Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

INFRAESTRUCTURA DE COMERCIO ELECTRÓNICO EN ALTA DISPONIBILIDAD

Raul Alembert Veliz Rodriguez

Asesorado por: Ing. José Christian Bradna Villanueva

Guatemala, octubre 2004

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

INFRAESTRUCTURA DE COMERCIO ELECTRÓNICO EN ALTA DISPONIBILIDAD

TRABAJO DE GRADUACIÓN

PRESENTADO A JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

RAUL ALEMBERT VELIZ RODRIGUEZ

ASESORADO POR ING. JOSÉ CHRISTIAN BRADNA VILLANUEVA

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, OCTUBRE 2,004

HONORABLE TRIBUNAL EXAMINADOR

Cumpliendo con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

INFRAESTRUCTURA DE COMERCIO ELECTRÓNICO EN ALTA DISPONIBILIDAD

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas con fecha enero de 2003.

_____

Raul Alembert Veliz Rodriguez

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	ING. SYDNEY ALEXANDER SAMUELS MILSON
VOCAL I	ING. MURPHY OLYMPO PAIZ RECINOS
VOCAL II	LIC. AMAHÁN SÁNCHEZ ALVAREZ
VOCAL III	ING. JULIO DAVID GALICIA CELADA
VOCAL IV	BR. KENNETH ISSUR ESTRADA RUIZ
VOCAL V	BR. ELISA YAZMINDA VIDES LEIVA
SECRETARIO	ING. PEDRO ANTONIO AGUILAR POLANCO

TRIBUNAL QUE PRÁCTICO EL EXAMEN GENERAL PRIVADO

DECANO	ING. JULIO I. GONZALEZ PODSZUECK
EXAMINADOR	ING. JORE LUIS ALVAREZ
EXAMINADOR	ING. FRANCISCO GUEVARA
EXAMINADOR	ING. HOSWALD BLANCO SUCHITE
SECRETARIO	ING. FRANCISCO J. GONZALEZ LOPEZ

DEDICO EL ACTO A

Dios

Mis padres

Mi esposa

Mis hijos

Mis hermanos

Mis tíos

Mis primos

Mi familia

Universidad de San Carlos de Guatemala

AGRADECIMIENTO A

El ingeniero José Christian Bradna Villanueva por la orientación y asesoría académica que me brindó durante la elaboración del presente trabajo.

Al ingeniero Carlos Azurdia, por su colaboración en la revisión de tesis.

A la licenciada Zuigly Sol Rodríguez Torres de Lam por su ayuda, dedicación y constancia para que llegara a la culminación de este trabajo.

A la señora Wilna Noemí Rodríguez Torres, Mi madre, por su constante ayuda, dedicación, y perseverancia para llevar a la culminación mis estudios. Una madre abnegada a quién admiro y respeto.

ÍNDICE GENERAL

INDICE DE ILUSTRACIONES X
GLOSARIO XIV
RESUMEN XVII
OBJETIVOS XIX
INTRODUCCIÓN XX
1. CONCEPTOS GENERALES 1
1.1. Servicios comercio electrónico 1
1.1.1. Alta disponibilidad 2
1.1.1.1. Nivel de disponibilidad 4
1.1.2. Escalabilidad 7
1.1.3. Seguridad 9
1.2. Arquitecturas de comercio electrónico 12
1.2.1. Arquitectura de un sitio de comercio electrónico 12
1.2.2. Arquitectura de comercio electrónica con múltiples sitios 13
2. ALTA DISPONIBILIDAD EN LA CAPA DE RED 17
2.1. Balanceador geográfico de carga 18
2.1.1. Cisco distributed director 2500 19
2.1.1.1. Servicios cisco distributed director 20
2.1.1.2. Métricas 25
2.1.1.3. Disponibilidad de servidores 27
2.2. Ruteador borde 27
2.2.1. Ruteadores Cisco 7500 28
2.2.1.1. Redundancia de procesador (RPR) 30
2.2.1.2. Procesador de redundancia + (RPR+) 31
2.2.1.3. Actualización rápida de software 31

2.2.1.4. Carga de tarjetas individuales (SLCR) 31
2.2.1.5. Stateful switchover (SSO) 32
2.2.1.6. Non-stop forwarding (NSF) 32
2.2.2. Hot standby router protocol 33
2.2.2.1. Operación HSRP 33
2.3. Caché de contenido 34
2.3.1. Tolerancia y prevención de fallas 35
2.4. Switch múltiples capas 37
2.4.1. Redundancia switch fabric 38
2.4.2. Supervisor redundante 39
2.4.3. Características de alta disponibilidad del supervisor 40
2.4.3.1. Redundancia de protocolo stateful en el supervisor 41
2.4.3.2. Actualización de las imágenes de software en el supervisor 42
2.4.4. Protocolo spanning tree 43
2.5. Firewall 46
2.5.1. ¿En qué consiste el failover? 46
2.5.2. Stateful Failover 47
2.5.3. Características del stateful failover 48
2.5.4. Requerimientos de hardware y software 49
2.6. Balanceador de carga para servidores 50
2.6.1. Cisco LocalDirector 51
2.6.1.1. Monitoreo de la negociación TCP 51
2.6.1.2. Verificación de contenido 53
2.6.1.3. Protocolo de retroalimentación dinámico 55
2.6.2. LocalDirector failover 56
3. ALTA DISPONIBILIDAD EN LOS SERVIDORES 59
3.1. Componentes un Servidor 59
3.1.1. Procesador 59

3.1.2. Chipsets 60
3.1.3. Buses de entrada/salida 61
3.1.4. Ventiladores 63
3.1.5. Fuentes de poder 63
3.1.6. Almacenamiento 64
3.2. Tecnología para protección de memoria 65
3.2.1. Errores de memoria 65
3.2.1.1. Errores de un solo bit y múltiples bits 66
3.2.1.2. Errores duros y suaves 67
3.2.1.3. Incremento en la probabilidad de errores de memoria 67
3.2.2. Métodos para prevenir errores de memoria 68
3.2.2.1. Tecnologías de detección/corrección 68
3.2.2.2. Protección avanzada de memoria 71
3.3. Tecnología para almacenamiento de datos 75
3.3.1. Controladores SCSI para alta disponibilidad 76
3.3.1.1. Guías para seleccionar una tarjeta controladora SCSI 76
3.3.1.2. Ventajas y desventajas de seleccionar una controladora SCSI 77
3.3.2. HBA fibra canal para alta disponibilidad 79
3.3.2.1. Guías para seleccionar un HBA de fibra canal para alta disponibilidad 79
3.3.2.2. Ventajas y desventajas de seleccionar HBA de fibra canal 80
3.3.3. SAN para alta disponibilidad 81
3.3.3.1. Definición 81
3.3.3.2. Guías para seleccionar un SAN para alta disponibilidad 82
3.3.3.3. Ventajas y desventajas para seleccionar una SAN 83
3.3.3.4. Topología de SAN 83
3.3.3.5. Disponibilidad de los datos en una SAN 90
3.4. Niveles de RAID 95

3.4.1. RAID 0: Arreglo de discos distribuido sin tolerancia a fallas 95
3.4.1.1. Ventajas 96
3.4.1.2. Desventajas 96
3.4.1.3. Aplicaciones recomendadas 96
3.4.2. RAID 1: Discos Espejo 97
3.4.2.1. Ventajas 97
3.4.2.2. Desventajas 98
3.4.2.3. Aplicaciones recomendadas 98
3.4.3. RAID 2: Código Hamming ECC 98
3.4.3.1. Ventajas 99
3.4.3.2. Desventajas 99
3.4.4. RAID 3: Transferencia paralela con paridad 100
3.4.4.1. Ventajas 100
3.4.4.2. Desventajas 100
3.4.4.3. Aplicaciones recomendadas 101
3.4.5. RAID 4: Discos de datos independientes con disco de paridad compartida 101
3.4.5.1. Ventajas 102
3.4.5.2. Desventajas 102
3.4.6. RAID 5: Discos de datos independientes con bloques de paridad distribuidos 102
3.4.6.1. Ventajas 103
3.4.6.2. Desventajas 103
3.4.6.3. Aplicaciones recomendadas 103
3.4.7. RAID 6: Discos de datos independientes con dos esquemas de paridad independientes distribuidos 104
3.4.7.1. Ventajas 105
3.4.7.2. Desventajas 105

3.4.8. RAID 7: Asincronía optimizada para altas tasas de entrada/salida 105
3.4.8.1. Características de la arquitectura: 106
3.4.8.2. Ventajas 106
3.4.8.3. Desventajas 107
3.4.9. RAID 10: Muy alta confiabilidad combinada con alto desempeño 107
3.4.9.1. Ventajas 108
3.4.9.2. Desventajas 108
3.4.9.3. Aplicaciones recomendadas 108
3.4.10. RAID 53: Altas tasas de entrada/salida 109
3.4.10.1. Ventajas 109
3.4.10.2. Desventajas 109
3.4.11. RAID 0+1: Alto desempeño de transferencia de datos 110
3.4.11.1. Ventajas 110
3.4.11.2. Desventajas 111
3.4.11.3. Aplicaciones recomendadas 111
4. SISTEMAS OPERATIVOS EN ALTA DISPONIBILIDAD 113
4.1. Cluster 113
4.1.1. Componentes del cluster 113
4.1.2. Beneficios del cluster 114
4.1.3. Tecnología de cluster 115
4.1.3.1. Terminología 115
4.1.3.2. Modelos de cluster 116
4.1.4. Configuración activo/en espera 117
4.1.4.1. Ventajas activo/en espera 118
4.1.4.2. Consideraciones activo/en espera 118
4.1.5. Configuración activo/activo 118
4.1.5.1. Ventajas activo/activo 119
4.1.5.2. Consideraciones activo/activo 119

4.2. Cluster para Microsoft® Windows® 120
4.2.1. Configuración típica para un cluster Microsoft® 120
4.2.2. Cluster para Microsoft® Windows® 2000 121
4.2.2.1. Balanceo de carga de red 121
4.2.2.2. Cluster de servidores 122
4.2.3. Mejoras al servicio cluster Windows® 2000 datacenter server 123
4.2.4. Servicio de cluster en Windows® 2000 advanced server 123
4.2.4.1. Solución de cluster con una sola ruta 123
4.2.4.2. Solución sin puntos de falla 124
4.3. Novell® NetWare® Cluster 125
4.3.1. Componentes cluster Novell® NetWare® 126
4.3.2. Servicio cluster NetWare® 127
4.3.3. Novell® ConsoleOne® 128
4.3.4. Servicio de directorio NetWare® 130
4.4. Cluster con Linux 132
4.4.1. Lifekeeper para cluster Linux 133
4.4.1.1. Cluster de espejo replicado 133
4.4.1.2. Cluster SCSI compartido 135
4.4.1.3. Cluster fibra compartida 135
4.4.1.4. Interconexión del cluster 138
5. BASES DE DATOS EN ALTA DISPONIBILIDAD 139
5.1. Microsoft® SQL Server 2000 Cluster 139
5.1.1. Introducción 139
5.1.2. Tecnología de base de datos 139
5.1.3. Arquitectura de SQL Server 2000 140
5.1.3.1. Servicio de cluster 141
5.1.3.2. Comunicación 142
5.1.3.3. Estrategia de red 143

