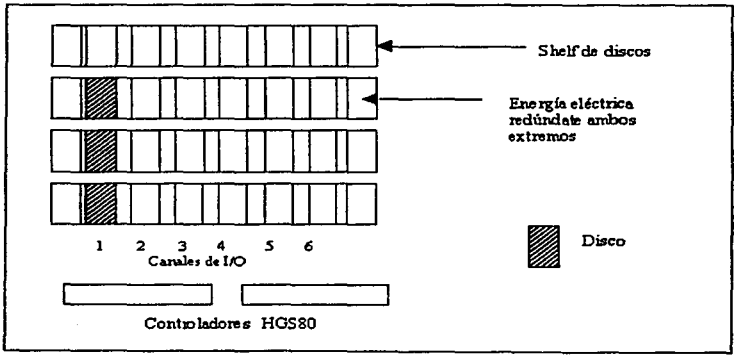


Figura 5.2.1-4 Conexión de los tres discos a un mismo canal de I/O.

Conexión optima.

La mejor forma de configurar los discos dentro del subsistema de almacenamiento es como lo muestra la Figura 5.2.1-5. Cada disco se insertará en un shelf diferente, además serán distribuidos en canales de I/O diferentes. Las ventajas de esta topología son las siguientes:

En caso de la falla de un shelf completo de discos, solo un disco fallará y por lo tanto la operación del arreglo de discos del cual forma parte ese disco podrá continuar. Una vez que el shelf sea reparado, el disco podrá ser reintegrado al arreglo.

En caso de la falla de un canal de I/O, la operación del arreglo no se verá afectada ya que solo un disco fallará y por lo tanto la operación del arreglo de discos del cual forma parte ese disco podrá continuar. Cuando el canal sea reparado, el disco podrá ser reintegrado al arreglo.

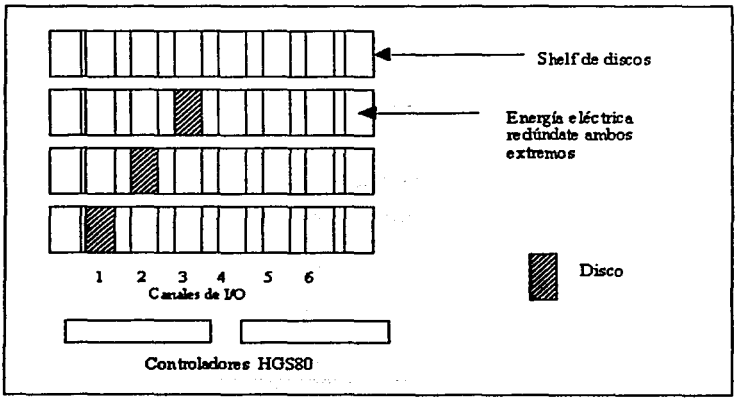


Figura 5.2.1-5 Conexión de los tres discos en diferente canal de I/O y en diferente shelf.

En resumen, si se insertan los discos de un mismo arreglo en diferente shelf y en diferente canal se maximiza la disponibilidad y el desempeño de cada arreglo RAID.

Red empresarial

Asumiendo que la infraestructura de red a la cual los nodos miembros del cluster van a ser conectados se encuentra correctamente diseñada (Ver recomendaciones hechas en el tema 4.2), nuestro interés en este momento es el de definir las direcciones IP que las interfaces conectadas a la red empresarial estarán usando.

Se requiere disponer de al menos tres direcciones IP, mismas que serán usadas de la siguiente forma:

- Una asociada al nombre del cluster, también conocida como la dirección asociada al alias del cluster
- La segunda y tercera serán usadas como la dirección principal de cada uno de los dos nodos del cluster

La Figura 5.2.1-6 muestra la asignación de las tres direcciones mínimas necesarias para conectar un cluster a una red empresarial.

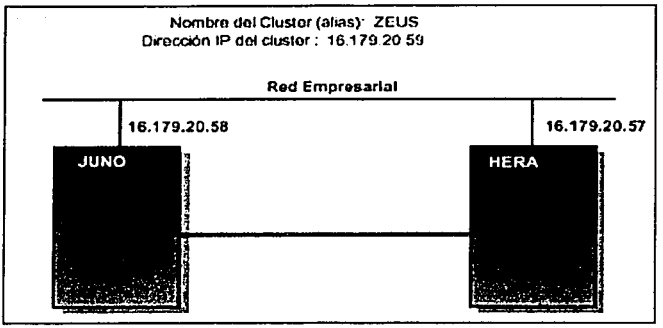


Figura 5.2.1-6 Asignación de las direcciones IP de la red empresarial del cluster.

En nuestro caso usaremos los nombres y direcciones IP que se describen en la Tabla 5.2.1-3

Nodo	Nombre	Dirección IP
0	ZEUS	16.179.20.59
1	HERA	16.179.20.57
2	JUNO	16.179.20.58

Tabla 5.2.1-3 Asignación de los nombres de las direcciones IP del cluster del caso de estudio.

Cluster Interconnect

Como mencionamos en temas anteriores el cluster interconnect se puede implementar de dos formas:

- Haciendo uso de una interfaz con tecnología propietaria llamada memory channel
- Usando una interfaz de red ethernet standard

En este caso, y con el objetivo de manejar componentes estándar usaremos interfaces ethernet 100 Base TX como cluster interconnect. Por lo tanto se necesitan los siguientes componentes mínimos:

- 2 Tarjetas Ethernet 100Base-TX Ethernet (RJ45)
- 1 Cable con configuración cross over (Cable cruzado) Ethernet 100Base TX (RJ45)

Para que la comunicación privada de los dos nodos no se vea interrumpida, es necesario contar con redundancia del cluster interconnect, entonces el material necesario para la implementación del cluster será el siguiente:

- 4 Tarjetas Ethernet 100Base-T Ethernet (RJ45)
- 2 Cables con la configuración cross over (Cable cruzado) Ethernet 100Base TX (RJ45)

La Figura 5.2.1-7 muestra la conexión del cluster interconnect usando redundancia.

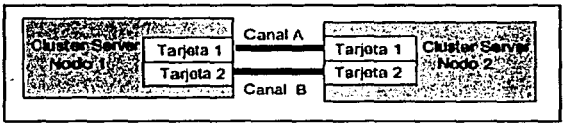


Figura 5.2.1-7 Conexión de los componentes usando redundancia

5.2.2 Planeación de la configuración de almacenamiento de datos

Antes de empezar la instalación del cluster necesitamos contar con los siguientes espacios en disco:

- Uno o más discos para almacenar el sistema operativo. Para lograr alta disponibilidad estos discos deberán estar instalados en el bus compartido del cluster
- Uno o más discos del bus compartido para guardar los sistemas de archivos comunes a todos los miembros del cluster:
  - root / cluster_root
  - usr /usr cluster_usr
  - var /var cluster_var
- Un disco del bus compartido que será usado como disco de arranque por cada miembro del cluster
- Dependiendo del número de miembros del cluster, un disco del bus compartido que funcionara como disco de quórum
- Los n discos del bus compartido que la aplicación altamente disponible requiera

Usar discos separados para cada una de estos sistemas de archivos nos ayudará a proporcionar mejor desempeño (Más I/O en paralelo.). Ésto también provee la oportunidad de balancear la carga de trabajo.

Cuando hablamos de un disco no necesariamente nos referimos a un disco físico, ya que el controlador de discos que estamos usando nos permite crear particiones a partir de un arreglo RAID, mismas que son mostradas al sistema operativo como discos independientes. Esto nos permitirá ahorrar discos y espacio en ellos.

Durante la instalación del sistema operativo, es importante indicar que el tipo de sistemas de archivos que estaremos usando será advanced file system (AdvFS.).

Durante la creación del cluster, se debe proveer el nombre del disco, y partición que contendrán los sistemas de archivos comunes a todos los miembros del cluster (Conocidos como cluster wide file systems /, /usr y /var). Cada sistema de archivos AdvFS deberá ser creado en una partición separada, sin embargo, las particiones no tienen que estar en el mismo disco. Por ejemplo:

- dk1b cluster_root#root
- dsk2c cluster_usr#usr
- dsk3c cluster_var#var

Si alguna partición de un disco es usada por algún sistema de archivos de tipo cluster wide (/, /usr, /var), ese mismo disco no podrá ser usado como disco de arranque de algún miembro del cluster o como disco de quórum.

Discos de arranque

Como ya se ha mencionado, cada miembro tiene su propio disco de arranque, el cual comúnmente se usa para almacenar las siguientes áreas del sistema:

- Partición a, para el sistema de arranque del servidor
- Partición b, para el área de swap
- Partición h, para almacenar información sobre el status del cluster

Disco de quórum

La función del disco de quórum es la de mantener el cluster siempre en operación, esto se logra por medio del uso de votos, mismos que son asignados por el usuario durante la instalación del cluster. El funcionamiento del disco quórum es un tanto simple y solo es recomendable para clusters de dos miembros ya que

para clusters de más de dos miembros los votos se repartirían entre los demás miembros del cluster permitiendo que siempre hubiera suficientes votos para que en caso de alguna falla la operación pudiera continuar.

Los votos asignados a este disco son contados cuando se calcula el quorum. De está manera, el disco de quorum ayuda a prevenir particiones del cluster. Para un cluster de dos miembros, recomendamos que se configure un disco de quorum. Si cada miembro y el disco tienen un voto, el cluster podra mantener el quorum y continuar operando siempre que dos votos esten presentes. Esto es, el cluster puede continuar operando cuando sus dos miembros esten arriba o cuando un miembro está arriba y el disco de quorum está disponible.

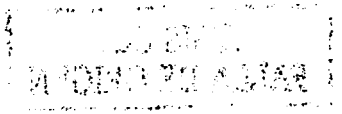
Un cluster solo puede tener un quorum disk y el mismo no puede ser usado para más nada que eso por tal motivo se recomienda que este disco sea muy pequeño.

Configuración mínima para un cluster de dos nodos

Para un cluster de dos miembros es necesario contar con un mínimo de cuatro discos (Aunque cinco discos sería lo más recomendable.).

- Un disco para el sistema operativo
- Un disco para los sistemas de archivos del cluster /, /usr, /var
- Dos discos de arranque, uno para cada miembro
- Un disco para ser usado como disco de quórum

La Figura 5.2.2-1 muestra la relación entre los sistemas de archivos que cada disco contiene y la estructura de directorios del cluster. El disco de quorum no se muestra debido a que el mismo no contiene un file system.



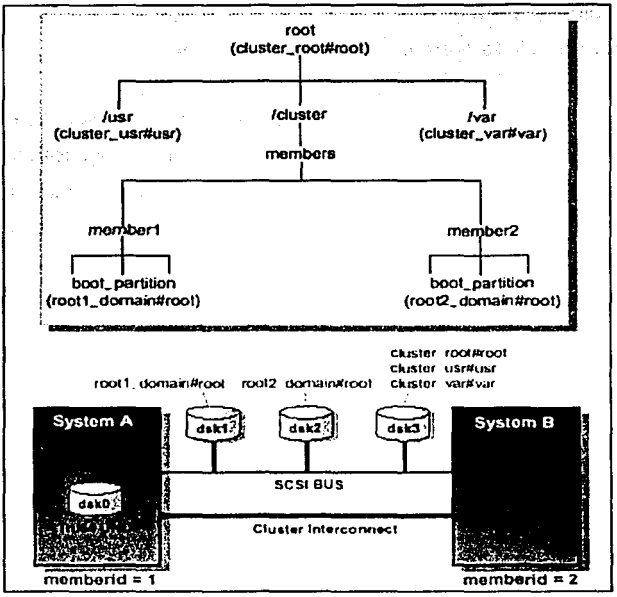


Figura 5.2.2-1 Relación entre los discos y sistemas de archivos.

Espacio en disco recomendado

El espacio recomendado ha sido determinado basándonos en la práctica ya que en el mundo real hay factores que se deben tomar en cuenta para prever futuras fallas en el sistema.

El sistema de archivos de root (/) es el que guarda la configuración del sistema operativo, por lo que en la práctica es recomendable que su tamaño sea de 1GB con el objetivo de permitir cambios futuros en el sistema.

TESIS CON
FALLA DE ORIGEN

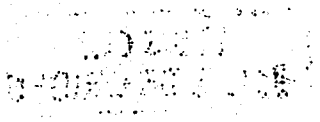
Se recomienda un tamaño de 3GB para el sisema de archivos de /usr ya que en él se almacenará el espacio en disco generado por todos los usuarios del sistema.

El tamaño del sistema de archivos de /var depende de la cantidad de memoria RAM que tenga el sistema ya que en él se almacena el archivo de DUMP (vaciado completo de memoria) que el sistema genera cuando se cae. Este archivo es muy importante, ya que ayuda a diagnosticar la falla que ocasionó la caída. De ahí la importancia de que el tamaño de esta partición sea por lo menos cuatro veces el tamaño de la memoria RAM.

Al igual que el sistema de archivos de root, se recomienda que el tamaño mínimo de los sistemas de archivos de arranque sea de 1GB con el objetivo de prever futuros cambios. Por ejemplo, cuando se re-compila un kernel nuevo para cambiar parámetros del sistema en un momento dado se debe tener suficiente espacio para almacenar el kernel anterior y el nuevo hasta estar seguros de la funcionalidad del nuevo.

Por último, el tamaño del disco de quórum será de 1MB ya que como mencionamos, en este disco no se almacenará ningún sistema de archivos y por lo tanto no es necesario más espacio.

La Tabla 5.2.2-1 muestra el espacio mínimo y recomendado para cada partición.



Sistema de archivos	Partición	Tamaño mínimo	Tamaño recomendado
cluster root / (AdvFS)	b	200 MB	1 GB
cluster /usr (AdvFS)	g	1000 MB	3 GB
cluster /var (AdvFS)	h	1000 MB	3 GB
Disco de arranque miembro 1	a	128 MB	4 GB
Disco de arranque miembro 2	a	128 MB	4 GB
Disco de quorum	h	1 MB	1 MB

Tabla 5.2.2-1 Tamaños recomendados para cada area del cluster

5.2.3 Planeación de la configuración del software

Los pasos necesarios para formar un cluster versión 5.x son:

1. La configuración de hardware y de almacenamiento de datos se debe tener lista y terminada antes de empezar la instalación del software.

2. Se debe recordar que el sistema operativo solo se instalará por primera y única vez en el primer miembro del cluster. No es necesario instalar el sistema en cada uno de los miembros, ya que el procedimiento de adición de un miembro se encarga de propagarlo a los nuevos miembros.

3. Configurar el sistema operativo incluyendo los servicios de red y de tiempo, cargar y configurar las aplicaciones que se planean usar en el cluster.

4. Cargar las licencias y el producto de cluster.

5. Ejecutar el comando **clu_create** para crear el disco de arranque del primer sistema miembro del cluster y a la vez popular los sistemas de archivos comunes de **/ root, /usr** y **/var**.

6. Una vez creado el disco de arranque, levantar el primer miembro a apartir de su disco de arranque. Cuando el sistema termina de levantar se crea un cluster de un miembro y se montan los sistemas de archivos comunes (root /, /usr, /var).

7. Accesar el primer miembro del cluster con la cuenta de root y ejecutar el comando clu_add_member para agregar el segundo miembro al cluster. Se debe dar reboot a cada nuevo miembro antes de adicionar el siguiente.

Actividades de preparación

Antes de arrancar la instalación del software del cluster es importante realizar las siguientes actividades:

- Decidir los IDs que serán asignados a los miembros del cluster
- Obtener los nombres y direcciones IP necesarios
- Decidir que discos y que particiones usarán para instalar el sistema operativo y el software del cluster
- Decidir cuantos votos se asignarán a cada miembros del cluster y en caso de ser configurado, al disco de quorum

Identificadores de los miembros del cluster

Cada miembro dentro de un cluster tiene un identificador único, cuyo valor es un entero entre uno y sesenta y tres. El software de cluster los utiliza para identificar cada miembro del cluster. Durante la instalación se podrá determinar el valor del identificador de cada miembro.

Por ejemplo, en nuestro cluster de dos nodos llamados Zeus estaremos usando los siguientes identificadores:

Hera	Identificador de miembro 1
Juno	Identificador de miembro 2

Nombres y direcciones IP

Como hemos comentado anteriormente, se necesitan tres direcciones IP dentro de la red empresarial, dos para los miembros del cluster y una o más para el o los alias del cluster. Además, también son necesarias dos direcciones que serán usadas por las interfaces del cluster interconect, en este caso se recomienda usar direcciones dentro del espacio de direcciones privadas.

La Figura 5.2.3-1 muestra el diagrama de un cluster de dos miembros, sus nombres, alias y direcciones IP respectivas a cada uno de ellos.

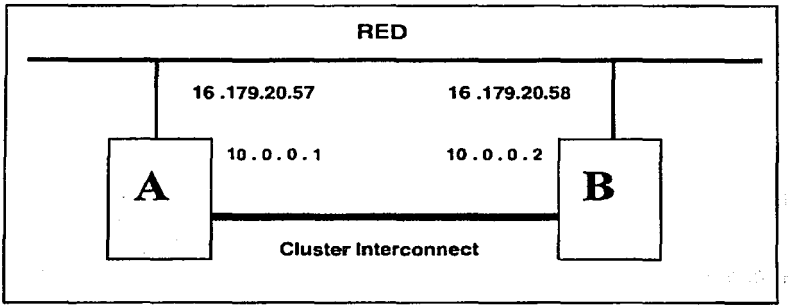


Figura 5.2.3-1 Manejo de direcciones IP en un cluster.

5.3

IMPLEMENTACIÓN

5.3.1 Red Empresarial

Al momento de crear el cluster y adicionar los miembros del mismo a la red corporativa, necesitaremos los siguientes nombres y direcciones IP:

- Un nombre y dirección para el cluster, mismos que serán identificados como el nombre y la dirección de default del "alias del cluster" (Dirección virtual)
- Para cada miembro del cluster un nombre y una dirección IP por cada interfaz externa de red

Cada cluster deberá tener un nombre, el cual debe ser diferente al nombre de cualquiera de los miembros del mismo y podrá ser usado para hacer referencia al cluster por completo, en nuestro caso este mismo nombre será asociado al default del "alias del cluster". Además, una dirección IP asignada también a todo el cluster. Esta dirección proveerá el método a través del cual los clientes externos

tendrán acceso a los servicios proporcionados por el cluster en lugar de acceder de manera directa a los nodos miembro del mismo. De esta forma, la caída de uno de los miembros será transparente para los usuarios ya que ellos harán referencia siempre a la dirección del alias del cluster (Dirección virtual) y no a la de los miembros del mismo.

Esta dirección que como ya mencionamos estará asociada al default del alias del cluster, debe ser una dirección válida dentro de la red corporativa para que de esta manera los clientes puedan tener acceso al cluster. No debe ser una dirección "broadcast" y no puede residir en la misma subred donde reside la dirección asignada al cluster interconnect.

Cada miembro del cluster deberá tener un nombre. Por convención, este nombre usualmente se asocia al nombre y a la dirección IP previamente configurados y asignados a la interfaz primaria del servidor.

La Tabla 5.3.1-1 Muestra las direcciones que usaremos en nuestro caso.

Descripción	Nombre	Dirección IP
Cluster	Zeus	16.179.20.59
Miembro 1	Hera	16.179.20.57
Miembro 2	Juno	16.179.20.58

Tabla 5.3.1-1 Asignación de los nombres de las direcciones IP del cluster del caso de estudio.

La interfaz primaria del primer servidor se deberá configurar antes de la creación del cluster, proveyendo la información necesaria para tal fin a través del comando netsetup. La interfaz primaria de los demás servidores así como la dirección IP de alias del cluster serán configuradas automáticamente al momento de crear el cluster usando el comando clu_create.

Ejemplo (Configuración de la interfaz primaria del servidor hera):

```
hera># netsetup

**** MAIN MENU ****
1 Configure Network Interfaces
2 Enable/Disable Network Daemons and Add Static Routes
3 Add/Delete Host Information
4 Display Network Configuration
5 Exit

Enter the number for your choice: 1

***** CONFIGURE/DELETE NETWORK INTERFACES *****
You can configure or delete network interfaces. Configuration information
is updated in /etc/rc.config and /etc/hosts. Choose "configure" or
"delete" at the prompt.
Enter whether you want to "(c)onfigure" or "(d)elete" network interfaces.
If you are finished, press the RETURN key: c

You want to "configure" interfaces. Is this correct [yes]? Return

You will now be asked a series of questions about the system.
Default answers are shown in square brackets ([]). To use a
default answer, press the RETURN key.
This machine contains the following network interfaces:
ee0
tu0
sl0
Which interface do you want to configure [ee0]: Return

You want to configure "ee0". Is this correct [yes]? Return
Enter the hostname for the system []: hera
The hostname for the system is "hera".
Is this correct [yes]? Return
Enter the Internet Protocol (IP) address for interface "ee0"
in dot notation []: 16.179.20.57
The IP address for interface "ee0" is "16.179.20.57".
Is this correct [yes]? Return
Subnetworks allow the systems on a local area network to be on different
physical networks. For the following question, use the default answer
unless the existing local area network is using subnet routing.
If the local area network is using subnet routing, you need to know
the subnet mask.
Enter the subnet mask in dot notation [255.0.0.0]: 255.255.252.0
The subnet mask for "ee0" is "255.255.252.0".
Is this correct [yes]? Return
For the following question USE THE DEFAULT ANSWER unless you would like
to add additional flags (found in the ifconfig reference page) to the
ifconfig command. Normally, you will USE THE DEFAULT ANSWER.
Do you want to use additional ifconfig flags for this interface [no]?
Return
The configuration looks like:

system hostname: "hera"
```

```
ifconfig ee0 16.179.20.57 netmask 255.255.252.0
Is this correct [yes]? Return
***** UPDATING /etc/rc.config *****
"ee0" is configured in /etc/rc.config
***** UPDATING /etc/hosts *****
"16.179.20.57 hera" is configured in /etc/hosts
Do you want to configure another network interface [yes]? no
Enter whether you want to "(c)onfigure" or "(d)elete" network interfaces.
If you are finished, press the RETURN key: Return

**** MAIN MENU ****
1 Configure Network Interfaces
2 Enable/Disable Network Daemons and Add Static Routes
3 Add/Delete Host Information
4 Display Network Configuration
5 Exit
Enter the number for your choice: 2
```

PARA ACTIVAR EL PROTOCOLO DE RUTEO GATED

```
***** ENABLE/DISABLE NETWORK DAEMONS AND ADD STATIC ROUTES *****
You can choose whether you want to enable or disable rwhod (rwho
daemon), and either gated (gateway daemon) or routed (route daemon).
You can also configure static routes. Daemons and static route commands
are executed when the network is started up.
*********************************************************
* Note that the OLD version of gated will be set up. *
* You must run netconfig to set up the NEW version *
* of gated. *
*********************************************************
Do you want to run rwhod [no]? Return
rwhod is disabled.
You can run either gated or routed but not both.
Choose "(g)ated" or "(r)outed" at the prompt.
*********************************************************
Do you want to run "(g)ated" or "(r)outed.
If you do not want either, then press the RETURN key: g
```

EL CLUSTER REQUIERE LA ACTIVACION DEL PROTOCOLO DE RUTEO GATED

```
You want to run gated. Is this correct [yes]? Return
You can use flags (see the gated reference page) when you run
the gated daemon.
The gated flag is "-q". Is this correct [yes]? Return
gated is enabled.
Static route commands can be configured in /etc/routes that will be
executed when the network is started up on this system.
Do you want to add a static route [no]? Return

**** MAIN MENU ****
1 Configure Network Interfaces
2 Enable/Disable Network Daemons and Add Static Routes
3 Add/Delete Host Information
4 Display Network Configuration
5 Exit
Enter the number for your choice: 5
```

```
To perform additional tasks in setting up the network, see the Network
Configuration reference. For the netsetup modifications to take
effect, either restart the network services on this system with the
following
command:
/usr/sbin/rcinet restart
or reboot this system with the following command:
/usr/sbin/shutdown -r now
NOTE: If you are going to use the '/usr/sbin/rcinet restart' command,
warn the users that the network services on this system are going
to be restarted. Also, any NFS filesystems not mounted via fstab or
automount will not be remounted.
Do you want netsetup to automatically restart the network
services on this system [no]? yes          PARA ACTIVAR LA CONFIGURACION
                                           RECIENTEMENTE INTRODUCIDA

Stopping Internet services on this system. Please wait...
Unmounting NFS filesystems
Internet services on this system are stopped.
Starting Internet services on this system. Please wait...
Configuring network
hostname: rdunge
gated daemon started
Setting kernel timezone variable
Mounting NFS filesystems
3 aliases, longest (MAILER-DAEMON) 19 bytes, 53 bytes total
SMTP Mail Service started
Extensible SNMP master agent started
Base O/S sub-agent started
Internet services provided.
Internet services on this system are started.
***** NETWORK SETUP COMPLETE *****
```

Una vez configurada la interfaz, es posible revisar el correcto funcionamiento de la misma usando el comando "ifconfig":

```
hera># ifconfig –i ee0
ee0:
flags=c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX>
    inet 16.179.20.57 netmask fffffc00 broadcast 16.179.23.255 ipmtu 1500
```

Es importante recordar que para aumentar la disponibilidad de la red corporativa para con el cluster, se recomienda implementar alguna solución ya sea de hardware o de software que permita tener redundancia en ese sentido. En esta implementación estaremos usando NetRAIN que es una solución de software que

provee alta disponibilidad de adaptadores múltiples de red en caso de una falla (Ver capítulo 4.).

En caso de elegir NetRAIN como solución para proveer redundancia de las interfaces de red, la siguiente configuración NetRAIN deberá ser implementada en ambos miembros del cluster:

1.- **# rcmgr set NRDEV_0 nr0**
2.- **# rcmgr set NRCONFIG_0 ee0,ee1**
3.- **# rcmgr set NR_DEVICES 1**
4.- **# rcmgr set NETDEV_0 nr0**
5.- **# rcmgr set IFCONFIG_0 16.179.20.57 netmask 255.255.252.0**
6.- **# rcmgr set NUM_NETCONFIG 1**

1.- Creamos un objeto NetRAIN llamado nr0
2.- Indicamos que la nueva interfaz nr0 esta formada por las interfaces físicas ee0 y ee1
3.- Le indicamos al sistema que existe una interfaz de tipo NetRAIN
4.- Creamos una interfaz de red llamada nr0 para el objeto virtual NetRAIN
5.- Definimos la dirección IP y la mascara de red para la nueva interfaz NetRAIN
6.- Le indicamos al sistema la existencia de solo una interfaz de red

5.3.2 Red Privada (Cluster Interconnect)

Como ya mencionamos en capítulos anteriores, un cluster requiere de la implementación de una red dedicada; la cual deberá ser una subred físicamente separada de la red corporativa y en la cual solamente los miembros del cluster podrán residir. Esta red tomará el lugar primario en el cluster y todos los miembros deberán estar conectados a ella.

Se necesita un nombre y una dirección IP para la interfaz de interconexión de cada sistema miembro. Aunque cada sistema miembro puede tener interfaces de interconexión redundantes para propósitos de tolerancia a fallas, las mismas compartirán la misma dirección de red.

Ya que esta red es privada para todos los miembros del cluster, se recomienda usar direcciones IP dentro de espacios de direcciones definidos como privados. Por ejemplo la red 10.0.0 . La Tabla 5.3.2-1 muestra las direcciones que se usarán para el cluster interconnect.

Descripción	Nombre	Dirección IP
Cluster Interconnect Juno	juno-ics0	10.0.0.2
Cluster Interconnect Hera	hera-ics0	10.0.0.1

Tabla 5.3.2-1 Direcciones IP para del Cluster Interconnect.

Como se mencionó anteriormente, las características (IP, netmask, nombre, etc.) de las interfaces de la red privada son configuradas al momento de crear el cluster usando el comando clu_create (Ver Apéndice A)

Usando el comando ifconfig es posible observar las características de configuración de las interaces de red tanto corporativa (eex) como privada (tux) de cada uno de los servidores miembros del cluster:

```
hera># ifconfig -a
ee0: flags=c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX>
   inet 16.179.20.57 netmask fffffc00 broadcast 16.179.23.255 ipmtu 1500

lo0: flags=100c89<UP,LOOPBACK,NOARP,MULTICAST,SIMPLEX,NOCHECKSUM>
   inet 127.0.0.1 netmask ff000000 ipmtu 4096

sl0: flags=10<POINTOPOINT>

tu0:
flags=1000c63<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST,SIMPLEX,
CLUIF>
   inet 10.0.0.1 netmask ffffff00 broadcast 10.0.0.255 ipmtu 1500

tun0: flags=80<NOARP>
```

5.3.3 Bus de Almacenamiento Compartido

La implementación del bus de almacenamiento compartido del cluster en el nivel de hardware no es más que la conexión física de cada uno de los elementos que lo forman (Ver Tabla 5.2.1-1.). Para ello, construiremos la topología descrita por la Figura 5.2.1-2 haciendo uso de ocho cables de fibra óptica de 5 m cada uno.

Asumiendo que en la SAN que estamos construyendo solamente se encuentran conectados servidores bajo plataforma UNIX (SAN homogénea) y a que por esta razón no es necesario configurar zonas, no haremos uso de la capacidad de zoneo (Zonning) que los switches tienen. De esta manera y con fines de administración, solamente es necesario configurar la dirección IP de cada uno de ellos haciendo uso de una conexión tipo serial. A partir de esto, se podrá tener acceso vía telnet a la consola de los mismos.

El comando a ejecutar para tal efecto desde el prompt de consola del switch es:

Switch >> ipAddrSet — El cual interactivamente nos permitirá configurar los siguientes parámetros Ethernet IP Address, Ethernet Subnetmask, Gateway Address

Una vez conectados los elementos del bus, se debe estandarizar el modo de operación FABRIC en todos ellos haciendo lo siguiente:

Adaptadores FC SCSI de los servidores:

Ejecutar el siguiente comando en el prompt de consola de cada uno de los equipos.

P00 >>> wwidmgr -set adapter –item 9999 –topo FABRIC

Controladores de disco HSG80:

Ejecutar el siguiente comando.

HSG80 >> set this port_1_topology=FABRIC Puerto 1 del Controlador A

HSG80 >> set this port_2_topology=FABRIC Puerto 2 del Controlador A

HSG80 >> set other port_1_topology=FABRIC Puerto 1 del Controlador B
HSG80 >> set other port_2_topology=FABRIC Puerto 2 del Controlador B

Cuando los controladores de disco detectan los adaptadores SCSI a través de las conexiones o rutas establecidas por la topología construida, entonces crean una conexión por cada una de esas rutas que tiene acceso a cada adaptador (8 rutas, ya que son 4 puertos del controlador X 2 adaptadores de cada servidor.).

Cada una de ellas puede ser identificada fácilmente relacionando el wwid de los adaptadores con la salida del comando "show connections" en el nivel de los controladores.