5.1.3.4. Arquitectura de disco 144
5.2. Oracle® real application cluster 145
5.2.1. Arquitectura de real application cluster 145
5.2.1.1. Configuraciones típicas soportados por RAC 145
5.2.2. Beneficios del cluster 146
5.2.2.1. Escalabilidad 146
5.2.2.2. Alta disponibilidad 146
5.2.3. Estructura de disponibilidad del cluster 147
5.2.3.1. Aislamiento de la falla 148
5.2.3.2. Recuperación 148
5.2.4. Estructura de disponibilidad de ORAC 149
5.2.4.1. Arquitectura 149
5.2.4.2. Recuperación de la base de datos en un ambiente ORAC 150
5.2.4.3. Recuperación de falla sobre conexiones TCP/IP 151
5.2.4.4. Configuración en línea 152
5.2.5. Construyendo estructura de disponibilidad de ORAC 152
5.2.5.1. Requerimientos de disponibilidad 152
5.2.5.2. Consideraciones instalación sistemas cluster 153
5.2.5.3. Conexiones clientes 154
6. HERRAMIENTAS DE ADMINISTRACIÓN DE RED 155
6.1. Características y protocolos administración red 155
6.1.1. ¿Qué es administración de red? 155
6.1.2. Objetivos de la administración de red 156
6.1.3. Modelo de administración de red 157
6.1.4. Protocolo de administración de red - SNMP 158
6.1.5. Base de administración de información (MIB) 159
6.1.5.1. Nombre de los objetivos MIB 160
6.2. Áreas funcionales de administración de red 161

6.2.1. Modelo funcional 162
6.2.2. Administración de fallas 163
6.2.3. Administración de configuración 164
6.2.4. Administración utilización de recursos 166
6.2.5. Administración de rendimiento 166
6.2.6. Administración de seguridad 168
7. PROCEDIMIENTOS PARA ASEGURAR LA ALTA DISPONIBILIDAD 169
7.1. Introducción 169
7.2. Mejores prácticas comunes 171
7.2.1. Evaluar negocio/sistema actual y definir objetivos 172
7.2.1.1. Propósito del proyecto 172
7.2.1.2. Equipo proyecto 173
7.2.1.3. Alcance proyecto 174
7.2.1.4. Entregables 174
7.2.2. Creación de planes 175
7.2.2.2. Registro proyecto 176
7.2.2.3. Plan de comunicación 176
7.2.2.4. Plan de recursos 177
7.2.2.5. Plan de escalación de problemas 178
7.2.2.6. Plan de entrenamiento 179
7.2.2.7. Plan de pruebas 179
7.2.2.8. Plan de piloto 180
7.2.2.9. Plan de capacidad 181
7.2.2.10. Plan de recuperación de desastres 182
7.2.2.11. Punto de revisión 183
7.2.3. Creación diseño negocio/sistema 184
7.2.4. Pruebas y piloto 184
7.2.5. Lecciones aprendidas 186

7.2.6. Estrategia de implementación 186
7.2.7. Operación día a día 188
7.2.7.1. Administración sistema 188
7.2.7.2. Comunicación 189
7.2.7.3. Administración de cambios/configura ción 190
7.2.7.4. Administración de problemas 191
CONCLUSIONES 193
RECOMENDACIONES 195
BIBLIOGRAFÍA 197

ÍNDICE DE ILUSTRACIONES

FIGURAS

Numero		*Página*
1	Representación funcional de una red de un solo sitio	13
2	Representación funcional de una red de múltiples sitios	15
3	Arquitectura red sitio comercio electrónico en alta disponibilidad	18
4	Ejemplo modo servidor DNS cache nombres	23
5	Ejemplo modo re-director sesiones HTTP	24
6	Métricas DRP internas y externas	26
7	Diagrama de red con múltiples caché contenido	35
8	Redundancia protocolos entre los supervisores	41
9	Enlaces redundantes entre un par de switches	44
10	Switch con enlaces redundantes utilizando spanning-tree-protocol	44
11	Múltiples switches con rutas redundantes.	45
12	Firewalls en alta disponibilidad	48
13	Conexión LocalDirector en failover	56
14	Configuración LocalDirector en Failover	57
15	Procesador	60
16	Comparación buses PCI & PCI-X	62
17	Fuente de poder con capacidad de cambiarse en caliente	63
18	Sistema de almacenamiento externo	64
19	Caídas de servidores por fallas de memoria en un año	69
20	Memoria ECC avanzada	70
21	Falla de DIMM en modo de online spare memory	72
22	Memoria en espejo	73

23	Memoria RAID hot plug	74
24	Arquitectura de memoria RAID	75
25	Disco SCSI hot-pluggable	76
26	SAN con un solo switch	84
27	SAN en cascada	85
28	Diseño de una SAN en malla	86
29	Topología de SAN en anillo	88
30	Topología SAN en backbone	89
31	Nivel 1: Máxima conectividad	92
32	Nivel 2: infraestructura de SAN confiable	93
33	Nivel 3: alta disponibilidad en una SAN con múltiples enlaces	93
34	Nivel 4: SAN sin puntos de falla	94
35	RAID 0	95
36	RAID 1	97
37	RAID 2	98
38	RAID 3	100
39	RAID 4	101
40	RAID 5	102
41	RAID 6	104
42	RAID 7	105
43	RAID 10	107
44	RAID 53	109
45	RAID 0+1	110
46	Componentes del cluster	114
47	Ejemplo de disponibilidad de datos	115
48	Configuración activa/en espera	117
49	Configuración activo/activo	119
50	Configuración de un cluster Microsoft®	121

51	Servicio cluster para balanceo carga red	122
52	Solución de cluster con una sola ruta	124
53	Solución de cluster sin puntos de falla	125
54	Componentes principales de un cluster NetWare®	127
55	ConsoleOne® vista de consola	129
56	ConsoleOne® vista estado del cluster	130
57	Servicio de directorio NetWare®	131
58	Cluster imágenes replicadas con dos nodos	134
59	Cluster de espejos replicados con cuatro nodos	134
60	Cluster SCSI compartido	135
61	Cluster de fibra compartida estándar	136
62	Cluster de fibra compartida duplicada	137
63	Cluster de fibra compartida mixta	138
64	Cluster SQL Server 2000	140
65	Elementos del cluster: servidores, interconexión, y disco compartido	145
66	Componentes de software de la base de datos en cluster Oracle®	149
67	Modelo de administración de red	157
68	Protocolo SNMP	159
69	Jerarquía MIB	161
70	Mejores prácticas planeación y administración alta disponibilidad	170
71	Bloques fundamentales	1703

TABLAS

I	Porcentajes de disponibilidad de aplicaciones	5
II	Costo promedio por hora de tiempo fuera de línea	7
III	Rendimiento del bus PCI & PCI-X	61
IV	Diferencia sistemas operativos Microsoft® Windows® 2000	120

GLOSARIO

Bit	Cifra binaria. Unidad mínima de información que sólo puede tomar uno de los dos valores siguientes: 0 o 1.
Byte	Grupo de 8 bits con el que se representa un carácter.
Chipset	Grupo de circuitos integrados de la computadora, que cuando trabajan conjuntamente, manejan y controlan la computadora.
Cluster	Conjunto de dispositivos (computadoras, equipo de comunicación) que se ven como uno solo, agrupados para dar una mayor disponibilidad y rendimiento a la solución.
Failover	Proceso por medio del cual un dispositivo electrónico que ha fallado, pasa el control a otro que está en espera de tomarlo.
Firewall	Dispositivo electrónico utilizado para proteger una o varias de redes de datos de usuarios externos o internos, limitando la capacidad de comunicación que estos tienen dentro de la misma.

Hardware	Conjunto de componentes físicos (cables, tornillos, placas, etcétera) que constituyen una computadora.
Memoria	Dispositivo o parte de un equipo, destinado a almacenar de forma temporal o permanente informaciones codificadas y a devolverlas cuando se soliciten.
Procesador	Dispositivo electrónico que controla las operaciones que deben efectuarse en una computadora para obtner los resultados apetecidos.
Protocolo	Conjunto de reglas y métodos por seguir para intercambiar informacion entre dos computadoras.
RAM	(Random access memory) Memoria de acceso directo o aleatorio.
Ruteador	Dispositivo electrónico encargado de enviar paquetes de datos basados en las tablas de direcciones físicas de otros dispositivos. Se utilizan para la conexión de diferentes redes de área local.
Software	Conjunto de programas que pueden ejecutar una computadora.
Switch	Dispositivo electrónico utilizado para conectar dos o más dispositivos en una red de área local. Reconoce la

dirección física de los dispositivos, recibe paquetes de datos y selectivamente lo envía por uno de sus puertos.

RESUMEN

Internet, una red de datos que une al mundo, proporciona gran diversidad de servicios a sus usuarios. Internet abre nuevas oportunidad, y crean nuevos retos, los negocios pueden llegar a un número mayor de clientes, y mantener servicio a los mismos 24 horas al día 365 días al año, en forma continua, alrededor del globo terrestre. Todo esto apoya por tecnología de punta y en la cual se requieren altos niveles de disponibilidad.

Una solución de comercio electrónica, tiene tres características claves desde el punto de vista de la infraestructura, que son: alta disponibilidad, escalabilidad, y seguridad. Estas características deben observarse en toda la arquitectura de la solución en cada una de sus capas de *hardware* y *software*.

La alta disponibilidad es la habilidad para proveer acceso continuo a los servicios de comercio electrónico para los clientes. Un requisito fundamental para los negocios en Internet disponibles 24 horas al día.

Examinaremos las opciones que ofrece el mercado para alta disponibilidad a nivel de redes de datos, servidores, sistemas operativos, y bases de datos, también se examina qué se ofrece en las herramientas de administración de redes fundamental para completar la arquitectura de alta disponibilidad.

Por último, se tocarán los procedimientos necesarios para asegurar la alta disponibilidad, basados en las mejores prácticas y recomendaciones de los proveedores líderes en el mercado. Lo mejor en *hardware* y *software* no podrán

asegurar la alta disponibilidad de un sitio de comercio electrónico, si no son correctamente administrados.

OBJETIVOS

Generales

Dar a conocer como construir una infraestructura de alta disponibilidad para comercio electrónico utilizando tecnología de punta.

Específicos

1. Describir la arquitectura de la infraestructura de una solución de comercio electrónico, y sus componentes.
2. Describir la alta disponibilidad a nivel de los componentes y servicios de red.
3. Describir la alta disponibilidad a nivel de los servidores Web.
4. Describir la alta disponibilidad a nivel de los servidores de base de datos.
5. Describir la alta disponibilidad a nivel del diseño de la aplicación
6. Presentar laboratorios prácticos que demuestren los conceptos, y clarifiquen las ideas.
7. El trabajo se llevará a cabo alrededor de la tecnología vigente al momento de desarrollar la tesis.

INTRODUCCIÓN

Internet, una red de datos que une al mundo, proporciona gran diversidad de servicios a sus usuarios. Internet abre nuevas oportunidades, y crea nuevos retos; los negocios pueden aumentar su número de clientes, y mantener servicio a los mismos, 24 horas al día los 365 días del año, en forma continúa. Todo esto se apoya en la tecnología de información impone un nivel de servicio por prestar a los clientes.

Proponer una solución de comercio electrónica en producción requiere de varios componentes; entre ellos la infraestructura, que es el conjunto de componentes de *hardware* y *software* que forman la base de la solución. Una solución de comercio electrónico tiene tres características claves desde el punto de vista de la infraestructura: alta disponibilidad, escalabilidad, y seguridad. Estas características deben observarse en toda la arquitectura de la solución en cada una de sus capas, esto incluye la red de datos, el sistema operativo, las bases de datos, y las aplicaciones *web*.

Por alta disponibilidad se entiende a la habilidad para proveer acceso continuo a los servicios de comercio electrónico.

Una infraestructura de comercio electrónica de alta disponibilidad empieza con un diseño correcto de red. El diseño correcto de red asegura que las fallas no impacten la alta disponibilidad de todo el sistema. Diseñar para alta disponibilidad incluye la eliminación de cualquier punto singular de falla por la provisión de dispositivos y rutas de red redundantes.