Es recomendable renombrar las conexiones con el objetivo de relacionarlas fácilmente con cada uno de los equipos. Esto se puede hacer de la siguiente forma:

HSG80 >> rename ¡newcon01 hea1_cap1
HSG80 >> rename ¡newcon02 hea1_cbp2
HSG80 >> rename ¡newcon03 hea2_cap2
HSG80 >> rename ¡newcon04 hea2_cbp1
HSG80 >> rename ¡newcon05 jua1_cap1
HSG80 >> rename ¡newcon06 jua1_cbp2
HSG80 >> rename ¡newcon07 jua2_cap2
HSG80 >> rename ¡newcon08 jua2_cbp1

Donde la convención seguida para generar estos nombres es la siguiente:

- 2 primeras letras para el nombre del servidor (Hera=he, Juno=ju)
- 2 letras para el número del adaptador dentro del servidor (adaptador1=a1, adaptador2=a2)
- dos letras para el nombre del controlador (ControladorA= ca, ControladorB=cb)
- dos letras para el número de puerto del controlador (Puerto1=p1, Puerto2=p2

Quedando las conexiones de la siguiente manera:

```
HSG80 >>show connections
Connection                                        Unit
  Name    Operating system   Controller Port   Address   Status  FOCET
HEA1_CAP2    TRU64_UNIX        THIS     2     011100  OL this    0
       HOST_ID=2000-0000-C922-2190       ADAPTER_ID=1000-0000-C922-2190

HEA1_CBP1    TRU64_UNIX        OTHER    1     011100  OL other   0
       HOST_ID=2000-0000-C922-2190       ADAPTER_ID=1000-0000-C922-2190

HEA2_CAP1    TRU64_UNIX        THIS     1     011100  OL this    0
       HOST_ID=2000-0000-C922-2186       ADAPTER_ID=1000-0000-C922-2186

HEA2_CBP2    TRU64_UNIX        OTHER    2     011100  OL other   0
       HOST_ID=2000-0000-C922-2186       ADAPTER_ID=1000-0000-C922-2186

JUA1_CAP2    TRU64_UNIX        THIS     2     011000  OL this    0
       HOST_ID=2000-0000-C922-0622       ADAPTER_ID=1000-0000-C922-0622

JUA1_CBP1    TRU64_UNIX        OTHER    1     011000  OL other   0
       HOST_ID=2000-0000-C922-0622       ADAPTER_ID=1000-0000-C922-0622

JUA2_CAP1    TRU64_UNIX        THIS     1     011000  OL this    0
       HOST_ID=2000-0000-C922-07AE       ADAPTER_ID=1000-0000-C922-07AE

JUA2_CBP2    TRU64_UNIX        OTHER    2     011000  OL other   0
       HOST_ID=2000-0000-C922-07AE       ADAPTER_ID=1000-0000-C922-07AE
```

Concluidas las actividades de configuración del bus de almacenamiento, podemos entonces realizar la configuración de los espacios de disco necesarios para la instalación del cluster. El siguiente tema explica a detalle los pasos a seguir para llevara a cabo tal configuración.

5.3.4 Almacenamiento de datos

5.3.4.1 Tecnología RAID

Una vez realizado el análisis del nivel RAID a utilizar, se procederá con la implementación y configuración del sistema de almacenamiento de la siguiente manera.

Espacio en Disco

Tomando como base los espacios recomendados en Tabla 5.2.2-1 para cada partición, tenemos, que se utilizaran 16 GB de espacio para el sistema operativo y de acuerdo con la planeación de la capacidad actual y futura que el servidor web y la Base de Datos requerirán (35 GB), el espacio útil total necesario es de 51GB, aproximadamente.

Cantidad de discos y arreglos

Tomando en cuenta que disponemos de discos de 9GB. de capacidad, será necesario construir tres arreglos RAID para cubrir la cantidad de espacio en disco que se requiere. Por lo tanto los arreglos serán construidos de la siguiente manera.

Discos por arreglo	Espacio Total	Espacio consumido por redundancia	Espacio Útil
3	27	9	18

Si se utilizan 3 discos por arreglo obtendremos de espacio útil real de 18 GB., por lo tanto, utilizaremos un total de 9 discos formando 3 arreglos RAID de nivel 5 para obtener 54GB. de espacio útil real.

Organización de los Discos dentro del subsistema de Almacenamiento

La Figura 5.3.4-1 muestra la forma como serán insertados los discos en los shelfs de almacenamiento.

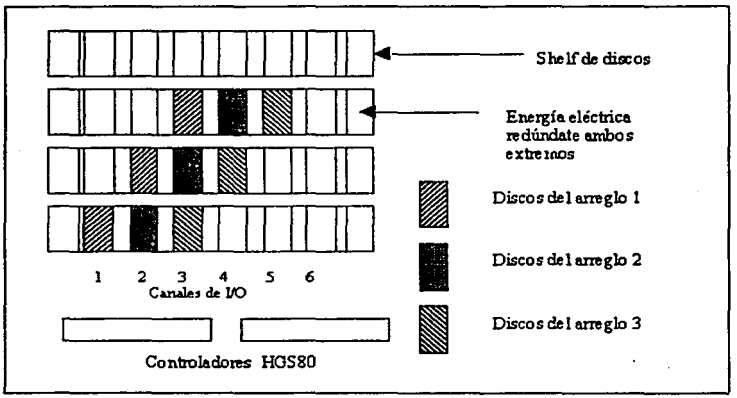


Figura 5.3.4-1 Conexión de los discos en los shelfs

Para que el controlador de discos HSG80, reconozca los discos insertados y les asigne un nombre lógico, será necesario conectar una terminal al puerto serial del mismo y ejecutar el comando RUN CONFIG

Sintaxis

```
HSG80>>RUN CONFIG
```

La Tabla 5.3.4-1 lista los nombres lógicos de cada uno de los discos, después de haber ejecutado el comando anteriormente descrito.

Arreglo al que pertenecerá	Nombre del disco
1	DISK10000
1	DISK20100
1	DISK30200
2	DISK20000
2	DISK30100
2	DISK40200
3	DISK30000
3	DISK40100
3	DISK50200

Tabla 5.3.4-1 Nombres lógicos de cada uno de los discos.

Configuración de los arreglos RAID de discos.

Hasta este momento, el controlador no reconoce como están organizados los arreglos de disco, así que, utilizando la misma terminal procederemos a ejecutar los comandos necesarios para crear cada uno de ellos.

Los pasos a seguir para la configuración de cada arreglo son:

- Crear cada uno de los arreglos
- Inicializar cada uno de ellos
- A partir de un arreglo crear las particiones necesarias indicando su tamaño
- Crear una unidad para cada partición que necesite ser vista por el sistema operativo
- Configurar los parámetros de cada unidad

Los comandos de configuración a utilizar se listan a continuación de acuerdo al orden antes mencionado.

```
ADD RAIDSET  Nombre_Arreglo DISCO1 DISCO2 DISCO3  [DISCO 8]
INITIALIZE Nombre_Arreglo
CREATE_PARTITION Nombre_Arreglo SIZE = Tamaño en porcentaje
ADD UNIT  Nombre_Unidad Nombre_Arreglo
SET MAXIMUM_CACHE_TRANSFER_SIZE = 1024 (Para mayor rendimiento)
SET ENABLE_ACCESS_PATH = (Conexiones de Hera y Juno)
SET  PREFERRED_PATH  =  [OTHER_CONTROLLER]   [THIS_CONTROLLER]
(Con el objetivo de balancear la carga de trabajo)
SET IDENTIFIER = n
```

La secuencia de comandos para crear los tres arreglos RAID de nivel 5 R100, R200 y R300 son:

```
HSG80>> ADD RAIDSET R100 DISK10000  DISK20100  DISK30200
HSG80>> ADD RAIDSET R200 DISK20000  DISK30100  DISK40200
HSG80>> ADD RAIDSET R300 DISK30000  DISK40100  DISK50200
```

Inicializando los arreglos.

```
HSG80>> INITIALIZE R100
HSG80>> INITIALIZE R200
HSG80>> INITIALIZE R300
```

Ahora creamos las particiones.

```
HSG80>> CREATE_PARTITION R100 SIZE = 40
HSG80>> CREATE_PARTITION R100 SIZE = 10
HSG80>> CREATE_PARTITION R100 SIZE = 10
HSG80>> CREATE_PARTITION R100 SIZE = LARGEST
```

```
HSG80>> CREATE_PARTITION R200 SIZE = 20
HSG80>> CREATE_PARTITION R200 SIZE = 20
HSG80>> CREATE_PARTITION R200 SIZE = LARGEST
```

Creamos las unidades de cada una de las particiones del arreglo R100:

```
HSG80>>ADD UNIT D1 R100 IDENTIFIER= 1 PARTITION = 1
HSG80>>ADD UNIT D2 R100 IDENTIFIER= 2 PARTITION = 2
HSG80>>ADD UNIT D3 R100 IDENTIFIER= 3 PARTITION = 3
HSG80>>ADD UNIT D4 R100 IDENTIFIER= 4 PARTITION = 4
```

Asignamos los parámetros de configuración para cada una de las unidades creadas:

```
HSG80>>SET D1 MAXIMUM_CACHE_TRANSFER_SIZE = 1024
HSG80>>SET D1 ENABLE_ACCESS_PATH= (JUNO,HERA)
HSG80>>SET D1 PREFERRED_PATH = THIS_CONTROLLER

HSG80>>SET D2 MAXIMUM_CACHE_TRANSFER_SIZE = 1024
HSG80>>SET D2 ENABLE_ACCESS_PATH= (JUNO,HERA)
HSG80>>SET D2 PREFERRED_PATH = THIS_CONTROLLER

HSG80>>SET D3 MAXIMUM_CACHE_TRANSFER_SIZE = 1024
HSG80>>SET D3 ENABLE_ACCESS_PATH= (JUNO,HERA)
HSG80>>SET D3 PREFERRED_PATH = THIS_CONTROLLER

HSG80>>SET D4 MAXIMUM_CACHE_TRANSFER_SIZE = 1024
HSG80>>SET D4 ENABLE_ACCESS_PATH= (JUNO,HERA)
HSG80>>SET D4 PREFERRED_PATH = THIS_CONTROLLER
```

Hasta este punto, solo se han definido las particiones del arreglo R100 que servirán para instalar el sistema operativo del cluster. Los arreglos R200 y R300 serán utilizados al 100% por las aplicaciones configuradas dentro del cluster.

Creamos las unidades de cada una de las particiones del arreglo R200:

```
HSG80>>ADD UNIT D5 R200 IDENTIFIER= 5 PARTITION = 1
HSG80>>ADD UNIT D6 R200 IDENTIFIER= 6 PARTITION = 2
HSG80>>ADD UNIT D7 R200 IDENTIFIER= 7 PARTITION = 3
```

Asignamos los parámetros de configuración para cada una de las unidades creadas:

```
HSG80>>SET D5 MAXIMUM_CACHE_TRANSFER_SIZE = 1024
HSG80>>SET D5 ENABLE_ACCESS_PATH= (JUNO,HERA)
HSG80>>SET D5 PREFERRED_PATH = OTHER_CONTROLLER

HSG80>>SET D6 MAXIMUM_CACHE_TRANSFER_SIZE = 1024
HSG80>>SET D6 ENABLE_ACCESS_PATH= (JUNO,HERA)
HSG80>>SET D6 PREFERRED_PATH = OTHER_CONTROLLER

HSG80>>SET D7 MAXIMUM_CACHE_TRANSFER_SIZE = 1024
HSG80>>SET D7 ENABLE_ACCESS_PATH= (JUNO,HERA)
HSG80>>SET D7 PREFERRED_PATH = OTHER_CONTROLLER
```

Creamos la unidad y se asignan los parámetros de configuración para el arreglo R300:

```
HSG80>>ADD UNIT D8 R300 IDENTIFIER= 8
```

```
HSG80>>SET D8 MAXIMUM_CACHE_TRANSFER_SIZE = 1024
HSG80>>SET D8 ENABLE_ACCESS_PATH= (JUNO,HERA)
HSG80>>SET D8 PREFERRED_PATH = OTHER_CONTROLLER
```

Con esto, habremos creado ocho discos, mismos que podrán ser accesados por cada uno de los servidores conectados al bus de almacenamiento.

La Tabla 5.3.4-2 describe el uso que se le dará a cada uno de las unidades previamente creadas.

Unidades	Utilidad
D1	Sistema de archivos de cluster (root, usr,var)
D2	Disco de arranque del miembro 1
D3	Disco de arranque del miembro 2
D4	Disco de Quórum
D5	Disco usado por aplicación Servidor Web
D6	Disco usado por aplicación Manejador de B. D.
D7	Disco usado por B. D.
D8	Disco usado por B. D.

Tabla 5.3.4-2 Descripción del uso de cada unidad.

5.3.4.2 Sistema de Archivos del Cluster (Cluster File System)

Todos los sistemas de archivos creados al momento de instalar y configurar el cluster estarán disponibles a todos los miembros del mismo. Esto significa que cualquier nodo podrá leer y escribir información de y hacia ellos. Sin embargo, los sistemas de archivos que usaremos para instalar y configurar la aplicación así como para el almacenamiento de la información deben ser creados posteriormente.

Para soportar el funcionamiento del Servidor Web y la Base de Datos, se necesita crear dos sistemas de archivos, la Tabla 5.3.4.2-1 muestra las características de cada uno de ellos.

Nombre	Disco	Tamaño	Punto de montaje
Web	dsk5	4 GB	/web
Dbm	Dsk6	4 GB	/dbm
datos1	Dsk7	10 GB	/datos
datos2	Dsk8	18 GB	/datos

Tabla 5.3.4.2-1 Características de los sistemas de archivos de la aplicación.

Los pasos a seguir para configurar los sistemas de archivos son:

- Crear los dominios
  - `mkfdmn Dispositivo Nombre_dominio`
- Crear los filesets
  - `mkfset Nombre_dominio Nombre_Fest`
- Montar los sistemas de archivos
  - `mount Nombre_dominio# Nombre_fset montaje`

La secuencia de comandos para crear los sistemas de archivos es la siguiente:

```
mkfdmn /dev/disk/dsk5c webdmn
mkfdmn /dev/disk/dsk6c dbmdmn
```

Para los discos dsk7 y dsk8 queremos crear un solo dispositivo, para ello se define el dominio con uno de los discos y se le adiciona a ese dominio el otro disco:

```
mkfdmn /dev/disk/dsk7c datosdmn
addvol /dev/dsik/dsk8c datosdmn

mkfset webdmn webfset
mkfset dbmdmn dbmfset
mkfset datosdmn datosfset

mount webdmn#webfset /web
mount dbmdmn#dbmfset /dbm
mount datosdmn#datosfset /datos
```

Después de esto, los tres sistemas de archivos (/web, /dbm y /datos) podrán ser accedidos por cualquiera de los miembros del cluster.

5.3.5 Redundancia de otros componentes

Además de la redundancia del sistema, tal como los discos, las tarjetas de red, los buses, las fuentes de alimentación, cableado, entre otros más, es recomendable asegurar elementos que tienen como función proteger al site y a los elementos del cluster.

La fuente de energía en nuestro sistema de cómputo es muy importante y si consideramos que los apagones de la compañía suministradora son muy comunes estableceremos redundancia en la fuente, un SAI nos permite lograr tal efecto: Baterías, cargador de baterías, onduladores, no-breaks. Se contará con doble acometida de red eléctrica, una por parte de la compañía suministradora y otra por parte del SAI.

La redundancia en la ventilación es muy importante, pues algunos elementos tienden a incrementar su temperatura con la actividad diaria (24 horas por día), de esta manera aplicaremos redundancia en los ventiladores internos para cada uno de los dispositivos. La redundancia en dispositivos que generen un ambiente agradable en el site es también necesaria; la temperatura y humedad deben estar en condiciones que permitan el buen funcionamiento de los dispositivos, tal como aire acondicionado y ventiladores.

Estableceremos redundancia también en el sistema de cableado y los elementos relacionados con ésta característica, tal como cajas de conexiones, conectores especiales, entre otros más para lograr circulación continua de información y alimentación en el sistema.

Otra redundancia de vital importancia, está en los equipos que resguardan la seguridad ante la eventualidad de un incendio, tal como los extinguidores y detectores de humo.

Otra redundancia que también se deberá implementar, está en la seguridad de acceso al site tal como puertas con código de acceso y cámaras con circuito cerrado. En la administración del sistema se contará con una contraseña altamente confidencial para ingresar al mismo. Otra redundancia muy útil estará en los canales del cableado, pues para cada línea que corra en el cluster, existirá un canal específico. Para una fácil reparación y mantenimiento de los canales se contará con piso de rejilla.

5.3.6 Servicios o aplicaciones altamente disponibles

El paso final en la implementación de un cluster como solución de alta disponibilidad para una aplicación, es el de poner bajo el control del mismo los procesos y/o programas que la forman.

Para tal efecto, es necesario realizar las actividades siguientes basándonos principalmente en el componente del cluster llamado CAA (Cluster Availability Aplication).

- Crear un perfil del recurso, en el cual se puede especificar la dependencia con otros recursos, los miembros que la pueden acceder y el procedimiento a ejecutar en caso de falla
- Crear uno o varios scripts de acción, que serán usados por el cluster para iniciar y detener los procesos y programas de la aplicación de forma automática en caso necesario
- Registrar la aplicación (Darla de alta en la base de datos del CAA del cluster)
- Haciendo uso de procedimientos de cluster, iniciar la aplicación
- Realizar pruebas de relocalización de los procesos y programas que forman la nueva aplicación altamente disponible

En nuestro caso, describiremos cada una de las actividades mencionadas al dar de alta en el cluster el servidor web Apache y el manejador de base de datos, necesarios para la implementación de un sistema altamente disponible para la venta en línea por Internet.

Servidor Web Apache

El primer paso es tener la aplicación instalada, ya que sobre ella se trabajará en los scripts para su manejo (iniciar, detener y verificar la aplicación).

Al instalar la aplicación se debe tomar en cuenta se sea instalado en el directorio creado anteriormente para dicho propósito, este directorio tiene el nombre de `/web`, y además se debe utilizar el alias del cluster (zeus.dominio.com) como nombre del servidor de web.

Es recomendable definir un alias de cluster , como se menciono anteriormente, para romper la dependencia de los usuarios de la aplicación hacia las direcciones IP propias de cada miembro del cluster. Por este motivo, se recomienda también configurar el puerto de conexión que el servidor web utilizará bajo el control del cluster; ya que en caso de falla tanto la dirección IP virtual como el puerto de conexión serán relocalizados al servidor de respaldo. Para ello, se debe agregar la siguiente línea en el archivo `/etc/clua_services`, el cual es utilizado para definir los puertos, protocolos y atributos de conexión para servicios de Internet que usará el cluster. Nuestro servidor web utilizará el puerto de comunicación 80.

```
http        80/tcp          in_single
```

El atributo "`in_single`" le indica al subsistema del cluster que deberá canalizar las peticiones de conexión al nodo predeterminado del cluster pero solo a uno a la vez. En caso de que este nodo no esté disponible, se redirigirán las peticiones al otro nodo.

Se necesita recargar las definiciones de los servicios para que los cambios recientes sean activados, para ello se utiliza el siguiente comando en los dos nodos del cluster:

```
# cluamgr -f
```

El comando `cluamgr` es una interfaz de línea de comando con el cual se especifican y manejan los alias del cluster. Con el parámetro `-f`, se lee nuevamente el archivo de configuración de servicios `/etc/clua_services`.

El servidor de web Apache se iniciará con un script llamado `apache.scr`, el cual se copiará a `/var/cluster/caa/script/apache.scr`. Este script se verá posteriormente con más detalle.

El siguiente paso es crear un perfil de recursos tipo aplicación que en este caso se llamara *apache*. Para ello se utiliza el comando caa_profile, el cual a través de la línea de comando nos permite ejecutar tareas relacionadas con los perfiles de los recursos CAA. En este caso, se utilizó el comando como sigue:

```
# caa_profile -create apache -t application
```

Este comando crea un archivo en `/var/cluster/caa/profile/apache.cap` el cual tiene los siguientes parámetros, a los cuales hay que establecerles algunos valores, ya que este comando lo genera con los valores genéricos para que funcione. Aunque se pueden definir valores agregándole parámetros al comando, en este caso dicho comando sería muy largo, por eso mejor se edita el archivo generado.

apache.cap

`NAME = apache`	Nombre del recurso.
`TYPE = application`	Tipo de recurso.
`ACTION_SCRIPT = apache.scr`	Script para iniciar, detener y checar el recurso.
`ACTIVE_PLACEMENT = 0`	Si es 1 reevalúa el nodo en el se alojará el recurso.
`AUTO_START = 1`	Si es 1 automáticamente inicia el recurso después de un reinicio del cluster.
`CHECK_INTERVAL = 60`	Intervalo de tiempo, en segundos, en el cual se verifica el estado del recurso.

`DESCRIPTION = Apache Web Server`	Descripción del recurso.
`FAILOVER_DELAY = 10`	Intervalo de tiempo, en segundos, que hay que esperar para reiniciar o hacer un failover del recurso.
`FAILURE_INTERVAL = 15`	Intervalo de tiempo, en segundos, en el que se aplica el valor de FAILURE_THRESHOLD.
`FAILURE_THRESHOLD = 1`	Número de fallas detectadas en un intervalo de tiempo (FAILURE_INTERVAL) antes de marcar el recurso como NO disponible para dejar de monitorearlo.
`HOSTING_MEMBERS = juno hera`	Una lista ordenada separada por espacios en blanco donde se indican los miembros donde se puede albergar el recurso.
`OPTIONAL_RESOURCES =`	Una lista separada por espacios en blanco de los recursos opcionales que utiliza este recurso.
`PLACEMENT = favored`	Se define la forma en que el CAA elegirá un nodo para alojar el recurso.
`REQUIRED_RESOURCES = net1`	Una lista separada por espacios en blanco de los recursos de los cuales depende.
`RESTART_ATTEMPTS = 3`	Número de intentos de iniciar el recurso en el miembro actual del cluster, antes de intentar elegir otro miembro.
`SCRIPT_TIMEOUT = 60`	Intervalo de tiempo, en segundos, que se tarda en ejecutar el script de verificación del estado del recurso.

Como se puede apreciar, existe un recurso requerido por nuestra aplicación, que se define como `net1`, el cual es la conexión a la red. Este recurso se debe definir para que el cluster pueda reaccionar en caso de existir una falla en la conexión de red. Para crear el recurso requerido de `net1`, se teclea el comando como sigue:

```
# caa_profile -create net1 -t network -s "16.179.20.0"
```

El archivo `net1.cap` tiene los siguientes parámetros:

`NAME = net1`	Nombre del recurso.
`TYPE = network`	Tipo de recurso.
`DESCRIPTION = Gb Network`	Descripción del recurso.
`FAILURE_INTERVAL = 15`	Intervalo de tiempo, en segundos, en el que se aplica el valor de FAILURE_THRESHOLD.
`FAILURE_THRESHOLD = 1`	Número de fallas detectadas en un intervalo de tiempo (FAILURE_INTERVAL) antes de marcar el recurso como NO disponible para dejar de monitorearlo.
`SUBNET = 16.179.20.0`	Dirección de la subred al que pertenece el recurso de red.

Después se deben registrar ambos recursos en el cluster. A partir de este momento el cluster se entera y toma control de los mismos. Para ello se ejecutan los siguientes comandos:

```
# caa_register apache
# caa_register net1
```

Los comandos caa_register y caa_unregister permiten poner o sacar del control del cluster las aplicaciones.

Para iniciar y detener los recursos o aplicaciones que se encuentran registrados se utilizan los comandos `caa_start` y `caa_stop`. Lo que hacen estos comandos es ejecutar los scripts que se definieron en el perfil del recurso. El caa_start llama y ejecuta el script de arranque, mientras que el comando caa_stop llama y ejecuta el script de paro de los procesos y programas de la aplicación. De esta forma en cualquier momento se puede detener y arrancar cualquier aplicación del cluster.

Con el siguiente comando se puede verificar en todo momento el estado de las aplicaciones bajo el control del cluster.

```
# caa_stat apache
```

La salida que proporciona es:

```
NAME = apache
TYPE = application
STATE = OFFLINE
```

Es recomendable que antes de registrar la aplicación, se pruebe el correcto funcionamiento de los scripts asociados con dicha aplicación. Para probarlos se deben ejecutar los scripts primero a nivel sistema operativo como sigue:

Para iniciar:

```
# /var/cluster/caa/script/apache.scr start
```

Para detener:

```
# /var/cluster/caa/script/apache.scr stop
```

Ya que el recurso de red "net1" se definió como recurso requerido dentro del perfil de las aplicaciones, es importante iniciar primero el mismo para que al momento de iniciar las aplicaciones ya se encuentre disponible. Para ello se ejecuta el siguiente comando:

```
# caa_start net1
```

Para iniciar el Servidor Web Apache se usa el comando:

```
# caa_start apache
```

Para verificar el estado del Servidor Web Apache se tiene lo siguiente:

```
# caa_stat apache
RESOURCE = apache
TYPE = application
STATE = ONLINE on juno
```

El comando caa_stat tiene el propósito de verificar que cualquier recurso registrado esté en línea. Al ejecutar el comando se muestra una salida, donde se debe tener especial atención en el parámetro STATE. En este ejemplo si la

aplicación está corriendo satisfactoriamente el valor será ONLINE, en caso contrario el parámetro STATE tendrá el valor OFFLINE indicando que la aplicación esta fuera de servicio o que por alguna razón, la misma no está corriendo satisfactoriamente.

Manejador de Base de Datos

La aplicación de base de datos utilizará un sistema de archivos para almacenar el manejador (/dbm) y otro para almacenar los datos (/datos.). Ésto, por razones de administración y desempeño.

Recordemos que al momento de crear un sistema de archivos y montarlo, el mismo queda automáticamente disponible para todos los miembros del cluster. Seguiremos la misma secuencia de actividades que fueron usadas al configurar el servidor web para poner bajo el control del cluster esta aplicación.

Se necesita introducir la siguiente línea en el archivo `/etc/clua_services`:

```
lsnrctl          1025/tcp          in_single
```

Se reinicia el alias del cluster con la modificación recientemente hecha:

```
# cluamgr -f
```

El script que el cluster usará para iniciar y detener la aplicación de base de datos es `oracle.scr`, el cual también se encuentra en:
`/var/cluster/caa/script/oracle.scr`

Entonces creamos el perfil del recurso tipo aplicación:

```
# caa_profile -create oracle -t application
```

El perfil de este recurso que es `/var/cluster/caa/profile/oracle.cap` es:

oracle.cap

```
NAME = oracle
TYPE = application
ACTION_SCRIPT = oracle.scr
ACTIVE_PLACEMENT = 0
AUTO_START = 1
CHECK_INTERVAL = 60
DESCRIPTION = Base de Datos
FAILOVER_DELAY = 10
FAILURE_INTERVAL = 15
FAILURE_THRESHOLD = 1
HOSTING_MEMBERS = hera juno
OPTIONAL_RESOURCES =
PLACEMENT = favored
REQUIRED_RESOURCES = net1
RESTART_ATTEMPTS = 3
SCRIPT_TIMEOUT = 60
```

Antes de registrarlo se prueba el script del recurso.

Para iniciar:

```
# /var/cluster/caa/script/oracle.scr start
```

Para detener:

```
# /var/cluster/caa/script/oracle.scr stop
```

Se registra el servicio en el CAA:

```
# caa_register oracle
```

Se inicia el servicio y se verifica su estado:

```
# caa_start oracle
```

```
# caa_stat oracle
RESOURCE = oracle
TYPE = application
STATE = ONLINE on hera
```

5.3.7 Scripts

Script que utiliza el cluster para el control del servidor web Apache.

/var/cluster/caa/script/apache.scr

```
#!/usr/bin/ksh -p
#
# TruCluster V5 CAA script for Apache Webserver
#
# 8<----------- Start variables 8<----------
#
svcName="apache" # Nombre de aplicación
CAA_ADMIN="root" # Cuenta que recibirá avisos de la aplicación
CAALOGDIR="/var/cluster/caa/log" # Directorio de las bitácoras
ACTION=$1 # Acción a ejecutar (puede ser start o stop)
LOG="${CAALOGDIR}/${ACTION}_${svcName}.$$" # Bitácora de sucesos de la
aplicación
#LOG="/dev/console" # Mandar los mensajes de sucesos a la consola.
#
# Application specific stuff
#
PROBE_PROCS="httpd" # Proceso a probar
START_APPCMD="/web/apache/bin/apachectl start" # Comando de inicio de la
aplicación
START_APPCMD2="" # Comando de inicio de la aplicación 2
STOP_APPCMD="/web/apache/bin/apachectl stop" # Comando para detener  la
aplicación
STOP_APPCMD2="" # Comando para detener  la aplicación 2
APPDIR="/web/apache" # Directorio donde está instalada la aplicación
ADVFSDIRS=" " # Application directories to
#
FUSER="/usr/sbin/fuser" # Comando para cerrar conexiones de usuarios
EVMPOST="/usr/bin/evmpost -p 650 -a" # Comando para registrar eventos
#
export START_APPCMD START_APPCMD2 STOP_APPCMD STOP_APPCMD2 APPDIR
export ADVFSDIRS PROBE_PROCS
#
#
# Main section
#
# Start section
#
case $1 in
'start')
echo "" >> ${LOG}
echo ""Start action script for service : ${svcName} \
`/bin/date +"%A %d %B %H:%M:%S"` "" >> ${LOG}
#
# Start Apache Web Server
#
echo "Inicializando Apache Web Server... " >> ${LOG}
cd $APPDIR
```

```
$START_APPCMD >> ${LOG}
if [ $? -ne 0 ]; then
postevent "Apache Web Server" start
exit 2
fi
echo "Iniciado Apache Web Server" >> ${LOG}
#
# All done ...
#
${EVMPOST} "Start action script for service ${svcName} DONE"
echo ""Start action script for service ${svcName} DONE, \
`/bin/date +"%A %d %B %H:%M:%S"` "" >> ${LOG}
echo "" >> ${LOG}
exit 0
#
;;
# Stop section
#
'stop')
echo "" >> ${LOG}
echo ""Stop action script for service : ${svcName} \
`/bin/date +"%A %d %B %H:%M:%S"` "" >> ${LOG}
#
# Stop Apache Web Server
#
echo "Stopping Apache Web Server ... " >> ${LOG}
$STOP_APPCMD >> ${LOG}
if [ $? -ne 0 ]; then
postevent "Apache Web Server" stop
exit 2
fi
echo "Apache Web Server shutdown done ." >> ${LOG}

${EVMPOST} "Stop action script for service ${svcName} DONE"
echo ""Stop action script for service ${svcName} DONE, \
`/bin/date +"%A %d %B %H:%M:%S"` "" >> ${LOG}
echo "" >> ${LOG}
exit 0
;;
#
# Probe if application is still alive
#
'check')
echo ""Probing Apache daemons at \
`/bin/date +"%A %d %B %H:%M:%S"`"" >> ${LOG}
for i in ${PROBE_PROCS}
do
probeapp ${i} >> ${LOG}
done
echo ""Probing Apache daemons DONE at \
`/bin/date +"%A %d %B %H:%M:%S"`"" >> ${LOG}
exit 0
;;
*) echo "usage: $0 {start|stop|check}"
exit 1
;;
esac
```

Script que inicia o detiene el servidor web Apache.