La alta disponibilidad también puede lograrse en el sistema operativo, servicios del sistema, y el código de la aplicación a través una mezcla de redundancia en el servidor y *failover*. Con un sitio de comercio electrónico, redundancia en el servidor significa múltiples servidores que están disponibles para procesar un requerimiento. El concepto de *failover* es que una característica es implementada vía un proceso específico, si el proceso falla entonces un proceso alternativo automáticamente corre y toma el control.

En el presente documento encontrará un planteamiento para la arquitectura de una solución de comercio electrónico, sobre la cual se construirá la alta disponibilidad en cada una de sus capas.

Dentro de las capas de arquitectura de comercio electrónica que se plantean se encuentra la infraestructura de red. La cual tiene varios componentes que pueden provocar que la solución total falle, por lo que punto a punto hay que trabajarla en la alta disponibilidad. Dentro de la red se examinan los enlaces a los proveedores de servicio (ISP), los ruteadores, *firewalls*, *content engine*, *switches*, y se examinan algunos servicios de red, necesarios para el funcionamiento de la red Microsoft® como *DNS*, *WINS*, y *Active Directory*.

La siguiente capa a examinar son los servidores *web*. En este punto se revisara los componentes de *hardware* que integran el servidor, y las alternativas que existen para lograr alta disponibilidad en ellos. Esta capa se complementa con un examen de las características más relevantes del sistema operativo Microsoft® Windows 2000, y la utilización de tecnología *cluster*, y balance de carga de red.

En el siguiente escalón en la solución de comercio electrónico está la base de datos, el repositorio donde se almacena la información. Al igual que las capas anteriores, esta es de gran importancia ya que su falla puede provocar que el servicio a los clientes se detenga, incluso puede provocar perdidas de información, que se traducen en perdidas financieras para el negocio.

Por último, está el diseño de la aplicación. Esta contiene la lógica del negocio, y que es nuestra carta de presentación ante los usuarios, requiere un diseño que permita mantener la operación del negocio aun cuando falle alguno de sus componentes.

1. CONCEPTOS GENERALES

En la económica de Internet actual, las compañías deben proveer de sitios de comercio electrónico que sean altamente disponibles, escalables y seguros. Estos sitios deben también implementarse rápidamente; esto no es fácil. De cualquier forma estos son los retos que deben enfrentar los negocios de hoy para desarrollar su arquitectura de comercio electrónica.

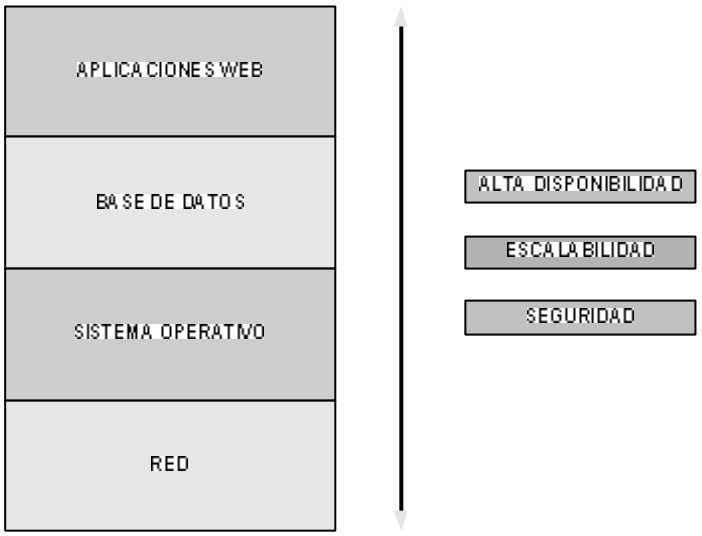
Conforme el número de usuarios de Internet y aplicaciones de misión crítica basadas en Internet se incrementan a diario a un ritmo sin precedentes, los proveedores de servicios y los clientes corporativos demandan mayor confiabilidad y disponibilidad. Cuando cada minuto fuera de línea puede significar millones de dólares en pérdidas y encabezados vergonzosos, las compañías buscan soluciones para lograr alta disponibilidad en sus sistemas. Las características de alta disponibilidad de los equipos ayudan a los usuarios a incrementar la disponibilidad de los servicios y protegen el rendimiento financiero, reputación, y la lealtad de los clientes.

1.1. Servicios comercio electrónico

Para implementar una solución exitosa de comercio electrónica, deben cubrirse tres características claves: alta disponibilidad, escalabilidad, y seguridad. Una solución sólida de comercio electrónico solo puede lograrse a través de una arquitectura que cumpla estos requerimientos a través de las diversas capas de

sus componentes, tal como: la red, el *hardware* del servidor, el sistema operativo y los servicios de red, la aplicación *web*, y la base de datos.

Figura 1. Requerimiento servicios de comercio electrónico



1.1.1. Alta disponibilidad

La alta disponibilidad es la habilidad para proveer acceso continuo a los servicios de comercio electrónico para sus clientes. Para entregar estos servicios de comercio electrónico exitosamente, la alta disponibilidad debe ser maximizada a través de todas las capas de una infraestructura para incluir la disponibilidad al nivel de sesión y servicio. La disponibilidad de sesión es la habilidad de la infraestructura para mantener el estado de una sesión de red en el evento de una falla. La disponibilidad de servicio es la habilidad de los usuarios para conectarse a un servicio de comercio electrónico en el evento de una falla.

Una alta disponibilidad de infraestructura de comercio electrónica empieza con el diseño correcto de red. El diseño correcto de red asegura que una falla no impacte la alta disponibilidad del sistema completo. Diseñando para alta disponibilidad incluye la eliminación de cualquier punto individual de falla mediante dispositivos y rutas de red redundantes. En el evento de una falla, la infraestructura debe ser capaz de responder rápidamente re-direccionando alrededor del dispositivo fallido. Adicionalmente, donde sea necesario, los dispositivos deben proveer de *stateful failover* a una unidad en espera. Esto asegura que ciertas sesiones de aplicación, tal como transacciones de comercio, no expiren, y causen que las sesiones de usuario se pierdan.

Para niveles adicionales de alta disponibilidad, se puede construir un sitio remoto que ofrezca servicios de comercio electrónico geográficamente dispersos y que actúe como un respaldo para tomar ventaja de un balanceo de carga geográfico. Esta solución varia dependiendo en el grado de transacciones deseado del sitio remoto.

Alta disponibilidad puede también ser lograda en el sistema operativo, servicios del sistema y las capas de código de la aplicación a través de una mezcla de redundancia y *failover* en el servidor. En un sitio de comercio electrónico, redundancia en el servidor significa que múltiples servidores están disponibles para procesar las peticiones. Por ejemplo, una página Web puede ser servida desde cualquiera de múltiples servidores Web en la granja. El concepto de *failover* es una característica que es implementada vía un proceso específico; si este proceso falla entonces un proceso alterno automáticamente se levanta y toma el control. Por ejemplo, un servidor de base de datos implementa el *failover* a otro servidor de base de datos.

Relacionado con la alta disponibilidad está la tolerancia a desastres, día a día nos enteramos de desastres que suceden alrededor del mundo, y es importante considerar esto dentro del diseño del sistema. De forma que el negocio pueda sobrevivir a un desastres, al menos en sus servicios críticos, y que le permita restablecerlos en un corto período de tiempo.

1.1.1.1. Nivel de disponibilidad

Hay que determinar el nivel de disponibilidad que el sitio de comercio electrónico debe lograr. El nivel de disponibilidad de un sitio de comercio electrónico es una medida del máximo tiempo fuera de línea que el usuario puede tolerar. Es importante comprender el nivel de disponibilidad que el usuario quiere puesto que el nivel de disponibilidad deseado impactará en la diseño de la infraestructura del sitio.

El nivel de disponibilidad es representada en términos del porcentaje de tiempo en línea para el sitio web. El porcentaje de tiempo en línea puede ser 99%, 99.9%, 99.99%, 99.999% o más allá, el cual es referido como el nivel de nueves (9). Debe determinar el nivel de disponibilidad que quiere lograr y el marco de tiempo dentro del cual se medirá la disponibilidad. Por ejemplo, un banco puede tener una aplicación B2B, negocio a negocio, (por sus siglas en inglés) con un nivel de disponibilidad deseado de 99.999% de 8:00 AM a 8:00 PM. Fuera de este horario, la aplicación puede estar fuera de línea.

La disponibilidad no es solo un concepto, es una ciencia; esta puede expresarse matemática. Un sistema disponible es uno que es utilizable cuando el cliente lo necesita. Un sistema puede ser altamente disponible, operando de 8

a.m. a 5 p.m., si eso es lo que demanda el negocio. El tiempo restante puede ser usado para programar mantenimientos y reparaciones. Disponibilidad es definido como el servicio actual por el servicio requerido. El reto para muchos sistemas hoy es operar 24 horas al día, 365 días al año (también referido como 7x24, o 365 x 24).

La disponibilidad también es expresada en porcentaje. Un sistema 365 x 24 con 99.9 por ciento de disponibilidad tiene un promedio de downtime de 8.76 horas por año (525 minutos). Un sistema con solo 3 minutos fuera de servicio debe tener una disponibilidad de 99.999 por ciento.

Tabla I: Porcentajes de disponibilidad de aplicaciones

Porcentaje de disponibilidad	**Minutos tiempo fuera de línea**	**Aplicación típica**
99	5256 (3.65 días)	
99.9	525 (8.75 horas)	Escritorio o servidor típico
99.99	52	Servidor tipo empresarial
99.999	5	Servidor tipo *carrier*
99.9999	0.5 (30 segundos)	Equipo *carrier switch*

La disponibilidad se calcula usando modelos estadísticos para todos los componentes del sistema, el modelo más simple para un componente de momento es binario. El componente está en o fuera de servicio. La disponibilidad puede ser calculada a partir de la tasa de fallas, medida en el tiempo medio entre fallas (MTBF por sus siglas en inglés, *mean time between failures*), y el tiempo de reparación, medido en el tiempo medio de reparación (MTTR, por sus siglas en inglés, *mean time to repair*). La contribución promedio

de tiempo fuera de línea de cualquier componente es calculado amortizando el tiempo MTTR sobre el período MTBF. Por ejemplo, si un componente crítico de la operación de la plataforma tiene un MTBF de 250,000 horas y un MTTR de 1 hora, esto contribuye 2.1 minutos (60 minutos / 250, 000 horas / 8760 horas / año) a la indisponibilidad del sistema por año.

La disponibilidad en el rango de dos nueves o tres nueves (99 a 99.9 por ciento) puede lograrse al maximizar la confiabilidad de los componentes y minimizar el tiempo de reparación. Para lograr alta disponibilidad o para compensar los componentes menos confiables, se usa la redundancia. Además, tiene un respaldo para el componente que falla se logra mantener el sistema operando. La disponibilidad de configuraciones redundantes es calculada basada en el tiempo que toma detectar y mover la operación al componente redundante.

¿Cuál es el costo del tiempo fuera de línea?

- Productividad, número de empleados impactados por hora fuera por su costo por hora.
- Daño a la reputación, clientes, proveedores, mercado financiero, bancos, socios de negocios, …
- Facturación, pérdidas directas, pagos compensatorios, perdida de futuras ventas, perdidas de facturación, perdida de inversión.

El Gartner Group estudió en 1998 el costo del tiempo fuera de línea para una variedad de industrias en Estados Unidos de América, y los resultados se resumen en la siguiente tabla:

Tabla II. Costo promedio por hora de tiempo fuera de línea

Industria	**Aplicación**	**Costo promedio por hora fuera de línea**
Financiera	Operaciones Bolsa	USD6,500,000.00
Financiera	Tarjetas Crédito	USD2,600,000.00
Media	Pago por ver	USD1,150,000.00
Menudeo	Compra por TV	USD 113,000.00
Menudeo	Venta por Catálogo	USD 90,000.00
Transporte	Reservación Líneas aéreas	USD 89,500.00

1.1.2. Escalabilidad

Uno de los errores más comunes en los sitios de comercio electrónico es desestimar sus requerimientos de escalabilidad; porque escalabilidad es asociada solo con ampliación de rendimiento tal como incremento en la velocidad de CPU, incremento ancho de banda en la red, entre otros. Sin embargo, debe tomarse en cuenta el soporte a un gran número de sesiones de usuario y transacciones de comercio simultáneas. Es decir, que escalabilidad debe tomarse en cuenta a través de todas las facetas de la infraestructura de comercio electrónico, incluyendo aplicaciones *web*, bases de datos, sistemas operativos servidor, y la red.