/web/apache/bin/apachectl

```
#!/bin/sh
#
# Copyright (c) 2000-2002 The Apache Software Foundation.
# See license at the end of this file.
#
# Apache control script designed to allow an easy command line interface
# to controlling Apache.  Written by Marc Slemko, 1997/08/23
#
# The exit codes returned are:
#   XXX this doc is no longer correct now that the interesting
#   XXX functions are handled by httpd
#     0 - operation completed successfully
#     1 -
#     2 - usage error
#     3 - httpd could not be started
#     4 - httpd could not be stopped
#     5 - httpd could not be started during a restart
#     6 - httpd could not be restarted during a restart
#     7 - httpd could not be restarted during a graceful restart
#     8 - configuration syntax error
#
# When multiple arguments are given, only the error from the _last_
# one is reported.  Run "apachectl help" for usage info
#
ARGV="$@"
#
# ||||||||||||||||||||| START CONFIGURATION SECTION ||||||||||||||||||||||
# ----------------------                            ----------------------
#
# the path to your httpd binary, including options if necessary
HTTPD='/web/apache/bin/httpd'
#
# pick up any necessary environment variables
if test -f /web/apache/bin/envvars; then
  . /web/apache/bin/envvars
fi
#
# a command that outputs a formatted text version of the HTML at the
# url given on the command line.  Designed for lynx, however other
# programs may work.
LYNX="lynx -dump"
#
# the URL to your server's mod_status status page.  If you do not
# have one, then status and fullstatus will not work.
STATUSURL="http://localhost:80/server-status"
#
# ----------------------                            ----------------------
# |||||||||||||||||||||   END CONFIGURATION SECTION  |||||||||||||||||||||

ERROR=0
```

```
if [ "x$ARGV" = "x" ] ; then
    ARGV="-h"
fi

case $ARGV in
start|stop|restart|graceful)
    $HTTPD -k $ARGV
    ERROR=$?
    ;;
startssl|sslstart|start-SSL)
    $HTTPD -k start -DSSL
    ERROR=$?
    ;;
configtest)
    $HTTPD -t
    ERROR=$?
    ;;
status)
    $LYNX $STATUSURL | awk ' /process$/ { print; exit } { print } '
    ;;
fullstatus)
    $LYNX $STATUSURL
    ;;
*)
    $HTTPD $ARGV
    ERROR=$?
esac

exit $ERROR

# ====================================================================
# The Apache Software License, Version 1.1
#
# Copyright (c) 2000-2002 The Apache Software Foundation.  All rights
# reserved.
#
# Redistribution and use in source and binary forms, with or without
# modification, are permitted provided that the following conditions
# are met:
#
# 1. Redistributions of source code must retain the above copyright
#    notice, this list of conditions and the following disclaimer.
#
# 2. Redistributions in binary form must reproduce the above copyright
#    notice, this list of conditions and the following disclaimer in
#    the documentation and/or other materials provided with the
#    distribution.
#
# 3. The end-user documentation included with the redistribution,
#    if any, must include the following acknowledgment:
#       "This product includes software developed by the
#        Apache Software Foundation (http://www.apache.org/)."
#    Alternately, this acknowledgment may appear in the software itself,
#    if and wherever such third-party acknowledgments normally appear.
#
# 4. The names "Apache" and "Apache Software Foundation" must
#    not be used to endorse or promote products derived from this
```

```
#    software without prior written permission. For written
#    permission, please contact apache@apache.org.
#
# 5. Products derived from this software may not be called "Apache",
#    nor may "Apache" appear in their name, without prior written
#    permission of the Apache Software Foundation.
#
# THIS SOFTWARE IS PROVIDED ``AS IS'' AND ANY EXPRESSED OR IMPLIED
# WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
# OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE
# DISCLAIMED.  IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR
# ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
# SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
# LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF
# USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND
# ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,
# OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT
# OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
# SUCH DAMAGE.
# ====================================================================
#
# This software consists of voluntary contributions made by many
# individuals on behalf of the Apache Software Foundation.  For more
# information on the Apache Software Foundation, please see
# <http://www.apache.org/>.
#
# Portions of this software are based upon public domain software
# originally written at the National Center for Supercomputing
Applications,
# University of Illinois, Urbana-Champaign.
#
```

Script que utiliza el cluster para el control del manejador de base de datos.

/var/cluster/caa/script/oracle.scr

```
#!/usr/bin/ksh -p
#
# TruCluster V5 CAA script for Secure Shell
#
# 8<------------ Start variables 8<----------
#
svcName="oracle" # Nombre de aplicación
CAA_ADMIN="root" # Cuenta que recibirá avisos de la aplicación
CAALOGDIR="/var/cluster/caa/log" # Directorio de las bitácoras
ACTION=$1 # Acción a ejecutar (puede ser start o stop)
LOG="${CAALOGDIR}/${ACTION}_${svcName}.$$" # Bitácora de sucesos de la
aplicación
#LOG="/dev/console" # Mandar los mensajes de sucesos a la consola.
```

```
#
# Application specific stuff
#
PROBE_PROCS="oracle" # Proceso a probar
START_APPCMD="/dbm/dbm.sh start" # Comando de inicio de la aplicación
START_APPCMD2="" # Comando de inicio de la aplicación 2
STOP_APPCMD="/dbm/dbm.sh stop" # Comando para detener  la aplicación
STOP_APPCMD2="" # Comando para detener  la aplicación 2
APPDIR="/dbm" # Directorio donde está instalada la aplicación
ADVFSDIRS=" " # Application directories to
#
FUSER="/usr/sbin/fuser" # Comando para cerrar conexiones de usuarios
EVMPOST="/usr/bin/evmpost -p 650 -a" # Comando para registrar eventos
#
export START_APPCMD START_APPCMD2 STOP_APPCMD STOP_APPCMD2 APPDIR
export ADVFSDIRS PROBE_PROCS
#
#
# Main section
#
# Start section
#
case $1 in
'start')
echo "" >> ${LOG}
echo ""Start action script for service : ${svcName} \
`/bin/date +"%A %d %B %H:%M:%S"` "" >> ${LOG}
#
# Start Base de Datos
#
echo "Inicializando Base de Datos... " >> ${LOG}
cd $APPDIR
$START_APPCMD >> ${LOG}
if [ $? -ne 0 ]; then
postevent "Base de Datos" start
exit 2
fi
echo "Iniciada Base de Datos" >> ${LOG}
#
# All done ...
#
${EVMPOST} "Start action script for service ${svcName} DONE"
echo ""Start action script for service ${svcName} DONE, \
`/bin/date +"%A %d %B %H:%M:%S"` "" >> ${LOG}
echo "" >> ${LOG}
exit 0
#
;;
#
# Stop section
#
'stop')
echo "" >> ${LOG}
echo ""Stop action script for service : ${svcName} \
`/bin/date +"%A %d %B %H:%M:%S"` "" >> ${LOG}
#
# Detener Base de Datos
```

```
#
echo "Deteniendo Base de Datos ... " >> ${LOG}
$STOP_APPCMD >> ${LOG}
if [ $? -ne 0 ]; then
postevent "Base de Datos" stop
exit 2
fi
echo "Base de Datos shutdown done ." >> ${LOG}

${EVMPOST} "Stop action script for service ${svcName} DONE"
echo ""Stop action script for service ${svcName} DONE, \
`/bin/date +"%A %d %B %H:%M:%S"` "" >> ${LOG}
echo "" >> ${LOG}
exit 0
;;
#
# Probe if application is still alive
#
'check')
echo ""Probing ssh daemons at \
`/bin/date +"%A %d %B %H:%M:%S"`"" >> ${LOG}
for i in ${PROBE_PROCS}
do
probeapp ${i} >> ${LOG}
done
echo ""Probing ssh daemons DONE at \
`/bin/date +"%A %d %B %H:%M:%S"`"" >> ${LOG}
exit 0
;;
*) echo "usage: $0 {start|stop|check}"
exit 1
;;
esac
```

Script que inicia o detiene el manejador de base de datos.

/dbm/dbm.sh

```
#!/bin/ksh
# Descripcion: Script para integracion a cluster
#

Cluster_Up () {
print - "Entering ${SAM_SERV} START section:"
#####################################################
#
# Configuration and level setup on priority balancig
# on IP service address to balancing to 10 value.
# This increase priority on server with application
# running
#
#####################################################
print - " "
```

```
print - " "
print - "-------------------------------------------"
proc_timeout "$CLUAMGR_UPDOWN1" 60 10 &
print - "Action: Increasing Priority in IP service address"
print - " "
print - " "
integer returncode=0
/usr/sbin/cluamgr -a selw=100,selp=10,join,alias=$CLUAMGR1
returncode=$?
case $returncode in
                0) print - "Success: Increase Priority in IP service
succeeded"
                   ;;
                *) print - "Error: Increase Priority in IP service
failed"
                   ValUp=$ValUp+1
                   ;;
esac
#####################################################
#
# Configuration and level setup on priority balancing
# on Gigabit IP service address to balancing to 10 value.
# This increase priority on server with application
# running
#
#####################################################
print - " "
print - " "
print - "-------------------------------------------"
proc_timeout "$CLUAMGR_UPDOWN1" 60 10 &
print - "Action: Increasing Priority in Gigabit IP service address"
print - " "
print - " "
integer returncode=0
/usr/sbin/cluamgr -a selw=100,selp=10,join,alias=$CLUAMGR2
returncode=$returncode+$?
case $returncode in
                0) print - "Success: Increase Priority in GB IP service
succeeded"
                   ;;
                *) print - "Error: Increase Priority in GB IP service
failed"
                   ValUp=$ValUp+1
                   ;;
esac
integer verialiasF=0
integer verialiasG=0
integer returncode=0
verialiasF=`/sbin/arp -a | grep $CLUAMGR1 | wc -l`
verialiasG=`/sbin/arp -a | grep $CLUAMGR2 | wc -l`
Verito=$verialiasF+$verialiasG
if [ $Verito -eq 0 ] ; then
   CurrentHost=`hostname`
   if [ "$CurrentHost" = "hera" ] ; then
      rsh juno "/usr/sbin/cluamgr -r start"
      returncode=$returncode+$?
      case $returncode in
```

```
                        0) print - "Success: Start IP service succeeded"
                           ;;
                        *) print - "Error: Start IP service failed"
                           ValUp=$ValUp+1
                           ;;
      esac
   else
       rsh hera "/usr/sbin/cluamgr -r start"
       returncode=$returncode+$?
       case $returncode in
                        0) print - "Success: Start IP service succeeded"
                           ;;
                        *) print - "Error: Start IP service failed"
                           ValUp=$ValUp+1
                           ;;
       esac
   fi
fi
#####################################################
#
# Start oracle database SAM
#
#####################################################
print - " "
print - " "
print - "-------------------------------------------"
proc_timeout "$SAM_START" 300 10 &
print - "Action: check the database type and start it..."
print - " "
print - " "
Usu=`grep -E "oracle|oracle73" /etc/passwd|wc -l`
if [ ${Usu} -gt 0 ] ; then
   Ruta=/etc
   NomArch=oratab
   integer Tot_Proc_Oracle=0
   integer NumProc=0
   Ora=`grep -v "#" /etc/oratab |grep -E ":Y|:N"  |awk -F: '{print $1}'`
   Ver=`grep -v "#" /etc/oratab |grep -E ":Y|:N"  |awk -F: '{print $2}'`
   set -A Db $Ora
   set -A Vs $Ver
   integer DB_UP=2
   for file in $Ora
       do
       Tot_Proc_Oracle=`ps -fea | grep -E "ora_" | grep _${file} | wc -l`
       if [ $Tot_Proc_Oracle -ge 6 ] ; then
          integer DB_UP=1
       else
           integer DB_UP=0
       fi
       NumProc=${NumProc}+1
       done
fi
if [ $DB_UP -eq 0 ] ; then
   su - ${ADMUSER} -c ". /dbm/.profile ; dbstart"
   returncode=$?
   case $returncode in
                    0) print - "Success: startup DB oracle SAM succeeded"
```

```
                    ;;
                *) print - "Error: startup DB oracle SAM failed"
                   ValUp=$ValUp+1
                   ;;
   esac
else
    if [ $DB_UP -eq 1 ] ; then
       print - "Info: Database SAM already running "
    fi
fi
#####################################################
#
# Start listener
#
#####################################################
print - " "
print - " "
print - "-------------------------------------------"
proc_timeout "$LISTENER_START" 60 10 &
print - "Action: listener starting..."
print - " "
print - " "
integer Proc_List=0
Proc_List=`ps -ef|grep -v grep|grep tnslsnr|wc -l`
if [ $Proc_List -eq 0 ] ; then
   su - ${ADMUSER} -c "lsnrctl start > /dev/null 2>&1"
   returncode=$?
   case $returncode in
               0) print - "Success: startup listener SAM succeeded"
                  ;;
               *) print - "Error: startup listener SAM failed"
                  ValUp=$ValUp+1
                  ;;
   esac
else
    print - "Info: Listener already running"
fi
#####################################################
#
# Start CRON
#
#####################################################
print - " "
print - " "
print - "-------------------------------------------"
print - "Action: starting ccmirand account cron..."
print - " "
print - " "
echo  su - ccmirand -c "crontab /usr/people/ccmirand/cron_ccmirand.ase"
returncode=$?
case $returncode in
                0) print - "Success: startup of ccmirand-cron succeeded"
                   ;;
                *) print - "Error: startup of ccmirand-cron failed"
                   ValUp=$ValUp+1
                   ;;
esac
```

```
#####################################################
#
# end of start section
#
#####################################################
print - " "
print - " "
print - "-------------------------------------------"
print - "Info: Completed - Start of script ${SCRIPT} for service \
      ${SAM_SERV}"
print - "Info: on hostname ${HOSTNAME} at `/bin/date`"
}

Cluster_Down () {
print - "Entering ${SAM_SERV} STOP section:"
#####################################################
#
# Configuration and level setup on priority balancig
# on IP service address to balancing to 5 value.
# This decrease priority on server with application
# running
# Is necessary to have the file:
# /etc/clu_alias.config
#
#####################################################
print - " "
print - " "
print - "-------------------------------------------"
proc_timeout "$CLUAMGR_UPDOWN1" 60 10 &
print - "Action: Decreasing Priority in IP service address"
print - " "
print - " "
integer returncode=0
valprio=`grep $CLUAMGR1 /etc/clu_alias.config | awk -F, '{print $2}'|sed
"s/selp\=//"`
/usr/sbin/cluamgr -a selw=100,selp=$valprio,join,alias=$CLUAMGR1
returncode=$returncode+$?
case $returncode in
                0) print - "Success: Decrease Priority in IP service
succeeded"
                   ;;
                *) print - "Error: Decrease Priority in IP service
failed"
                   ValUp=$ValUp+1
                   ;;
esac
#####################################################
#
# Configuration and level setup on priority balancig
# on Gigabit IP service address to balancing to 5 value.
# This decrease priority on server with application
# running
# Is necessary to have the file:
# /etc/clu_alias.config
#
#####################################################
print - " "
```