Calcular los requerimientos de escalabilidad puede ser bastante dificultoso; por ejemplo, Forrester Research analizó el crecimiento de 50 sitios de comercio electrónico en 1999. El resultado en este reporte muestra que el crecimiento de estos sitios varía de 0 a 400 por ciento. Administrar la escalabilidad de un sitio de comercio electrónico que está creciendo en un 400

por ciento no es fácil. La clave es identificar cualquier problema de escalabilidad con un sitio de comercio electrónico y atacarlo tan rápido como sea posible.

Escalar un sitio de comercio electrónico puede lograrse por medio de servidores más grandes o escalando a más servidores. Crecimiento vertical es cuando un solo servidor es hecho más grande a través de la adición de procesadores, memoria, almacenamiento en disco, y así sucesivamente. Crecimiento vertical requiere un sistema operativo, servicios del sistema y código de aplicación que pueda usar el *hardware* adicional. Los sitios de comercio electrónico pueden escalar sus servidores *web*, de aplicación y de datos para incrementar el número de peticiones que el sitio puede procesar. Crecimiento horizontal es cuando múltiples servidores funcionan como una sola unidad lógica o "granja". El crecimiento horizontal también logra el resultado deseado de incrementar el número de peticiones que un sitio puede procesar. Tal como el crecimiento vertical, como horizontal puede ser hecho en cualquiera de las capas lógicas del sitio. Los sitios de comercio electrónico deben ser posicionados para tomar ventaja de ambos crecimiento vertical y crecimiento horizontal.

¿Cuando un sitio de comercio electrónico crece verticalmente versus horizontalmente? En el pasado, los sitios típicamente crecían verticalmente sus servidores de datos y horizontalmente sus servidores *web*. Los pros y contras de crecer verticalmente versus horizontalmente son generalmente opuestos. Por ejemplo, el costo asociado con crecer verticalmente es usualmente más que el costo asociado con crecer horizontalmente. Del mismo modo, crecer horizontalmente servidores de datos es más complejo que servidores *web*, pero administrar una granja horizontal es más completo que administrar un solo servidor. Finalmente, crecer horizontalmente toma ventaja de la capacidad de *hardware* incrementada mientras que múltiples servidores en una solución

horizontal provee redundancia, lo cual significa alta disponibilidad. Las soluciones de hoy ofrecen a los sitios de comercio electrónico una mezcla de crecimiento vertical y horizontal a través de sus servidores de *web*, aplicación y datos. Los sitios deben diseñarse para una capacidad sin límites de crecer horizontalmente mientras maximizan los beneficios de crecer verticalmente. Esto soporta un enfoque "pague por lo que usted crece" para expandir la tecnología como un opuesto a un enfoque "crezca en lo que usted compra". El resultado es una inversión inicial pequeña en *software* y *hardware*, lo cual puede ser expandido al crecer el negocio, y soporte las estrategias claves de comercio electrónico de acelerar la salida al mercado y una baja inversión inicial.

Finalmente, un sitio de comercio electrónico puede lograr escalabilidad en la infraestructura tomando ventaja de ciertos productos de red. Por ejemplo, una infraestructura de red puede escalar los servidores Web a través del uso de productos de balanceo de carga. Los productos de balanceo de carga inteligentemente distribuyen los requerimientos de los usuarios a través de un grupo de servidores para maximizar el uso de los servidores. También puede tomar ventaja cache de contenido para descargar las peticiones de usuario para el contenido estático de los servidores *web*. Esto ayuda a acelerar la entrega de contenido a los usuarios finales y permite a los servidores enfocarse más en las sesiones interactivas.

1.1.3. Seguridad

Una fuerte seguridad es una de las mayores consideraciones para la infraestructura de red de comercio electrónico. Puesto que la naturaleza de una red de comercio electrónica es conducir transacciones financieras, esta es objeto de actividad maliciosa originada desde la comunidad de Internet en general. De

cualquier forma, la solución de seguridad escogida debe basarse en la naturaleza del negocio de comercio electrónico que será conducido, el nivel de confort de la organización de **TI**, y la comprensión de los riesgos asociados con cada grado de implementación de seguridad. Los componentes de seguridad de una solución de comercio electrónico incluyen cinco elementos claves:

- Seguridad perimetral – Protege contra actividad maliciosa.

- Seguridad de identificación – Provee servicios de autenticación de usuarios.

- Integridad y privacidad de datos – Asegura la confiabilidad de los datos a través de encripción.

- Seguridad *firewall* - Provee servicio de seguridad *stateful***.**

- Monitoreo de seguridad – Reconoce las vulnerabilidades, detecta y reacciona ante intrusos.

La seguridad perimetral provee la primera línea de defensa para una red de comercio electrónico. Esta seguridad se logra por el uso de ruteadores de borde o *firewalls* en la red. Los servicios de seguridad pueden establecerse en el ruteador de borde o *firewall* para proteger contra actividad maliciosa y solo permitir el tráfico válido en la red de comercio electrónico. Por ejemplo, un ruteador o *firewall* de borde puede ser configurado para permitir solo tráfico *web* válido.

Para seguridad de identidad, la autenticación es la primera tarea en cada petición, aun si se iguala a usuarios públicos o anónimos. Autenticación identifica quién esta haciendo la petición y si esta es la base de una autorización, la cual controla qué contenido y a qué servicios puede tener acceso una petición.

La autenticación puede ocurrir a través de varios niveles de seguridad, desde una simple combinación de identificador de usuario y palabra clave a una alta certificación encriptada. Los niveles de seguridad también pueden ser mezclados.

Para incrementar la integridad y privacidad de datos, los sitios de comercio electrónico deben soportar conexiones *Secure Socket Layer* (SSL). SSL puede ser implementado en la capa de *software* o tarjetas aceleradoras de *hardware* y pueden ser usados para descargar procesamiento del CPU del servidor.

Esto puede complementarse con la utilización de dispositivos de *hardware*, conocidos como *tokens*, para la autenticación de usuarios en lo que se conoce como autenticación de dos fases. Bajo este concepto la autenticación se basa en algo que el usuario tiene (*token*), y algo que el usuario conoce su contraseña.

La seguridad de firewal es usada en áreas de la red de comercio electrónica donde los servicios de seguridad *stateful* son requeridos. Esto es típico en el frente de los servidores de base de datos que contienen información confidencial de los clientes para asegurar que la integridad de los datos no esta comprometida. Los servicios de seguridad *stateful* llevan el rastro del estado de cada sesión de usuario y terminan la conexión al final de la sesión.

Finalmente, todo comercio electrónico debe incluir cierto grado de monitoreo de seguridad. El monitoreo de seguridad provee la habilidad de rastrear rutinariamente su infraestructura de comercio electrónico, detectando cualquier hoyo potencial de seguridad, y reportarlos para ser corregidos. El monitoreo de seguridad también provee la habilidad para descubrir un ataque en progreso, generar una alerta, y detener el ataque.

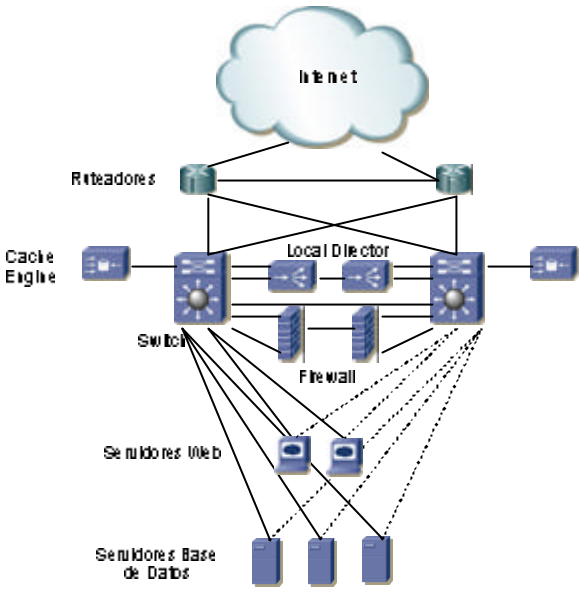
1.2. Arquitecturas de comercio electrónico

La arquitectura de un sitio de comercio electrónico cabe dentro de dos categorías básicas: un solo sitio y múltiples sitios; esta sección describe los componentes básicos de estas dos arquitecturas.

1.2.1. Arquitectura de un sitio de comercio electrónico

Una arquitectura de un solo sitio de comercio electrónico consiste en dos secciones principales: la red de *front-end* y *back-end*. La red de *front-end* consiste de los servidores de Web y aplicación que son accesibles a los usuarios de Internet. Los dispositivos de red que conectan los servidores de Web y aplicación incluye ruteadores de borde, *switch* de multi-capas, dispositivos cache de contenido, balanceadores de carga y sistemas de detección de intrusos. La red de *back-end* consiste en servidores de base de datos, *firewalls*, y *switches* multi-capa. Un *firewall* típicamente sirve como un punto delimitador entre las secciones de la red *back-end* y *front-end*.

Figura 1: Representación funcional de una red de un solo sitio



La figura anterior es una representación funcional de una implementación de un solo sitio simple con un alto grado de redundancia en la red y los servidores.

1.2.2. Arquitectura de comercio electrónica con múltiples sitios

Una arquitectura de múltiples sitios puede ser construida en varias capas. La arquitectura típicamente incluye un sitio de comercio electrónico principal y uno o más sitios satélites que extienden los servicios de comercio electrónico que ofrece la compañía. Los sitios satélites pueden contener la arquitectura parcial o completa del sitio principal. La clave para determinar los factores en la selección de la arquitectura es el grado de sincronización de las bases de datos deseado

entre los sitios de comercio electrónico y la cantidad de tráfico que debe llevarse al sitio principal.

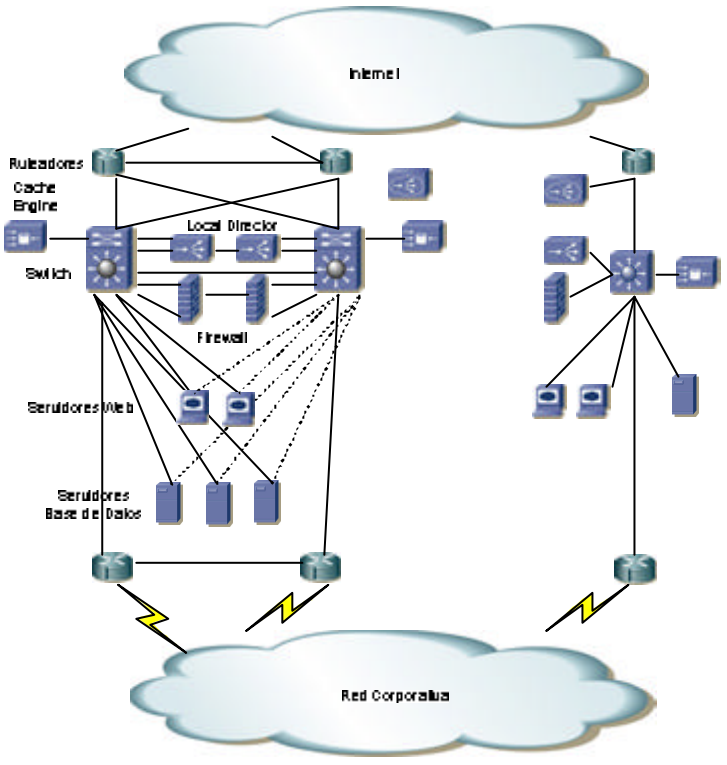
Las compañías se mueven a una arquitectura de múltiples sitios cuando la base de sus usuarios se extiende más allá de su geografía local, y deben de mejorar los tiempos de respuestas de su aplicación de comercio electrónico a estos usuarios geográficamente dispersos. Una arquitectura de múltiples sitios también provee cierto grado de redundancia y respaldo a la compañía en caso de que el sitio principal falle. Los sitios de comercio electrónico satélite están conectados al sitio principal sobre un *backbone* corporativo, tal como *frame relay* o ATM. La sincronización y actualización de las bases de datos, la administración remota, y la integración con un sistema corporativo ERP son efectuadas por medio del *backbone* corporativo. Los diferentes tipos de arquitectura de múltiples sitios son discutidos a continuación:

1. El *front end* del sitio principal de comercio electrónico es replicado y distribuido geográficamente. Puesto que el *front end* consiste principalmente de servidores *web* y su contenido asociado, la habilidad de replicar y distribuir los datos a estos servidores permite a los sitios remotos manejar las peticiones de los usuarios de contenido estático. Usando estos sitios remotos se alivia la necesidad de *backhaul* las peticiones de los usuarios por contenido estático en el sitio principal. Esto también mejora el tiempo de respuesta en las peticiones de los usuarios por contenido *web*.

2. Consiste en la replicación de la red *front-end* del sitio principal “a lo largo de” con una porción de la red *back-end*. Los servidores de aplicación y de base de datos, los cuales son principalmente responsables de

mantener y servir el contenido estático, son replicados al sitio remoto. Información tal como los datos de la información de la cuenta de usuario, información de catalogo de productos e información especial (por ejemplo, descuentos especiales, precio, entre otros) puede ser replicado a los servidores remotos y alivian la necesidad de *backhaul* este tráfico al sitio principal. Solo el tráfico que involucra información dinámica tal como transacciones de comercio es *backhauled* al sitio principal. Esta solución también mejora el tiempo de respuesta en las peticiones de usuario por contenido.

Figura 2: Representación funcional de una red de múltiples sitios



La figura anterior muestra la arquitectura de múltiples sitios descrito anteriormente.

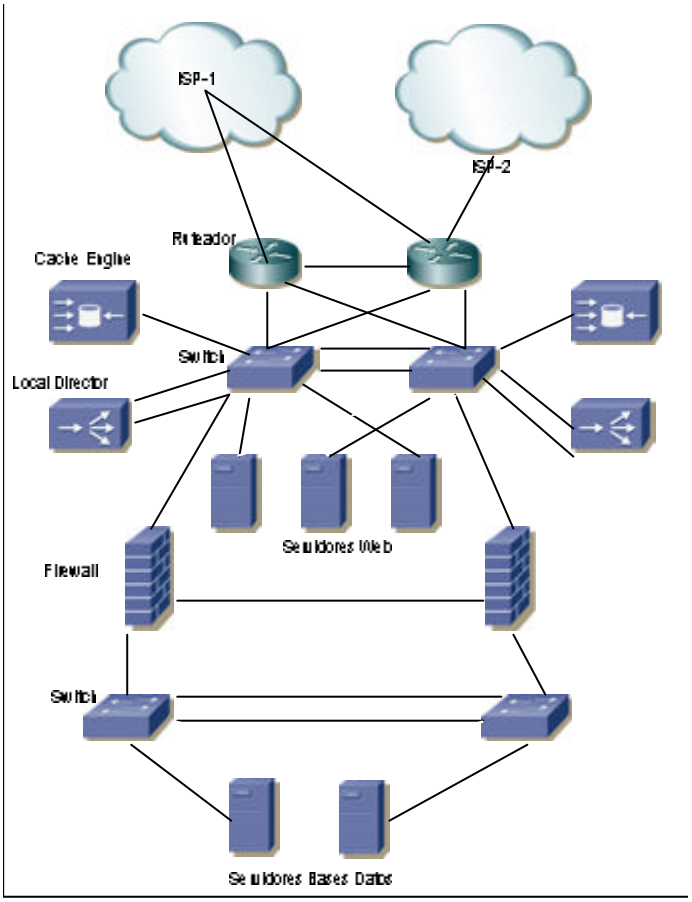
3. Envuelve la creación de un sitio redundante completo que puede mantener el conjunto completo de servicios de comercio electrónico por si falla el sitio principal. En este caso, todas las bases de datos y aplicaciones son completamente replicadas y sincronizadas en tiempo real, o tan cerca al tiempo real como sea posible. Este caso puede permitir que el sitio primario falle completamente sin perder la habilidad de proveer servicios de comercio electrónico a los usuarios. Esta solución provee lo más actual en disponibilidad de servicio para comercio electrónico.

2. ALTA DISPONIBILIDAD EN LA CAPA DE RED

Un usuario ejecutando una transacción crea muchas conexiones de red dentro de un sitio de comercio electrónico. Estas conexiones pasan a través de una serie de dispositivos que definen las capas de la arquitectura del sitio de comercio electrónico. Cada uno de estos dispositivos provee diferentes servicios conforme son necesarios para hacer un sitio de comercio electrónico exitoso. En este capítulo se tratan los diferentes dispositivos que trabajan en las primeras capas de la arquitectura y el servicio que proveen, entre ellos están:

- Balanceadores de carga geográficos,
- Ruteadores de borde,
- *Cache* de contenido,
- *Switch* múltiples capas,
- *Firewall*.

Figura 3. Arquitectura red sitio comercio electrónico en alta disponibilidad



de la topología de red. Esto ayuda a mejorar el tiempo de respuesta de las aplicaciones de comercio electrónico tal como se ven por los usuarios finales, especialmente cuando los sitios geográficos de comercio electrónico están ampliamente distribuidos.

El uso de un balanceador de carga geográfico provee escalabilidad para múltiples sitios, y entrega un alto grado de disponibilidad para el monitoreo del estado de cada sitio de comercio electrónico distribuido. Si un sitio es inoperable, el balanceador de carga geográfico no envía nuevas conexiones clientes al sitio fallido.

Los arquitectos del sitio deben estar preparados para manejar la complejidad de la replicación de contenido bajo una solución de balanceo de carga geográfico. Habrá un retardo entre el momento en que se modifica el contenido original y el momento en el cual es consistente a través de todos los sitios. La solución es relativamente simple si el modelo de negocios permite para los sitios continuar corriendo durante esta inconsistencia. Mientras, que si el modelo de negocio requiere que todos los sitios funcionen solo cuando todo el contenido está consistente entonces habrá que implementar una solución en etapas y sincronización.

2.1.1. Cisco Distributed Director 2500

Cisco Systems ofrece productos como Cisco Distributed Director 2500, el cual provee la función de balanceo de carga entre sitios de comercio electrónico desde una perspectiva global. Dentro de una arquitectura distribuida, uno de los más importantes puntos del diseño es el balanceo de carga entre diferentes

centro de datos. El Cisco Distributed Director (DD) ofrece balanceo de carga a sitios geográficamente dispersos.

2.1.1.1. Servicios Cisco Distributed Director

El Cisco Distributed Director 2500 Series es un dispositivo que eficientemente distribuye los servicios de Internet a través de servidores distribuidos topológica mente dispersos en el Internet o en una Intranet. Esto provee escalabilidad, transparencia e inteligencia en la distribución de carga en la red.

Usando el Director Response Protocol (DRP), una aplicación desarrollada por Cisco utilizando User Datagram Protocol (UDP), el DistributedDirector puede consultar a los ruteadores Cisco apropiadamente configurados con Exterior Gateway Protocol (EGP) e Internal Gateway Procotol (IGP), por metricas de "distancia". Con esta información y otras métricas configuradas, el DistributedDirector puede asignar un servidor distribuido óptimo para cada cliente. Como resultado, los usuarios pueden ser asignados transparente y automáticamente a un servidor distribuido en cualquier lugar en el Internet.

Adicionalmente, el dispositivo utilizado como Director también se requiere la participación del siguiente equipo en el sistema:

- **Un ruteador Cisco para cada servidor distribuido**. El ruteador debe estar topológica mente cerca de el servidor, tener tablas completas de *Border Gateway Protocol* (BGP) o IGP, o ambas, y ser configurado como un agente servidor DRP, el cual tiene habilitado DRP. Un agente DRP puede soportar más de un servidor.

- **Uno o más servidores *Domain Name System* (DNS)**, dependiendo del modo en el cual se configure el Director. El Director pude operar en dos modos: modo servidor DNS cache nombres o modo re-director sesiones HTTP**.** Los servidores distribuidos son asignados a un sub-dominio o nombre de máquina que es servidor por el Director. El Director puede soportar múltiples sub-dominos y nombres máquina que son configurados separadamente, y puede usar diferentes modos.

2.1.1.1.1. Modo servidor DNS cache nombres

En el modo servidor DNS cache nombres, el Director actúa como un servidor DNS de cache de nombre para un sub-dominio específico. A continuación se muestra la forma en que atiende las peticiones (ver figura a continuación).

1. Un cliente solicita un servicio nombra que dispara una resolución DNS-por ejemplo, preguntar por la dirección IP asociada con el nombre de máquina www.sat.gob.gt.

2. El servidor local de DNS del cliente hace una petición DNS recursiva por la dirección IP asociada con www.sat.gob.gt.

3. El servidor primario DNS para el dominio sat.com.gt recibe la petición. El servidor primario de DNS refiere al servidor DNS local del cliente a el Director como un servidor de nombres autoritativo para el subdominio www.sat.gob.gt.

4. El servidor de DNS local del cliente consulta al Director por la direccion IP asociada con www.sat.gob.gt.

5. El Director recibe la consulta y efectúa una búsqueda interna en sus tablas por la información de configuración que identifican a la dirección IP del *DRP server agents* y los servidores que ellos soportan.

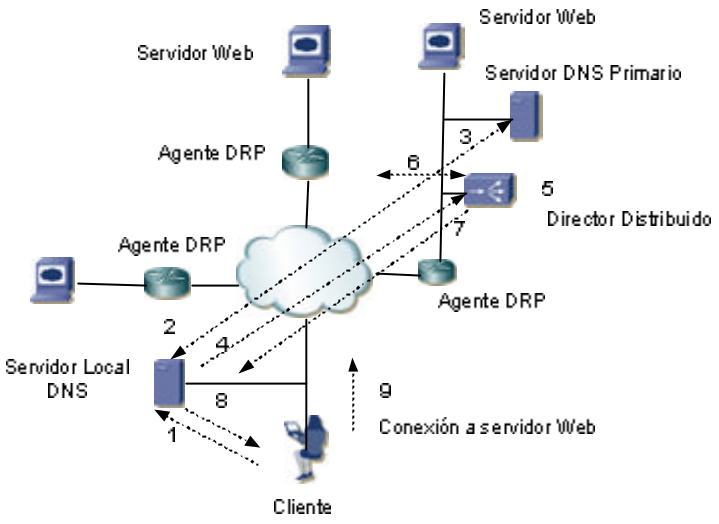
6. Si el subdominio es configurado con ciertas métricas DRP, el Director solicita a cada agente server DRP que seleccione al mejor servidor de acuerdo con el criterio configurado.

7. A partir de la respuesta y las métricas configuradas, el Director selecciona el “mejor” servidor distribuido y retorna su dirección IP al servidor DNS local del cliente.