ESTA TESIS NO SALE
DE LA BIBLIOTECA

```
print - " "
print - "-------------------------------------------"
proc_timeout "$CLUAMGR_UPDOWN1" 60 10 &
print - "Action: Decreasing Priority in Gigabit IP service address"
print - " "
print - " "
integer returncode=0
valprio=`grep $CLUAMGR2 /etc/clu_alias.config | awk -F, '{print $2}'|sed
"s/selp\=//"`
/usr/sbin/cluamgr -a selw=100,selp=$valprio,join,alias=$CLUAMGR2
returncode=$returncode+$?
case $returncode in
                0) print - "Success: Decrease Priority in GB IP service
succeeded"
                   ;;
                *) print - "Error: Decrease Priority in GB IP service
failed"
                   ValUp=$ValUp+1
                   ;;
esac
##################################################
#
# Stop the oracle database SAM
#
##################################################
print - " "
print - " "
print - "-------------------------------------------"
proc_timeout "$SAM_STOP" 300 10 &
print - "Action: oracle database stopping..."
print - " "
print - " "
Usu=`grep -E "oracle|oracle73" /etc/passwd|wc -l`
if [ ${Usu} -gt 0 ] ; then
   Ruta=/etc
   NomArch=oratab
   integer Tot_Proc_Oracle=0
   integer NumProc=0
   Ora=`grep -v "#" /etc/oratab |grep -E ":Y|:N"  |awk -F: '{print $1}'`
   Ver=`grep -v "#" /etc/oratab |grep -E ":Y|:N"  |awk -F: '{print $2}'`
   set -A Db $Ora
   set -A Vs $Ver
   integer DB_UP=2
   for file in $Ora
       do
       Tot_Proc_Oracle=`ps -fea | grep -E "ora_" | grep _${file} | wc -l`
       if [ $Tot_Proc_Oracle -ge 6 ] ; then
          integer DB_UP=1
       else
           integer DB_UP=0
           fi
       NumProc=${NumProc}+1
       done
   fi
if [ $DB_UP -eq 1 ] ; then
   su - ${ADMUSER} -c ". /dbm/.profile ; dbshut"
   returncode=$?
```

```
   case $returncode in
                    0) print - "Success: Stop for DB oracle SAM
succeeded"
                       ;;
                    *) print - "Error: Stop for DB oracle SAM failed"
                       ValUp=$ValUp+1
                       ;;
   esac
else
    if [ $DB_UP -eq 0 ] ; then
       print - "Info: Database SAM not running "
       fi
fi
#####################################################
#
# Stop listener
#
#####################################################
print - " "
print - " "
print - "-------------------------------------------"
proc_timeout "$LISTENER_STOP" 60 10 &
print - "Action: listener Stopping..."
print - " "
print - " "
integer Proc_List=0
Proc_List=`ps -ef|grep -v grep|grep tnslsnr|wc -l`
if [ $Proc_List -eq 1 ] ; then
   su - ${ADMUSER} -c "lsnrctl stop"
   returncode=$?
   case $returncode in
                    0) print - "Info: stop listener SAM succeeded"
                       ;;
                    *) print - "Info: stop listener SAM failed"
                       ValUp=$ValUp+1
                       ;;
   esac
else
    print - "Info: Listener not running"
fi
#####################################################
#
# Stop CRON
#
#####################################################
print - " "
print - " "
print - "-------------------------------------------"
print - "Action: stopping ccmirand account cron..."
print - " "
print - " "
echo  su - ccmirand -c "crontab -r"
returncode=$?
case $returncode in
                0) print - "Success: stop of ccmirand-cron succeeded"
                   ;;
                *) print - "Error: stop of ccmirand-cron failed"
```

```
                    ValUp=$ValUp+1
                    ;;
esac
#######################################################
#
# end of stop section
#
#######################################################
print - " "
print - " "
print - "-------------------------------------------"
print - "Info: Completed - Stop of script ${SCRIPT} for service \
       ${SAM_SERV}"
print - "Info: on hostname ${HOSTNAME} at `/bin/date`"
}


#####################################################
#
# Begin Main Program
#
#####################################################
if [ $# -eq 1 ] ; then
   CMD=$1
   ###################################################
   #
   # Read in the site specific definitions from:
   # /var/cluster/caa/profile/dbmenv.conf
   # modify this entry if your directory path differs
   # from the one that is used next.
   #
   ###################################################
   . /var/cluster/caa/profile/dbmenv.conf
   print - " "
   print - "-------------------------------------------"
   print - "-------------------------------------------"
   integer ValUp=0
   case "$CMD" in

            start)
                print - "Info : Running script ${SCRIPT} ${CMD}"
                print - "Info : at `/bin/date` on  hostname ${HOSTNAME}"
                Cluster_Up
                if [ $ValUp -ne 0 ] ; then
                   print - "Info : With problems plase check"
                fi
                ;;
            stop)
                print - "Info : Running script ${SCRIPT} ${CMD}"
                print - "Info : at `/bin/date` on  hostname ${HOSTNAME}"
                Cluster_Down
                if [ $ValUp -ne 0 ] ; then
                   print - "Info : With problems plase check"
                fi
                ;;
           *)
                echo "utilizar: $0 start | stop"
```

```
                ;;
    esac
    print - "----------------------------------------"
    print - "----------------------------------------"
    exit $ValUp
else
    echo "utilizar: $0 start | stop"
fi
```

6 CONCLUSIONES

En la actualidad muchas actividades cotidianas dependen de un sistema de cómputo, debido a eso, la disponibilidad que los sistemas que usan tecnología de información requieren hoy en día, es mucho mayor que antes.

Diversos estudios muestran que el costo económico de la caída de un sistema de información es muy grande a tal grado que puede llevar a la quiebra a una organización. Por tal motivo es muy importante que mientras más alto sea el riesgo y el costo del downtime, se busque implementar una solución que lo impida.

Debido a que aproximadamente el treinta por ciento de la disponibilidad del sistema depende de la plataforma de hardware sobre la cual está operando, es de vital importancia que un sistema altamente disponible, sea puesto en operación sobre una solución que incluya una plataforma que ofrezca el porcentaje de disponibilidad necesario para el mismo.

Proveer un adecuado nivel de disponibilidad a un sistema, es una actividad que requiere de planeación desde la concepción del mismo. Como mencionamos a lo largo de este trabajo, es casi imposible alcanzar el cien por ciento. Sin embargo, existen soluciones que en conjunto con otros factores inherentes al sistema (Procedimientos de operación, implementación de la red, ambiente de operación, software, servicios contratados, etc.) nos permitirán lograr niveles cercanos al 99.999%.

La tecnología cluster es una de las alternativas que nos permite incrementar el nivel de disponibilidad, además es una de las más económicas, de fácil implementación y prácticamente disponible en cualquier plataforma. Mientras más robusto se construya un cluster y su entorno, el porcentaje de disponibilidad se verá favorablemente elevado.

Además, la tecnología cluster ofrece la ventaja de ser escalable. Esto significa que su implementación podrá empezar de una forma muy simple (Cluster de dos nodos en un mismo centro de cómputo) y poco a poco podrá ir creciendo , haciendose más robusta hasta lograr por ejemplo una solución de disaster recovery (Cluster de varios nodos localizados en sites diferentes, los cuales pueden estar ubicados incluso en ciudades remotas).

El uso de los cluster como solución a la alta disponibilidad para sistemas informaticos de empresas de cualquier nivel es factible, ya que como se ha mencionado a lo largo del desarrollo de la tesis, la opción del cluster resulta altamente económica para la empresa y se le agrega el beneficio de que a medida que van creciendo las necesidades informáticas de los sistemas de la empresa, el cluster puede ir creciendo y actualizándose en función de las necesidades de la misma ya que debido a qué los componentes que usa el cluster en su estructura física son de tecnología estándar, reduce el costo y facilita su adaptabilidad, proporcionando con esto un mutuo crecimiento y una solución que beneficiará a la empresa durante un lapso de tiempo satisfactorio.