8. El servidor de DNS local del cliente regresa la dirección IP al cliente.

9. El cliente transparentemente se conecta a la dirección IP para obtener el servicio solicitado

Figura 4. Ejemplo modo servidor DNS cache nombres



2.1.1.1.2. Modo re-director sesiones HTTP

En el modo de sesión HTTP, el re-director provee servicios de redirección http. De esta forma es como las peticiones son atendidas (ver a continuación):

1. Un cliente efectúa una petición http por una dirección URL específica.

2. El director acepta la conexión http, y aparece como el servidor Web solicitado. El director determina el nombre del servidor solicitado por el cliente basado en la dirección IP en la cual la petición http llega.

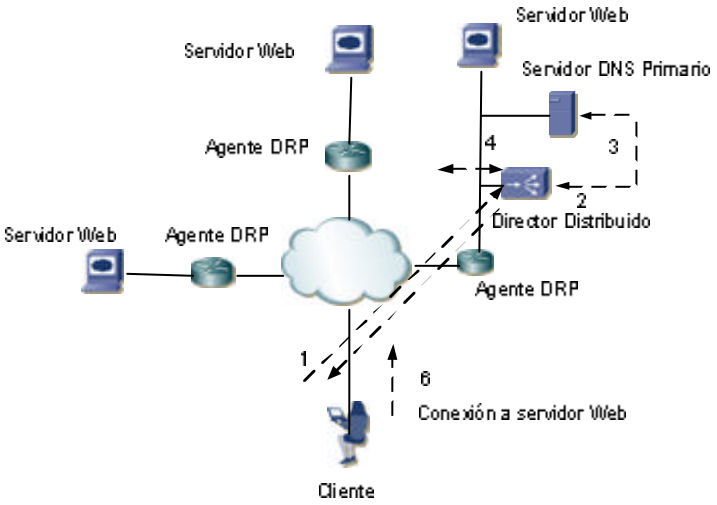
3. Si la información no esta disponible en el caché del director, el director solicita el registro del recurso del servidor de DNS primario. Este registro identifica la dirección IP del agente servidor DRP así como también del servidor Web que el soporta.

4. Si el sub-dominio es configurado con ciertas métricas DRP, el director solicita DRP a cada agente servidor DRP para seleccionar el mejor servidor Web de acuerdo con el criterio configurado.

5. De la respuesta y las métricas configuradas, el Director selecciona a el servidor Web y forma un nuevo URL para este. El Director retorna el URL al cliente, junto con el código http 302, "Movido Temporalmente".

6. El cliente transparentemente se conecta al nuevo URL.

Figura 5. Ejemplo modo re-director sesiones HTTP



2.1.1.2. Métricas

Puede configurarse el director con una o más de las siguientes métricas:

- **DRP externo**. Esta métrica determina el número de saltos del sistema autónomo BGP entre el cliente solicitando el servicio y el agente servidor DRP. (Ver figura a continuación)

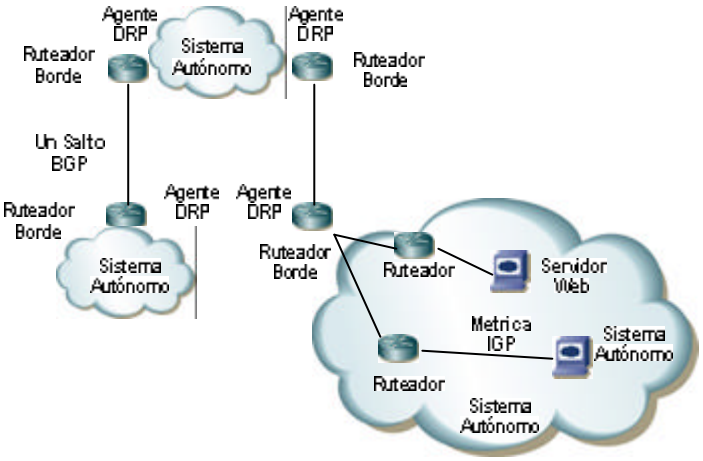
- **DRP interno**. Esta métrica encuentra la distancia IGP entre del agente servidor DRP y el ruteador de borde más cercano al límite del sistema autónomo. Esta métrica es determinada por el protocolo IGP que este en uso. Normalmente, esta métrica está combinada con múltiples valores de métrica: ancho de banda, retardo, entre otras, que son combinadas en una sola métrica compuesta. Este es el valor que el director usa para tomar decisiones de la distancia topológica.

- **Servidor DRP.** Esta métrica encuentra el valor IGP entre el agente del servidor DRP y el servidor asociado. Estas métricas son determinadas por el protocolo IGP que se encuentre en uso. Normalmente, esta métrica esta combinada con múltiples valores de métrica: ancho de banda, retardo, entre otras, que son combinadas en una sola métrica compuesta. Este es el valor que el Director usa para tomar decisiones de la distancia topológica.

- **Aleatoriedad**. Esta métrica es usada para seleccionar aleatoria mente entre cada servidor distribuido. Esto no requiere información de tablas de ruteo. Esta métrica no dispara ninguna petición DRP.

- **Costo Administrativo**. Esta métrica especifica una preferencia estática de un servidor sobre otro. Esto es utilizado para colocar un servidor fuera de servicio. Esta métrica no dispara ninguna consulta DRP.

Figura 6. Métricas DRP internas y externas



Las métricas en la lista aplican por sub-dominio o nombre de servidor. Puede asignársele peso a las métricas de forma que una sea más importante que otra o priorizar las métricas de forma que si múltiples servidores cumplen iguales otras métricas se aplique para encontrar el mejor servidor, o ambos.

La efectividad de las métricas DRP externas esta determinada por la calidad de los datos en la tabla de ruteo BGP. La efectividad de las métricas DRP internas está determinada por la calidad de los datos en las tablas de ruteo

IGP. Todos los agentes de servidores DRP asignados a un sub-dominio o nombre de servidor deben usar el mismo tipo de IGP, tal como *Routing Information Protocol* (RIP o RIP2), *Interior Gateway Routing Protocol* (IGRP), o *Open Shortest Path First* (OSPF).

2.1.1.3. Disponibilidad de servidores

Cuando el parámetro de disponibilidad del servidor es habilitado para un servidor distribuido, el Director usa periódicamente, conexiones temporales TCP para verificar que el servidor este disponible y prevenir que el Director dirija a los clientes a un servidor que no puede responder.

2.2. Ruteador borde

Los ruteadores de borde están localizados en el perímetro de una red de comercio electrónico y proveen varias funciones. Los ruteadores de borde conectan un sitio de comercio electrónico a la Internet y anuncian a los sitios. A través del uso de protocolos de ruteo exterior, tal como *Border Gateway Prococol (BGP)*, los ruteadores de borde propagan las direcciones IP usadas en el frente de la red de comercio electrónico a la comunidad de Internet. Si existen conexiones redundantes a proveedores de Internet (ISP, por sus siglas en inglés) el protocolo BGP permite la distribución de carga a través de múltiples conexiones a Internet y *failover* a través de tales conexiones.

Los ruteadores de borde también provén servicios de seguridad preliminar. A través del uso de filtro de paquetes o listas de control de acceso extendidas (ACL, por sus siglas en inglés), los ruteadores de borde pueden bloquear

cualquier tráfico no deseado y permiten solo el tráfico deseado en la red de comercio electrónico. Por ejemplo, filtros pueden aplicarse en los ruteadores de borde para permitir solo tráfico HTTP Web, tráfico SSL, y tráfico DNS en la red. Los filtros pueden también ser aplicados para bloquear el tráfico con direcciones fuente de usuario inválidas que son un indicio de un posible ataque malicioso. Para servicios de seguridad adicionales, los ruteadores de borde también pueden proveer filtros *stateful*, lo cual registra el estado de todas las conexiones de red.

2.2.1. Ruteadores Cisco 7500

Los ruteadores de la serie 7500 de Cisco están enfocados para incrementar la disponibilidad durante caída de la red planeada o no. Dada la posición prominente en las redes a gran escala como un ruteador de borde, la alta disponibilidad es una característica importante para los clientes que lo demandan. Los ruteadores de borde no se benefician de la redundancia en la arquitectura de la topología de la red de la cual los ruteadores centrales si lo hacen, por lo anterior pueden ser un punto individual de falla en la red. Los clientes ven el downtime como su mayor obstáculo para los objetivos de su negocio y para la relación con sus clientes. De cualquier forma, no siempre es posible construir equipo y circuitos redundantes a través de la red completa.

- Aislar cualquier error en una parte del ruteador de tal forma que no afecte al resto del sistema.
- Permitir que cualquier procesador dañado se cambie a cualquier procesador redundante en el caso de una falla.
- Minimizar el tiempo de cambio entre procesadores.

Las características de alta disponibilidad de la serie de ruteadores 7500 incluyen:

- Cisco 7500 *Route Processor Redundancy* (RPR)
- Cisco 7500 *Fast Software Upgrade* (FSU)
- Cisco 7500 *Route Processor Redundancy+* (RPR+)
- Cisco 7500 *Single Line Card Reload* (SLCR)
- Cisco 7500 *Stateful SwitchOver* (SSO)
- Cisco 7500 Non-Stop Forwarding (NSF)

La redundancia es una parte clave la metodología para incrementar la disponibilidad del sistema. Cuando un *Route Switch Processor* (RSP) falla, el *Route Switch Processor* toma el control para que el sistema continuara procesando y reenviando. La alta disponibilidad de sistemas de Cisco (HSA por sus siglas en inglés, *Cisco High System Availability*) emplea esta metodología para tratar con la falla de procesadores en el ruteador e incrementa la disponibilidad del sistema. Sin embargo, hay aun áreas en este proceso que pueden ser optimizadas. En el proceso HSA, el tiempo desde la falla inicial a la transmisión del primer paquete puede ser dividido de la siguiente forma:

1. El tiempo para identificar la falla
2. El tiempo para cargar y levantar el *software* en el RSP en espera.
3. El tiempo para cargar la nueva configuración en el RSP en espera.
4. El tiempo para reiniciar y recargar las tarjetas.
5. El tiempo para cargar la nueva configuración en las tarjetas.
6. El tiempo para aprender las rutas, pasar el mensaje keepalive, y reenviar el tráfico.
7. Convergencia de rutas.

Este proceso es llamado *cold standby*, lo cual implica que el sistema completo perderá funcionalidad por el período que dure la restauración. Todo el tráfico fluyendo por el ruteador se pierde durante este tiempo. El beneficio de usar *cold standby* es que el dispositivo puede reiniciar sin intervención manual mediante el reinicio del RSP en espera tomando control del ruteador.

2.2.1.1. Redundancia de Procesador (RPR)

El Cisco 7500 implementa la característica RPR (por sus siglas en inglés *Route Processor Redundancy*) para eliminar los pasos 2 y 3 en el proceso de cambio de procesos, reduciendo el tiempo de recuperación de falla. El tiempo de recuperación se reduce puesto que el RSP en espera, ya empezó el proceso de arranque antes de tomar el control del ruteador. Esto se conoce como modo *warm standby*.

En el modo *warm standby*, cuando el ruteador se enciende, ambos RSP el activo y el de espera se encienden e inicializan. El RSP en espera efectúa la mayor parte del proceso de arranque, pero no completa los pasos finales. La inicialización tomará lugar como si todas las tarjetas fueron removidas por una inserción y remoción en línea (OIR por sus siglas en inglés, Online-insertion-and-removal). Si el RSP activo falla, el RSP en espera toma el control. El RSP en espera necesida solo completar los pasos finales del proceso de arranque cuando se convierte en RSP activo; entonces, se reduce el tiempo de recuperación. Las tarjetas son en ese momento insertadas en línea por el RSP en espera (ahora el RSP activo) durante el proceso de cambio. Esta nueva estrategia de cambio reduce el tiempo en un 50 por ciento comparado con el escenario *cold standby* (de 8-10 minutos baja a 4-5 minutos).

2.2.1.2. Procesador de redundancia + (RPR+)

Construido sobre la característica RPR, la característica RPR+ elimina los pasos 4 y 5 en el proceso de cambio. El RPR+ en el Cisco 7500 mantiene las tarjetas arriba durante el cambio. Las tarjetas no serán recargadas o reiniciadas. Esta característica reduce el tiempo para cambiar el procesador del ruteador en un 90 por ciento (baja a 30-40 segundos) comparado a RPR.

2.2.1.3. Actualización rápida de *software*

RPR y RPR+ son usados para combatir las fallas no anticipadas de RSP. La actualización rápida de *software* es usada para incrementar la disponibilidad durante los tiempos fuera de línea planeados, tal como actualización de *software* y mantenimiento. Usando el mismo proceso como RPR, el tiempo fuera de línea planeado es dramáticamente reducido. En lugar de usar la misma imagen de Cisco IOS (sistema operativo del ruteador Cisco) en ambos el RSP activo y el RSP en espera, una imagen del *software* Cisco IOS puede cargarse en el RSP en espera antes del cambio. Esto tomará la misma cantidad de tiempo que el escenario RPR, incluyendo el tiempo de bajar, descomprimir e inicializar la imagen actualizada de IOS.

2.2.1.4. Carga de tarjetas individuales (SLCR)

Previo a esta característica de disponibilidad, cuando una tarjeta fallaba, el backplane completo era inactivo y todas las tarjetas eran recargadas, durante este tiempo ningún paquete era reenviado. Para incrementar la disponibilidad en

los ruteadores Cisco 7500, **SLCR** (por siglas en inglés, Single Line Card Reload) es usado como un nuevo proceso de recuperación. Hasta la ocurrencia de un error en una tarjeta individual, sola la tarjeta fallida es recargada- en lugar de todas las tarjetas de comunicaciones. Este nuevo proceso reduce dramáticamente el tiempo de recuperación de la falla de una tarjeta en un 85%.

2.2.1.5. *Stateful switchover* (SSO)

Esta característica, se basa en el RPR+, reduce el tiempo en los pasos 6 y 7. La característica de cambio *stateful* permite al RSP activo pasar la información de estado necesaria de los principales protocolos de ruteo e interfaces a el RSP en espera para el cambio, reduciendo el tiempo para el RSP en espera para aprender y converger las rutas.

2.2.1.6. *Non-stop forwarding* (NSF)

También basado en RPR+, *Non-Stop Forwarding* permite a los ruteadores con RSP redundantes continuar reenviando datos a el RSP en espera durante el cambio. Esta característica usa *Forwarding Information Base* **(FIB)** que esta actualizada al tiempo del cambio. Una vez los protocolos de ruteo converjan las tablas FIB se actualizan y las entradas de rutas viejas son borradas. Esta característica elimina el tiempo fuera de línea durante el cambio.

2.2.2. Hot standby router protocol

Una forma de lograr cerca del 100 por ciento de disponibilidad de la red es usando el *hot standby router protocol* **(HSRP)**, el cual provee redundancia para la red IP, asegurándose que el tráfico de usuario se recupere inmediata y transparentemente del primer salto fallido en los dispositivos de red o circuitos de acceso en los bordes. Al compartir las direcciones IP y las direcciones MAC (capa 2), dos o más ruteadores pueden actuar como un solo routeador virtual. Los miembros de este grupo de ruteadores virtuales continuamente intercambian mensajes de estado. De esta forma, un ruteador puede asumir la responsabilidad de ruteao de otro, no importando si este sale de línea en forma planeada o no planeada. Las computadoras continúan reenviando los paquetes IP a una dirección IP y MAC consistente, y el cambio del dispositivo haciendo el ruteo transparente.

2.2.2.1. Operación HSRP

Un conjunto de ruteadores trabajan en conjunto, usando HSRP para presentar la ilusión de un solo ruteador virtual a las computadoras de la red local. Este conjunto es conocido como grupo HSRP o un grupo en espera. Un solo ruteador elegido del grupo es responsable de reenviar los paquetes que las computadoras envían al ruteador virtual. Este ruteador es conocido como el ruteador Activo. Otro ruteador es elegido como el ruteador en espera. En el caso de que el ruteador activo falle, el ruteador en espera asume el deber de reenviar paquetes del ruteador activo. Aunque un número arbitrario de ruteadores pueden ejecutar HSRP, solo el ruteador activo reenvía los paquetes enviados al ruteador virtual.

Para minimizar el tráfico de red, solo los ruteadores activo y en espera envían periódicamente mensajes HSRP una vez el protocolo a completado el proceso de elección. Si el ruteador activo falla, el ruteador en espera se convierte en el ruteador activo. Si el ruteador en espera falla o se vuelve el ruteador activo, entonces otro ruteador es elegido como ruteador en espera.

En una red local particular, múltiples grupos en espera pueden coexistir y mezclarse. Cada grupo en espera emula un solo ruteador virtual. Los ruteadores individuales pueden participar en múltiples grupos. En este caso, el ruteador mantiene un estado separado y contadores de tiempo para cada grupo. Cada grupo en espera tiene una sola, dirección MAC, como también dirección IP.

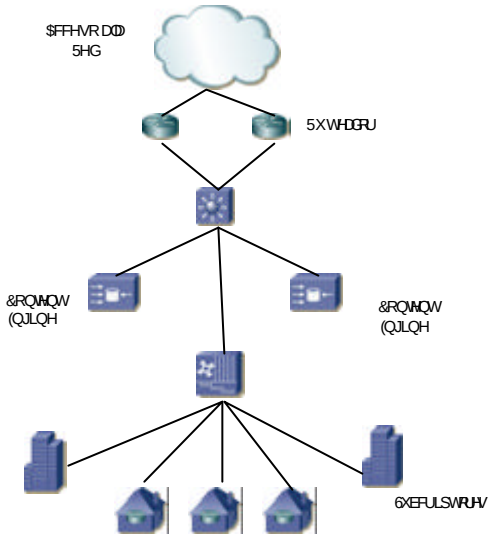
2.3. Caché de contenido

Los dispositivos de caché de contenido proveen servicios acelerados a los usuarios de comercio electrónico a través del aumento de la capacidad de los servidores *web* frontales para manejar las conexiones de los clientes. Los dispositivos de caché de contenido se sitúan al frente de los servidores *web* y manejar las peticiones de los usuarios de contenido estático. Estas soluciones son muy efectivas en ambientes que tienen un alto grado de contenido *web* estático. El contenido estático incluye gráficas, texto, y barras de herramientas.

En un ambiente de caché de contenido, las peticiones *web* de los usuarios son reenviadas a los dispositivos de caché. Si el contenido requerido puede colocarse en el caché**,** el dispositivo de caché llena la petición y almacena una copia local del contenido para peticiones futuras. Las peticiones futuras para el mismo contenido son manejadas directamente desde el dispositivo de caché.

Cuando los dispositivos de caché atienden las peticiones de los usuarios con contenido local, descargan el tráfico de los servidores *Web*. Esto ayuda a mejorar los tiempos para bajar contenido e incrementar la capacidad de los servidores *Web* para más sesiones interactivas.

Figura 7. Diagrama de red con múltiples caché contenido



2.3.1. Tolerancia y prevención de fallas

La solución de caché de Cisco provee tolerancia a fallas en el caché y en la red, eliminando punto singular de falla. En el caso que el dispositivo de caché falle, el tráfico es automáticamente redistribuido entre los otros dispositivos caché miembros del *cluster*. Si todos los dispositivos de caché en el *cluster* fallan, el tráfico ya no es redirigido a estos y el tráfico es reenviado hacia arriba.

La solución de caché de Cisco permite a un par de ruteadores compartir un *cluster* de dispositivos de caché, creando un sistema completamente redundante. En este escenario si un ruteador falla, existe un mecanismo que permite superar la falla; por ejemplo, un ruteador en espera puede dinámicamente tomar el control de la operación, redirigiendo las peticiones *web* al *cluster* de caché.

La tecnología de *cluster* de Cisco permite a cada miembro del caché trabajar en paralelo. Un *cluster* de caché mejora la escalabilidad, redundancia, y disponibilidad de la solución. En este momento se pueden colocar hasta 32 unidades de caché en un *cluster*.

Dentro de las características de tolerancia y prevención de fallas que poseen las unidades de caché Cisco se encuentran:

- Adaptación dinámica a fallas – Si un cache engine pasa a estar indisponible o inalcanzable, otro caché miembro del *cluster* rápidamente compensará esta falla (tolerancia a fallas). Si todos los cache engines en un *cluster* fallan, el ruteador con *Web* Cache Communication Protocol (WCCP, por sus siglas en inglés) habilitado detendrá la redirección de tráfico a el *cluster* de cache.

- Soporte a ruteadores WCCP *multihome* – múltiples ruteadores pueden compartir el mismo *cluster* de cache engine. Esta característica y el Cisco *Multigroup Hot Standby Router Protocol* (MHSRP por sus siglas en inglés) pueden ser usados en conjunto en los ruteadores para eliminar puntos individuales de falla en el sistema de caché.

WCCP es un protocolo de caché para ruteadores que permite localizar el tráfico de red y provee distribución de carga a través de múltiples caches de red.

2.4. *Switch* múltiples capas

Los *switches* de múltiples capas proveen la red principal de *switch*eo de un sitio de comercio electrónico, incluyendo la conectividad de los servidores *Web*, aplicación y base de datos. Por lo tanto, deben proveer alto rendimiento de *switch*eo en la capa 2 y capa 3 mientras soportan los servicios que cumplan los requerimientos de disponibilidad, escalabilidad y seguridad en un ambiente de comercio electrónico.

Por ejemplo, *switches* de múltiples capas deben soportar interfaces de alta velocidad, fuentes de poder redundante, calidad de servicio (QoS, por sus siglas en inglés), redes de área local virtuales (VLAN, por sus siglas en inglés), alta densidad de puertos, y una rápida recuperación de fallas. Además, los *switches* deben ser capaces de soportar un gran número de conexiones de usuarios mientras proveen en la capa 3 reenvío de millones de paquetes por segundo. Esto asegura que el *switch* no sea un cuello de botella en la arquitectura de la red de comercio electrónico.

Los *switch* de la familia Catalyst 6500 proveen inter-conectividad redundante para todos los dispositivos en Internet, servidores *web*, y los ruteadores adyacentes del proveedor de servicio de Internet. Usa VLAN para crear dominios separados de *broadcast*. Esta familia de *switches* soporta múltiples niveles de resistencia y servicio. Tolerancia a fallas a nivel de dispositivo, para incluir las siguientes opciones:

- Supervisor redundante
- Fuentes de poder con carga compartida redundante
- Ventiladores con carga compartida redundantes
- Relojes del sistema redundantes
- Enlaces redundantes
- *Switch Fabric* redundante

Todos los elementos del sistema incluyendo las fuentes de poder, ventiladores, supervisores, módulos de tarjetas, y *switch fabric* se pueden cambiar en caliente, por lo que los elementos pueden ser agregados, removidos o remplazados sin interrupción del servicio. En configuraciones duales de supervisores, Cisco *Fast-Switchover* puede transferir el control al supervisor redundante en segundos necesario para aplicaciones de misión crítica que requieren la máxima disponibilidad de la red. Todos los elementos del sistema pueden remplazarse en el campo reduciendo el tiempo fuera de línea.

Para una mayor confiabilidad a nivel de la red, la familia de *switches* Catalyst 6000 también soporta recuperación automática de una falla usando el protocolo *Spanning Tree* por VLAN, y soporte a balanceo de carga para convergencia de enlaces rápidos usando tecnología *Fast EtherChannel* o *Gigabit EtherChannel*.