APENDICE A

```
# clu_create
This is the TruCluster Creation Program
You will need the following information in order to create a cluster:
- Cluster name (a hostname which is also used as the default cluster
alias)
- Cluster alias IP address
- Clusterwide root disk and partition (for example, dsk4b)
- Clusterwide usr disk and partition (for example, dsk4g)
- Clusterwide var disk and partition (for example, dsk4h)
- Quorum disk device (for example, dsk4)
- Number of votes assigned to the quorum disk
- Member ID
- Number of votes assigned to this member
- First member's boot disk (for example, dsk5)
- First member's virtual cluster interconnect IP name
- First member's virtual cluster interconnect IP address
- First member's physical cluster interconnect devices
- First member's NetRAIN device name
-  First member's physical cluster interconnect IP address


The program will prompt for this information, offering a default value
when one is available. To accept the default value, press Return.
If you need help responding to a prompt, either type the word 'help'
or type a question mark (?) at the prompt.
The program does not begin to create a cluster until you answer

all the prompts, and confirm that the answers are correct.
Cluster creation involves the following steps:
Labeling disks (when required)
Creating AdvFS domains
Copying the files on the current root, usr, and var
partitions to the clusterwide partitions
Creating additional CDSLs
Updating configuration files
Building a kernel and copying it to the first member's boot disk
After the kernel is built and copied, you will halt the system and boot
it using the first member's boot disk.
```

```
Do you want to continue creating the cluster? [yes]:
Each cluster has a unique cluster name, which is a hostname
used to identify the entire cluster.

Enter a fully-qualified cluster name []:zeus.compaq.com
Checking cluster name: zeus.compaq.com
You entered ' zeus.compaq.com' as your cluster name.
Is this correct? [yes]:
The cluster alias IP address is the IP address associated with the
default cluster alias. (192.168.168.1 is an example of an IP address.)
Enter the cluster alias IP address []:16.179.20.59
Checking cluster alias IP address: 16.179.20.59
You entered ' 16.179.20.59' as the IP address for the default cluster
alias.
Is this correct? [yes]:



The cluster root partition is the disk partition (for example, dsk4b)
that will hold the clusterwide root (/) file system.
Note: The default 'a' partition on most disks is not large
enough to hold the clusterwide root AdvFS domain.
Enter the device name of the cluster root partition []:dsk1b
Checking the cluster root partition: dsk1b
You entered ' dsk1b' as the device name of the cluster root partition.
Is this correct? [yes]:

The cluster usr partition is the disk partition (for example, dsk4g)
that will contain the clusterwide usr (/usr) file system.
Note: The default 'g' partition on most disks is usually
large enough to hold the clusterwide usr AdvFS domain.
Enter the device name of the cluster usr partition []:dsk1g
Checking the cluster usr partition: dsk1g
You entered 'dsk1g' as the device name of the cluster usr partition.
Is this correct? [yes]:

To use this default value, press Return at the prompt.
The cluster var device is the disk partition (for example, dsk4h)
that will hold the clusterwide var (/var) file system.
Note: The default 'h' partition on most disks is usually
large enough to hold the clusterwide var AdvFS domain.

Enter the device name of the cluster var partition [dsk3h]: dsk1h
Checking the cluster var partition: dsk1h
You entered 'dsk1h' as the device name of the cluster var partition.
Is this correct? [yes]:

Do you want to define a quorum disk device at this time? [yes]:n

The default member ID for the first cluster member is '1'.
To use this default value, press Return at the prompt.

A member ID is used to identify each member in a cluster.
Each member must have a unique member ID, which is an integer in
the range 1-63, inclusive.
Enter a cluster member ID [1]:1
Checking cluster member ID: 1
You entered '1' as the member ID.
```

```
Is this correct? [yes]:

By default the 1st member of a cluster is assigned '1' vote(s).
Checking number of votes for this member: 1


Each member has its own boot disk, which has an associated
device name; for example, 'dsk5'.
Enter the device name of the member boot disk []:dsk2
Checking the member boot disk: dsk2
You entered 'dsk2' as the device name of this member's boot disk.
Is this correct? [yes]:


Device 'ics0' is the default virtual cluster interconnect device.
Checking virtual cluster interconnect device: ics0
The virtual cluster interconnect IP name 'juliet-ics0' was formed by
appending '-ics0' to the system's hostname.
To use this default value, press Return at the prompt.
Each virtual cluster interconnect interface has a unique IP name (a
hostname) associated with it.


Enter the IP name for the virtual cluster interconnect [juliet-ics0]:
hera-ics0
Checking virtual cluster interconnect IP name: hera-ics0
You entered 'hera-ics0' as the IP name for the virtual cluster
interconnect.
Is this name correct? [yes]:


The virtual cluster interconnect IP address '10.0.0.2' was created by
replacing the last byte of the default virtual cluster interconnect
network
address '10.0.0.0' with the previously chosen member ID '2'.
To use this default value, press Return at the prompt.

The virtual cluster interconnect IP address is the IP address
associated with the virtual cluster interconnect IP name. (192.168.168.1
is an example of an IP address.)
Enter the IP address for the virtual cluster interconnect [10.0.0.2]:
Checking virtual cluster interconnect IP address: 10.0.0.2
You entered '10.0.0.2' as the IP address for the virtual cluster
interconnect.
Is this address correct? [yes]:

What type of cluster interconnect will you be using?
Selection Type of Interconnect
-------------------------------------------------------------------------
1 Memory Channel
2 Local Area Network
3 None of the above
4 Help
5 Display all options again
-------------------------------------------------------------------------
Enter your choice [1]:2
You selected option '2' for the cluster interconnect.
Is that correct? (y/n) [y]:
The physical cluster interconnect interface device is the name of the
physical device(s) which will be used for low level cluster node
communications. Examples of the physical cluster interconnect interface
device name are: tu0, ee0, and nr0.
```

```
Enter the physical cluster interconnect device name(s) []:tu0
Would you like to place this Ethernet device into a NetRAIN set?
[yes]:n

Checking physical cluster interconnect interface device name(s): tu0
You entered 'tu0' as your physical cluster interconnect interface
device name(s). Is this correct? [yes]:


The physical cluster interconnect IP name 'member1-icstcp0' was formed
by
appending '-icstcp0' to the word 'member' and the member ID.
Checking physical cluster interconnect IP name: member1-icstcp0
The physical cluster interconnect IP address '10.1.0.1' was created by
replacing the last byte of the default cluster interconnect network
address
'10.1.0.0' with the previously chosen member ID '1'.
To use this default value, press Return at the prompt.
The cluster physical interconnect IP address is the IP address
associated with the physical cluster interconnect IP name.
(192.168.168.1) is an example of an IP address.)
Enter the IP address for the physical cluster interconnect [10.1.0.1]:
Checking physical cluster interconnect IP address: 10.1.0.1
You entered '10.1.0.1' as the IP address for the physical cluster
interconnect.
Is this address correct? [yes]:


You entered the following information:
Cluster name: zeus
Cluster alias IP Address: 16.179.20.59
Clusterwide root partition: dsk1b
Clusterwide usr partition: dsk1g
Clusterwide var partition: dsk1h
Clusterwide i18n partition: Directory-In-/usr
Quorum disk device: Not-Selected
Number of votes assigned to the quorum disk: Not-Selected
First member's member ID: 1
Number of votes assigned to this member: 1
First member's boot disk: dsk2
First member's virtual cluster interconnect device name: ics0
First member's virtual cluster interconnect IP name: hera-ics0
First member's virtual cluster interconnect IP address: 10.0.0.1
First member's physical cluster interconnect devices tu0
First member's NetRAIN device name Not-Applicable
First member's physical cluster interconnect IP address 10.1.0.1
If you want to change any of the above information, answer 'n' to the
following prompt. You will then be given an opportunity to change your
selections.
Do you want to continue to create the cluster? [yes]:
Creating required disk labels.
Creating disk label on member disk : dsk2
Initializing cnx partition on member disk : dsk2h
Creating AdvFS domains:
Creating AdvFS domain 'root2_domain#root' on partition
'/dev/disk/dsk2a'.
Creating AdvFS domain 'cluster_root#root' on partition
      '/dev/disk/dsk1b'.
Creating AdvFS domain 'cluster_usr#usr' on partition '/dev/disk/dsk1g'.
Creating AdvFS domain 'cluster_var#var' on partition '/dev/disk/dsk1h'.
Populating clusterwide root, usr, and var file systems:
Copying root file system to 'cluster_root#root'.....
```

```
Copying usr file system to 'cluster_usr#usr'.

.....................................................................
Copying var file system to 'cluster_var#var'.
......
Creating Content Dependent Symbolic Links (CDSLs) for file systems:
Creating CDSLs in root file system.
Creating CDSLs in usr file system.
Creating CDSLs in var file system.
Creating links between clusterwide file systems
Populating member's root file system.
Modifying configuration files required for cluster operation:
Creating /etc/fstab file.
Configuring cluster alias.
Updating /etc/hosts - adding IP address '16.179.20.58' and hostname
'hera.compaq.com'
Updating member-specific /etc/inittab file with 'cms' entry.
Updating /etc/hosts - adding IP address '10.0.0.2' and hostname 'hera-
ics0'
Updating /etc/hosts - adding IP address '10.1.0.2' and hostname
'member2-icstcp0'
Updating /etc/rc.config file.
Updating /etc/sysconfigtab file.
Retrieving cluster_root major and minor device numbers.
Creating cluster device file CDSLs.
Updating /.rhosts - adding hostname 'hera'.
Updating /etc/hosts.equiv - adding hostname 'hera'
Updating /.rhosts - adding hostname 'hera-ics0'.
Updating /etc/hosts.equiv - adding hostname 'hera-ics0'
Updating /.rhosts - adding hostname 'member2-icstcp0'.
Updating /etc/hosts.equiv - adding hostname 'member2-icstcp0'
Updating /etc/ifaccess.conf - adding deny entry for 'ln0'
Updating /etc/ifaccess.conf - adding deny entry for 'sl0'
Updating /etc/ifaccess.conf - adding deny entry for 'tu0'
Updating /etc/ifaccess.conf - adding deny entry for 'tun0'
Updating /etc/ifaccess.conf - adding deny entry for 'ln0'
Updating /etc/ifaccess.conf - adding deny entry for 'sl0'
Updating /etc/ifaccess.conf - adding deny entry for 'tu0'
Updating /etc/ifaccess.conf - adding deny entry for 'tun0'
Updating /etc/cfgmgr.auth - adding hostname 'hera'
Finished updating member2-specific area.
Building a kernel for this member.
Saving kernel build configuration.
The kernel will now be configured using the doconfig program.
Warning: File in /usr/sys/BINARY found as a file, expected symlink:
GENERIC.mod
Warning: File in /usr/sys/BINARY found as a file, expected symlink:
GENERIC_EXTRAS.mod
*** KERNEL CONFIGURATION AND BUILD PROCEDURE ***
Saving /sys/conf/HERA as /sys/conf/HERA.bck
*** PERFORMING KERNEL BUILD ***
Working....Fri Jul 12 15:36:55 EDT 2001
Working....Fri Jul 12 15:38:57 EDT 2001
Working....Fri Jul 12 15:41:00 EDT 2001
The new kernel is /sys/HERA/vmunix
Finished running the doconfig program.
The kernel build was successful and the new kernel
has been copied to this member's boot disk.
Restoring kernel build configuration.
Updating console variables
Setting console variable 'bootdef_dev' to dsk2
Setting console variable 'boot_dev' to dsk2
Setting console variable 'boot_reset' to ON
Saving console variables to non-volatile storage
```

```
clu_create: Cluster created successfully.
To run this system as a single member cluster it must be rebooted.
If you answer yes to the following question clu_create will reboot the
system for you now. If you answer no, you must manually reboot the
system after clu_create exits.
Would you like clu_create to reboot this system now? [yes]:
Shutdown at 10:48 (in 0 minutes) [pid 23371]
```

```
# clu_add_member
This is the TruCluster Add Member Program
You will need the following information in order to add a member to
the cluster:
- Hostname
- Member ID (1-63)
- Members Votes
- Member's boot disk (for example, dsk7)
- Member's virtual cluster interconnect IP name
- Member's virtual cluster interconnect IP address
- Member's physical cluster interconnect devices
- Member's NetRAIN device name
- Member's physical cluster interconnect IP address
- Member's cluster license
The program will prompt for this information, offering a default
value when one is available. To accept the default value, press Return
If you need help responding to a prompt, either type the word 'help'
or type a question mark (?) at the prompt.
The program does not begin to add the member until you answer
all the prompts, and you confirm that the answers are correct.
Adding a member involves the following steps:
Labeling the boot disk (when required)
Creating AdvFS domains
Creating additional CDSLs
Updating configuration files
You then boot genvmunix from the new member's boot disk. At the first
boot the new member:
Configures layered product subsets
Builds a kernel and copies it to the member's boot disk
Boots the new kernel
Do you want to continue adding this member? [yes]:
Each cluster member has a hostname, which is assigned to the HOSTNAME
variable in /etc/rc.config.

Enter the new member's fully qualified hostname []:juno.compaq.com
Checking member's hostname: juno.compaq.com
You entered 'juno.compaq.com' as this member's hostname.
Is this name correct? [yes]:

The next available member ID for a cluster member is '2'.
To use this default value, press Return at the prompt.
A member ID is used to identify each member in a cluster.
Each member must have a unique member ID, which is an integer in
the range 1-63, inclusive.
Enter a cluster member ID [2]:
Checking cluster member ID: 2
You entered '2' as the member ID.
Is this correct? [yes]:

By default, when the current members expected votes is less then or
```

```
equal to 1, each added member is assigned '0' vote(s).
To use this default value, press Return at the prompt.
The number of votes for a member is an integer usually 0 or 1
Enter the number of votes for this member [0]: 1
Checking number of votes for this member: 1
You entered '1' as the number votes for this member.
Is this correct? [yes]:


Each member has its own boot disk, which has an associated
device name; for example, 'dsk5'.
Enter the device name of the member boot disk []:dsk3
Checking the member boot disk: dsk3
You entered 'dsk3' as the device name of this member's boot disk.
Is this correct? [yes]:


Device 'ics0' is the default virtual cluster interconnect device
Checking virtual cluster interconnect device: ics0
The virtual cluster interconnect IP name 'romeo-ics0' was formed by
appending '-ics0' to the system's hostname.
To use this default value, press Return at the prompt.
Each virtual cluster interconnect interface has a unique IP name (a
hostname) associated with it.
Enter the IP name for the virtual cluster interconnect [romeo-ics0]:
juno-ics0
Checking virtual cluster interconnect IP name: juno-ics0
You entered ' juno-ics0' as the IP name for the virtual cluster
interconnect.
Is this name correct? [yes]:

The virtual cluster interconnect IP address '10.0.0.2' was created by
replacing the last byte of the virtual cluster interconnect network
address
'10.0.0.0' with the previously chosen member ID '2'.
To use this default value, press Return at the prompt.

The virtual cluster interconnect IP address is the IP address
associated with the virtual cluster interconnect IP name. (192.168.168.1
is an example of an IP address.)
Enter the IP address for the virtual cluster interconnect [10.0.0.2]:
Checking virtual cluster interconnect IP address: 10.0.0.2
You entered '10.0.0.2' as the IP address for the virtual cluster
interconnect.
Is this address correct? [yes]:

The physical cluster interconnect interface device is the name of the
physical device(s) which will be used for low level cluster node
communications. Examples of the physical cluster interconnect interface
device name are: tu0, ee0, and nr0.
Enter the physical cluster interconnect device name(s) []:tu0
Would you like to place this Ethernet device into a NetRAIN set?
[yes]:n
Checking physical cluster interconnect interface device name(s): tu0
You entered 'tu0' as your physical cluster interconnect interface
device name(s). Is this correct? [yes]:

The physical cluster interconnect IP name 'member2-icstcp0' was formed
by appending '-icstcp0' to the word 'member' and the member ID.
Checking physical cluster interconnect IP name: member2-icstcp0
The physical cluster interconnect IP address '10.1.0.2' was created by
replacing the last byte of the physical cluster interconnect network
address '10.1.0.0' with the previously chosen member ID '2'.
To use this default value, press Return at the prompt.
```

```
The cluster physical interconnect IP address is the IP address
associated with the physical cluster interconnect IP name.
(192.168.168.1is an example of an IP address.)
Enter the IP address for the physical cluster interconnect [10.1.0.2]:
Checking physical cluster interconnect IP address: 10.1.0.2
You entered '10.1.0.2' as the IP address for the physical cluster
interconnect.
Is this address correct? [yes]:

Each cluster member must have its own registered TruCluster Server
license. The data required to register a new member is typically located
on the License PAK certificate or it may have been previously placed on
your system as a partial or complete license data file. If you are
prepared to enter this license data at this time, clu_add_member can
configure the new member to use this license data. If you do not have
the license data at this time you can enter this data on the new member
when it is up and running.
Do you want to register the TruCluster Server license for this new
member at
this time? [yes]:n
You entered the following information:
Member's hostname: juno.compaq.com
Member's ID: 1
Number of votes assigned to this member: 1
Member's boot disk: dsk3
Member's virtual cluster interconnect devices: ics0
Member's virtual cluster interconnect IP name: juno-ics0
Member's virtual cluster interconnect IP address: 10.0.0.2
Member's physical cluster interconnect devices: tu0
Member's NetRAIN device name: Not-Applicable
Member's physical cluster interconnect IP address: 10.1.0.2
Member's cluster license: Not Entered
If you want to change any of the above information answers 'n' to the
following prompt. You will then be given an opportunity to change your
selections.
Do you want to continue to add this member? [yes]:
Creating required disk labels.
Creating disk label on member disk : dsk3
Initializing cnx partition on member disk : dsk3h
Creating AdvFS domains:
Creating AdvFS domain 'root1_domain#root' on partition
'/dev/disk/dsk3a'.
Creating cluster member-specific files:
Creating new member's root member-specific files
Creating new member's usr member-specific files
Creating new member's var member-specific files
Creating new member's boot member-specific files
Modifying configuration files required for new member operation:
Updating /etc/hosts - adding IP address '10.0.0.2' and hostname 'juno-
ics0'
Updating /etc/hosts - adding IP address '10.1.0.2' and hostname
'member2-icstcp0'
Updating /etc/rc.config
Updating /etc/sysconfigtab
Updating member-specific /etc/inittab file with 'cms' entry.
Updating /.rhosts - adding hostname 'juno-ics0'
Updating /etc/hosts.equiv - adding hostname 'juno-ics0'
Updating /.rhosts - adding hostname 'member2-icstcp0'
Updating /etc/hosts.equiv - adding hostname 'member2-icstcp0'
Updating /etc/cfgmgr.auth - adding hostname 'juno.compaq.com'
Configuring cluster alias.
Configuring Network Time Protocol for new member
Adding interface 'juno-ics0' as an NTP peer to member 'juno.compaq.com'
Adding interface 'juno-ics0' as an NTP peer to member 'juno.compaq.com'
```

```
Configuring automatic subset configuration and kernel build.
clu_add_member: Initial member 2 configuration completed successfully.
From the newly added member's console, perform the following steps to
complete the newly added member's configuration:
1. Set the console variable 'boot_osflags' to 'A'.
2. Identify the console name of the newly added member's boots device.
>>>show device
The newly added member's boot device has the following properties:
Manufacturer: DEC
Model: RZ28 (C) DEC
Target: 4
Lun: 0
Serial Number: SCSI-WWID:0410003a:"DEC RZ28 (C) DECPCB=ZG52462664
; HDA=000041563084"
Note: The SCSI bus number may differ when viewed from different members.
3. Boot the newly added member using genvmunix:
>>>boot -file genvmunix <new-member-boot-device>
During this initial boot the newly added member will:
o Configure each installed subset.
o Attempt to build and install a new kernel. If the system cannot
build a kernel, it starts a shell where you can attempt to build
a kernel manually. If the build succeeds, copy the new kernel to
/vmunix. When you are finished exit the shell using ^D or 'exit'.
o The newly added member will attempt to set boot related console
variables and continue to boot to multi-user mode.
o After the newly added member boots you should setup your system
default network interface using the appropriate system management
command.
```

APÉNDICE B

PLANEACIÓN DE LA CAPACIDAD (CAPACITY PLANNING)

La planeación de la capacidad (capacity planning) se define como un proceso cíclico que su función es la de prevenir los requerimientos a futuro que el sistema pueda necesitar para soportar las aplicaciones y recursos para su pronta respuesta en los negocios. Es importante aclarar que la planeación de la capacidad utiliza una metodología pro activa lo cual quiere decir que esta se va a ir modificando con el tiempo si el sistema o las necesidades de la empresa así lo requiere, es decir, es una metodología flexible para soportar cambios que puedan garantizar en un futuro la disponibilidad y el óptimo funcionamiento de los sistemas de la empresa.

Básicamente la planeación de la capacidad da la pauta para poder diseñar un sistema que pueda ir creciendo junto con las necesidades de la empresa con el paso del tiempo.

A medida que los sistemas de una empresa van siendo utilizados se hace más notable la necesidad de actualizarlos para satisfacer las necesidades de

crecimiento que la empresa requiera, claramente, con el crecimiento de los negocios de la misma empresa sus sistemas serían insuficientes por el incremento de sus actividades comerciales y procesos.

Los puntos importantes que se deben de tomar en cuenta en el momento de crear un sistema son:

- Planeación y control de la red.
- Adquisición de un buen equipo de computo capaz de soportar las necesidades del negocio.
- Predicción del impacto en el momento de ingresar una nueva aplicación al sistema.
- Mantener un nivel fiable del servicio mediante mantenimiento del sistema, balanceo de carga de trabajo y nuevas configuraciones del mismo.

En la actualidad una empresa no puede ignorar este tipo de recomendaciones puesto que de no aplicarlas a sus sistemas de negocios, éstos irán decayendo siendo insuficientes para satisfacer las necesidades crecientes que los negocios de la empresa requieran.

La Figura B-1 muestra los aspectos a considerar en una planeación y como se relacionan entre si.

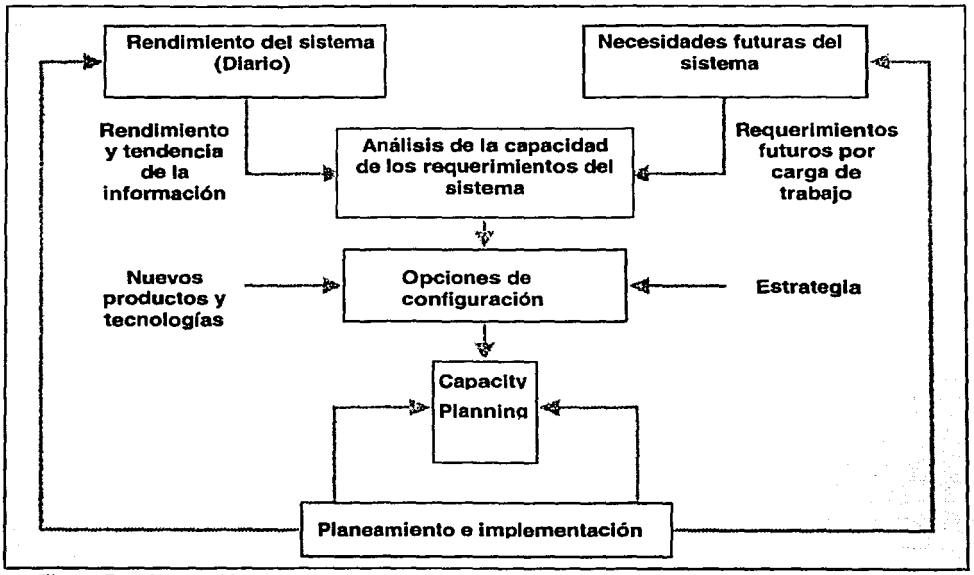


Figura B-1 Planeación de la capacidad.

Guía para la Planeación de la Capacidad

Para una buena instalación de una aplicación de red, se requiere de una planeación cuidadosa. A continuación se mencionan algunos aspectos que se deben tomar en cuenta en dicha planeación.

- Se debe tener suficiente poder de procesamiento para manejar el número de transacciones de usuarios estimado. Entre más alto sea el desempeño del procesador, mayor será el número conexiones de usuarios que podrá soportar la aplicación.

- Se debe tener suficiente memoria en el sistema para reducir o eliminar las operaciones de *swapping*.
- Verificar que el ancho de banda proporcionado es suficiente para manejar las transacciones esperadas. Actualmente una conexión de 10 Mbps es insuficiente para aplicaciones de misión crítica.
- Usar sistemas RAID en los sistemas de almacenamiento, ya que proporcionan una mayor integridad de la información.

GLOSARIO

-A-

AVAILABILITY: Accesibilidad del sistema de una computadora o red.

-B-

BACKUP/RESTORE: El acto de copiar archivos y bases de datos para proteger la información en caso de fallas o catástrofes y posteriormente regresarlos a su estado normal en una fecha posterior.

BUS: El canal de transmisión de una computadora o en una red que lleva las señales y dispositivos al canal.

BUSINESS CONTINUANCE: Es la implementación de técnicas de redundancia y tolerancia a fallas utilizando equipos y software específicos que garantizan la continuidad del negocio a pesar de errores naturales o humanos.

-C-

CAM: Common Access Method, El método común de acceso es un estándar ANSI implementado en DEC OSF/1 para estandarizar la operación en los dispositivos SCSI.

CHANNEL: Una conexión con gran capacidad de transferencia de datos entre un procesador y otros procesadores o equipos.

CLUSTER: Conjunto de servidores de alto desempeño interconectados entre sí, trabajando en un ambiente sencillo de procesamiento para proveer mayor escalabilidad, alta disponibilidad para los usuarios y para las aplicaciones.

CONTROL DE ACCESO: Término general que se utiliza para limitar el acceso a computadoras o redes solamente a usuarios autorizados con passwords o tarjetas inteligentes.

-D-

DEVICE: Subsistemas de computadoras como impresoras, discos magnéticos, adaptadores y monitores que normalmente necesitan de programas específicos de control (device drivers) para comunicarse con el sistema de computadoras.

DISASTER RECOVERY: Plano preventivo para garantizar que las operaciones críticas de los negocios pueden continuar su funcionamiento, y ser inmediatamente recuperadas en caso de accidentes (naturaleza o falla humana).

DOWNTIME: Tiempo en el cual un sistema se mantiene fuera de operación, ya sea por una falla imprevista o por una actividad programada.

-F-

FABRIC: Topología del Canal de Fibra con un dispositivo de switch.

FAILBACK : Recuperación de los recursos de un dispositivo al cual se le corrige una falla.

FAILOVER: En caso de una falla en los dispositivos como adaptadores, cables, unidades de administración de canales y otros, los datos son desviados inmediatamente por un canal alternativo, antes de que ocurra una interrupción del trabajo.

FAULT TOLERANCE: Habilidad de un sistema de cómputo para responder a un accidente, como fallas de energía o falla del equipo de forma que los datos no sean perdidos o afectados

FIBRE CHANNEL (FC): Interfaz con capacidad de transferencia de datos de un Gigabit por segundo, la especificación permite la transferencia de 133 Megabit por segundo arriba de 4.25 Gigabit por segundo. Los Datos pueden ser transmitidos y recibidos simultáneamente a tasa de un Gigabit por segundo. Los protocolos más comúnmente utilizados - Internet Protocol (IP) e Small Computer System Interface (SCSI) - trabajan con Fibra Óptica. Como consecuencia, las operaciones de entrada/salida de datos de alta velocidad y la propia red se pueden beneficiar de una tecnología única de conectividad.

-H-

HARD DISK: Dispositivo para almacenamiento de grandes cantidades de datos que consiste en un aparato herméticamente sellado, compuesto de un conjunto de discos magnéticos, mecanismos de rotación y cabezas de lectura y grabación.

HARDWARE RAID: Procesadores duales que aumentan la disponibilidad, desempeño y transferencia de información con protección para los discos magnéticos. Localizados externamente en el subsistema de almacenamiento de datos o procesador central (CPU) para tareas como paridad del RAID, segmentamiento y cálculos de reconstrucción de datos. Los circuitos inteligentes son manejados en el board de los drives del disco.

HOST: Servidor en red que procesa típicamente aplicaciones usadas por otras computadoras (Ejemplo: web servers, file servers, and application servers).

HUB: Dispositivo que consolida el enlace de líneas de comunicación, permitiendo una conexión única de todos los equipos de una red.

-J-

JOURNALY: Procedimiento mediante el cual se intenta mantener los datos de un sistema de cómputo actualizados, de tal manera que cuando se presente una falla el tiempo de recuperación del ultimo estado de la información se minimice al máximo.

-L-

LINK: Conexión existente entre dos adaptadores de Fibra.

LUN (Logical Unit Number): Código interno de identificación de 3 bits usado para arquitectura SCSI y para diferenciar entre 8 dispositivos (unidades lógicas) con el mismo SCSI ID. Un equipo de almacenamiento para el procesador central de una LUN. Cada conjunto de discos magnéticos pertenecientes a un RAID posee una LUN por lo que el host es accesado o direccionado por la información de esos componentes.

-M-

MAINFRAME: Computadora utilizada principalmente por grandes empresas para aplicaciones comerciales de gran escala. Esta computadora tiene la capacidad de soportar un gran volumen de usuarios simultáneamente.

-O-

OVERHEAD: Recursos consumidos por procesos que son incidentales pero necesarios para realizar el proceso principal.

-P-

PING : Programa que nos indica el tiempo exacto que tarda un paquete de datos en ir y volver a través de la red desde nuestra PC a un determinado servidor remoto.

PROTOCOLO: Conjunto de reglas elaboradas para hacer posible la comunicación entre computadoras.

PUERTO: En una computadora, es el punto de conexión física donde es conectado el equipo.

-R-

RAID (Redundant Array of Independent Disks): La información es almacenada en múltiples discos magnéticos o discos ópticos para aumentar el desempeño, y la capacidad de almacenamiento y para ofrecer niveles diferentes de redundancia y tolerancia a fallas. En lugar de Almacenar información en un único disco que puede fallar en cualquier momento, el RAID garantiza que una copia de la información siempre exista, a través de la distribución de datos entre múltiples discos.

REDUNDANTE: Implementación dual de discos, arrays, drives o fuentes de poder que posibilitan el procesamiento redundante de las funciones.

RELOCALIZACION: Migración de un servicio que se encuentra en un servidor a otro, por un comando de administración.

ROBUSTO: Capacidad para funcionar bien, en caso de situaciones no anticipadas.

-S-

SCSI: Conjunto de protocolos para que las computadoras - servidores puedan comunicarse con periféricos. SCSI permite conexiones hasta 6 periféricos, incluyendo impresoras, scanners, discos, zip drives y CD-ROM.

SCSI BUS: Canal paralelo que lleva datos y control de señales de unidades SCSI de control SCSI.

SHELF DE DISCOS: Nombre que usa en inglés, para hacer referencia del uso de una caja de discos.

SOFTWARE RAID: Procesadores - servidores para hacer cálculos RAID, evitando que las aplicaciones tengan ciclos de procesamiento de operación de lectura y escritura. El Software RAID es más barato que el hardware RAID pero la protección de la información es menos eficiente y escalable.

SWITCH: Componente de red que selecciona un camino o canal para el envío de información entre los diferentes destinos.

-T-

TARGET : Dispositivo SCSI que ejecuta líneas de comando para iniciar un dispositivo SCSI.

-U-

UPTIME : Tiempo en el cual un sistema se mantiene en operación, es decir que las aplicaciones y procesos se encuentran en un estado de procesamiento normal.

-V-

VOTADOR : Dispositivos de varias entradas por donde se transmite información redúndate enmascara las fallas y tiene una única salida, se comporta semejante a una compuerta OR.

-W-

WORKLOAD BALANCING: Metodología que garantiza que ninguna información puede ser sobrecargada cuando otros están sub-utilizados, causando problemas en operaciones de entrada/salida. Cuando existen caminos más ocupados que otros, el tráfico de entrada/salida y transferencias para otros caminos, aumentando la capacidad de efectuar operaciones de entrada/salida. Adicionalmente

WRITE-CACHE: Técnica que almacena temporalmente los datos de la memoria cache antes de ser grabados permanentemente en el disco. Esta técnica aumenta el desempeño general del sistema, disminuyendo el tiempo de procesar para leer o grabar datos en un disco.

WRITE-MODE: Momento en que un programa puede grabar información en un archivo. En el modo de escritura el usuario es permitido para hacer cambios en la información existente.

BIBLIOGRAFÍA

Barros, Alejandro. "Arquitectura de Alta Disponibilidad Para Ambientes de negocios". enable, S.A. Colombia. Octubre 2001.

Compaq Computer Corporation. *Course Guide: AdvFS, LSM, and RAID Configuration and Management* (EY-X776E-SG.0002). Houston, Texas. June 2000.

Compaq Computer Corporation. *Course Guide: TruCluster Server Configuration and Management* (CS879A-ESG). Houston, Texas. November 2001.

Compaq Computer Corporation. Software Product Description, Tru64 UNIX Logical Storage Manager Version 5.0A. April 2000.

Compaq Computer Corporation. *TruCluster Software Products: Administration* (Part Number: AA-R88JB-TE). Houston, Texas. April 1999.

Compaq Computer Corporation. *TruCluster Software Products: Hardware Configuration* (Part Number: AA-R88GA-TE). Houston, Texas. January 1998.

Compaq Computer Corporation. *TruCluster Software Products: Software Installation* (Part Number: AA-R88HA-TE). Houston, Texas. January 1998.

Compaq Computer Corporation. *TruCluster Software Products: Release Notes* (Part Number: AA-R0JAC-TE). Houston, Texas. January 1998.

Compaq Computer Corporation. *TruCluster Server, Cost Of Administration by Streamlining Operations.* NY, USA. June 2000.

Compaq Computer Corporation. *TruCluster Server, High Available Applications* (Part Number: AA-RHH0C-TE). Houston, Texas. August 2000.

Compaq Computer Corporation. *TruCluster Server, Technical Overview* (Part Number: AA-RHGVC-TE). Houston Texas. August 2000.

Compaq Computer Corporation, UNIX Software Division. *Cluster File System in Compaq, TrueCluster Server.* Houston, Texas. September 2001.

Day, Brad. "Trends in Cluster Architectures". Giga Information Group. Cambridge, MA. December 1998.

Digital Equipment Corporation. *DIGITAL UNIX TruCluster Software Versión 1.4: Customer Businessman Needs.* March 1997.

Digital Equipment Corporation. *DIGITAL UNIX TruCluster Software Versión 1.4: Technology Response.* March 1997.

Digital Equipment Corporation. *VAX open VMS at 20.* 1998. USA.

D.H. Brown Associates, Inc. *Competitive Analysis of UNIX Cluster functionality – Part One of a Two-Part HA Study.* March 2000.

D.H. Brown Associates, Inc. *Single-System High Availability - Part-Two of a Two-Part Study-*. July 2001.

D. H. Brown Associates, Inc. *TruCluster Server 5.0 Slashes, Cost of Administration by streamlining Operation.* Port Chester, NY. June 2000.

Douglas, Frank. "DECsafe ASE, Theory of Operation". Unix Software Group. Nashua, NH. 2001.

Ferguson, Adrian. "Idea Byte: How Available Is Your Server?". Giga Information Group.

Gil Vicente, Pedro Joaquín, Serrano Martín Juan José. "Grupo de Sistema Tolerante a Fallos (GSTF)". Universidad Politécnica de Valencia. Valencia, España.

Harvard Research Group. *Hewlett-Packard and Marathon Technologias Offering New Assured Availability (AE-4) Solutions for NT.* Harvard, Ma. 01451 USA.

IBM Global Services, Information Development Center. *Improving Systems Availability.* Atlanta, GA 30339 USA. 1999.

Lawrence S. Cohen, John H. Williams. "Technical Description of the DECsafe Available Server Environment". *Digital Technical Journal.* Volumen 7 No. 4 1995.

Medinets David. Herramientas de Programación para el Shell de UNIX. Editorial McGraw-Hill. México, D. F. 2001.

Microsoft Corporation, High Availability Operations Guide, Implementing System for Reliability and Availability. Redmond, WA 98052-6399, USA. 1999.

Mission Critical Linux, Inc. *Kimberlite Cluster*, "Mission Critical Linux, Inc. brings the features of expensive, proprietary software to the Linux community". Lowell, MA 01852.

Pérez, Miguel Ángel. "Arquitecturas Paralelas I, II". Networking Center. Mayo 2001.

Pramanik, Ira. "High Availability (HA)". I*EEE CS Task Force on Cluster Computing (TFFCC).*

Buyya Rajkumar. High Performance Cluster Computing: *Arquitecture and System, Volumen 1.* School of Computer Science and Software Engineering, Monash University. Melbourne, Australia. Editorial Prentice Hall PTR. 1999.

Scott, D. "High Availability Q&A: Failure, Standards and Metrics". Research Note Select Q&A. November 1998.

TechWise Research, Inc. *Total Cost of Ownership for Enterprise Class Cluster*, San Diego, Cal. January 2002.

UNISYS. *Calculating Availability for applications –A New Approach.* December 1998.

UNISYS. *Understading Enterprise-Class NT Clustering.* Spring 1999.

Walt Zajac. *Student Guide: Guardian Architecture And Service.* System Support Education. Tandem Gas Class.

Yudith Cardinale. "Diplomado en Memoria Compartida Distribuida". Universidad Simón Bolívar, Departamento de Computación. Caracas, Venezuela. Sep-Dic. 2001.

PÁGINAS WEB

http://iceman.networking-center.org

http://monografias.com/trabajos3/sais/sais.shtml

http://telematica.cicese.mx/computo/super/paralelo/Part3.html

http://yesca.alcd.net/cluster/

http://www.aui.es/biblio/libros/mi99/5seguridad_integral.htm

http://www.compaq.com

http://www.compaq.com/hpc/software/chan.html

http://www.enable.cl

http://www.gigaweb.com

http://www.hrgresearch.com

http://www.linuxfocus.org/Castellano/November2000/article179.shtml

http://www.linux-ha.org

http://www.microsoft.com/technet

http://www.missioncriticallinux.com

http://www.monografias.com/Computacion/Software/

http://www.sysinternals.com

http://www.sresearch.com

http://www.techrepublic.com

http://www.true64unix.compaq.com/unix/reliabilityvsavailability.htm

http://www.w2000mag.com/atrasados/1997/12_sept97/Revista/Art12.htm

http://www.windowstimag.com/atrasados/2001/50_feb01/articulos/especial.htm