2.4.1. Redundancia *switch fabric*

Desde su introducción, la serie Catalyst 6500 fue construida como un *switch* de arquitectura de un solo bus, lo cual provee la ruta de datos para todos los paquetes a través del sistema. La serie Catalyst 6500 también incluye un

crossbar switching fabric **(SFM)** como una arquitectura alternativa de *switching* a los altos requerimientos de ancho de banda. El SFM también provee otro nivel de redundancia de *hardware* al sistema. Las primera generación de tarjetas *fabric-enable* proveerán una conexión a ambos el *switching fabric* como al bus del sistema existente en el *backplane*. Esto permite a los sistemas 6500 usar el *switching fabric* como la primera opción para la transferencia de datos en las tarjetas *fabric-enable*. Si el *switch fabric* falla, el bus del sistema en el *backplane* tomará el control para asegurar que la función de *switching* continué, probablemente con un rendimiento menor, pero se mantendrá la red en línea.

2.4.2. Supervisor redundante

El supervisor dual provee redundancia de *hardware* para mantener la operación del *switch* al fallar uno de ellos. Un supervisor esta activo, mientras que el otro esta en espera. Cuando el supervisor activo es llevado fuera de línea o falla, el supervisor en espera toma el control del sistema.

Los dos supervisores en una configuración redundante de supervisores tienen diferentes responsabilidades. El supervisor activo es el responsable de controlar el bus del sistema y todas las tarjetas. Todos los protocolos están corriendo en el supervisor activo, y este efectúa todo el reenvió de paquetes. El supervisor en espera no se comunica con las tarjetas. Este recibe de la red los paquetes y llenas las tablas de reenvió con esta información, pero no participa en ningún reenvió de paquetes. Los protocolos relevantes en el sistema son inicializados, pero no son activados, en el supervisor en espera.

Los supervisores de la serie Catalyst 6500 se pueden cambiar en caliente, y el supervisor en espera puede ser instalado en un sistema activo sin afectar la

operación de la red. El supervisor redundante no efectúa balanceo de carga. El supervisor activo provee la inteligencia completa para el reenvió de paquetes para el sistema (redundancia N+1) Si el supervisor activo falla, el supervisor en espera puede mantener la misma carga del sistema. El supervisor en espera chequea periódicamente al supervisor activo vía un canal fuera de banda. El supervisor activo puede estar fuera de línea por varias razones, tal como fallas de *hardware*, condiciones de sobre carga del sistema, corrupción de la memoria, remoción desde el chasis, o simplemente por un reinicio del operador. El supervisor en espera detecta este tipo de falla y se convierte en el nuevo supervisor activo. El sistema operativo en el supervisor es el responsable de restaurar los protocolos, tarjetas, y colocar a los supervisores en operación normal. Esta restauración toma lugar vía un cambio rápido.

El mecanismo de cambio rápido permite a cada tarjeta saltar la respectiva carga de *software* y una porción de los diagnósticos, los cuales son parte de la inicialización del sistema. El cambio incluye reiniciar todos los protocolos (capa 2 y superior) como también reiniciar todos los puertos. El rendimiento resultante con los parámetros por omisión tomara alrededor de 60 segundos, dependiendo de los protocolos configurados. En una ambiente de red en producción, este cambio rápido de supervisor presenta una interrupción de la operación de la red.

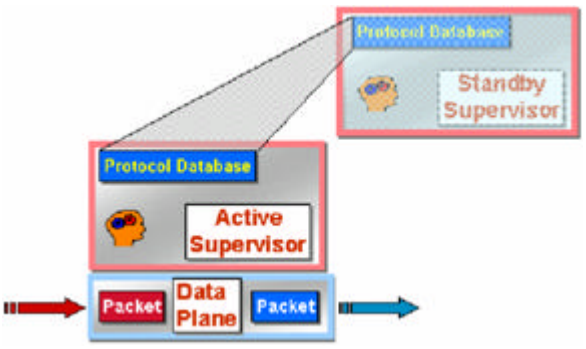
2.4.3. Características de alta disponibilidad del supervisor

Las características de alta disponibilidad del *software* del sistema operativo de los equipos Catalyst proveen redundancia a nivel de protocolo. Esta característica incluye dos funciones principales: redundancia de protocolo *stateful* y versiones de imágenes.

2.4.3.1. Redundancia de protocolo *stateful* en el supervisor

Con el cambio de alta disponibilidad del supervisor, el tiempo de cambio de un supervisor activo a uno en espera es minimizado a menos de 3 segundos para retornar a un estado de operación normal. Este tiempo fuera de línea minimizado es logrado por la sincronización de muchos de los protocolos de las capas 2, 3, y 4 entre en supervisor activo y el de espera. Esto se conoce como mantener el estado de los protocolos. Para una redundancia de protocolo *stateful* entre supervisores duales, una base de datos del estado de los protocolos es mantenida en cada supervisor para todos los protocolos y características que soportan alta disponibilidad. La mayoría de estos protocolos se ejecutan solo en el supervisor activo. En el caso de cambio de alta disponibilidad, el nuevo supervisor activo puede iniciar los protocolos desde la base de datos de estado actualizada en lugar de un estado de iniciación. De esta forma, un sistema con supervisor redundante puede mantener redundancia de protocolo *stateful* y minimizar el tiempo fuera de línea de la red cuando el supervisor activo va fuera de línea.

Figura 8. Redundancia protocolos entre los supervisores



2.4.3.2. Actualización de las imágenes de *software* en el supervisor

En una configuración redundante de supervisor, las imágenes del sistema operativo del *switch* necesitan administrarse propiamente para asegurar la alta disponibilidad del sistema. A continuación se describen algunas de las opciones para administrar las imágenes del sistema operativo de los *switches* Catalyst.

2.4.3.2.1. Sincronización de imágenes del supervisor

Por omisión en la serie Catalyst 6500, las imágenes de *software* del sistema operativo en el supervisor activo y en espera deben ser las mismas. Esto permita al sistema mantener un ambiente operativo estable para asegurar que el cambio de supervisor ocurre con las mismas características y revisiones de *software* en el nuevo supervisor activo como en el viejo supervisor activo. Si durante el inicio del sistema, las dos imágenes del supervisor no son de la misma versión, el supervisor activo descargara su imagen de carga actual al supervisor en espera.

La característica de sincronización de imágenes del sistema operativo de los Catalyst provee consistencia en el *software* entre los supervisores. Pero no permite actualizar el *software* sin llevar el sistema fuera de línea por un período extendido de tiempo. Para efectuar la actualización, el supervisor activo requiere un reinicio del supervisor para cargar la nueva versión de *software*. Después de este sincronizara las imágenes de *software* al supervisor en espera. Esto se efectúa típicamente durante un fuera de línea programado o una ventana de mantenimiento puesto que el sistema completo se iniciara en frío.

2.4.3.2.2. Característica de alta disponibilidad en el supervisor para versiones

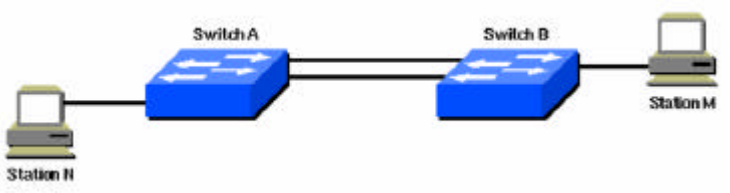
El manejo de versiones es la segunda porción de las características de alta disponibilidad del sistema operativo del Catalyst. Esta característica permite que diferentes imágenes pero compatibles estén ejecutando en los supervisores activo y en espera, deshabilitando el proceso por omisión de sincronización de imágenes del supervisor. Esta característica permite la actualización de la versión de *software* del supervisor "en vuelo" mediante el uso de cambio supervisor *stateful*. Este proceso se conoce como "actualización de *software* sin impacto", pero en realidad hay un pequeño "impacto" (menos de 3 segundos). Esto permite no solamente actualizar el *software* del sistema operativo del Catalyst sin reiniciar la caja completa, sino también mantener la versión previamente usada y probada del sistema operativo del Catalyst en el supervisor en espera como un plan de marcha atrás por s cualquier cosa sale mal en la actualización de *software*. No hay restricción en cual supervisor (activo/espera) puede ejecutar la versión de la imagen nueva/vieja.

2.4.4. Protocolo *spanning tree*

El protocolo *Spanning Tree* (STP por sus siglas en inglés) se ejecuta en puentes y swicth que cumplen con el estándar 802.1d. Hay diferentes versiones de STP, con el estándar IEEE 802.1d como el más popular y ampliamente implementado. STP es implementado en puentes y *switches* para prevenir lazos en la red. STP debe ser usado en situaciones donde se requieren enlaces redundantes, pero no lazos. Los enlaces redundantes son un respaldo importante en el caso de fallas en la red. Si un enlace primario falla, el enlace de respaldo es activado de forma que el usuario continué usando la red. Sin STP en

los puentes y *switches*, tal situación puede resultar en un lazo. Considere la siguiente red:

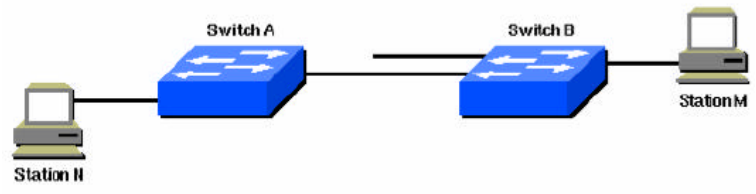
Figura 9. Enlaces redundantes entre un par de *switches*



En el diagrama de la red de arriba, un enlace redundante es planeado entre el *switch* A y B, pero esto crea la posibilidad de tener lazos en la red. Esto es porque, por ejemplo, un paquete de broadcasta o multicast transmitido desde la estación M y destinado para la estación N se mantendrá circulando una y otra vez entre ambos *switches*.

Sin embargo, con STP corriendo en ambos *switch*, la red lógicamente se ve como sigue:

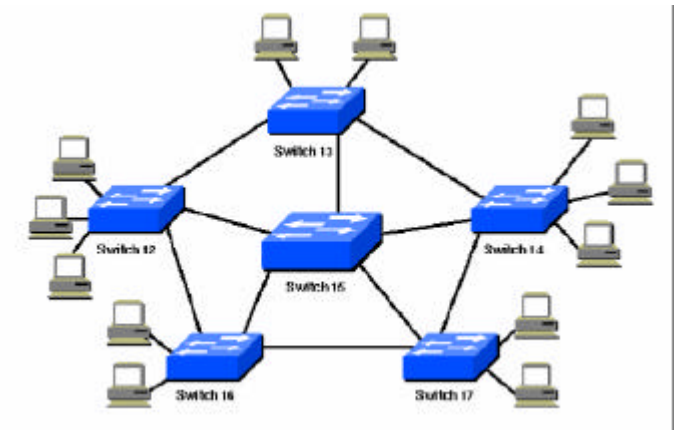
Figura 10. *Switch* con enlaces redundantes utilizando *spanning-tree-protocol*



Por ejemplo en el diagrama de red de abajo:

- El swicth 15 es el swicth central
- Los *switch* 12, 13, 14, 16, y 17 son de acceso, conectados directamente a las estaciones de trabajo.

Figura 11. Múltiples *switches* con rutas redundantes.



Para proveer este las rutas redundantes deseadas, como también para eliminar condiciones de lazos, STP define un árbol que se expande a todos los *switch* en una red extendida. STP fuerza a ciertas rutas de datos redundantes a un estado de bloqueo, mientras deja a los otros en un estado de reenvió. Si un enlace en estado de reenvió se muestra indispuesto, STP reconfigura la red y rerutea las rutas de datos para activar la ruta en bloqueo apropiada.

2.5. *Firewall*

Los *firewalls* proveen servicios de seguridad a través del control de conexiones. Estos son usados predominantemente cuando se protege datos de misión crítica o sensitiva de mucha importancia. Esto está típicamente en los servidores de bases de datos del *back-end* y los de aplicación. Los *firewalls* aseguran la comunicación a los servidores de aplicación y base de datos proveyendo de una inspección de todas las conexiones y permitiendo solo dispositivos autorización, tal como servidores *web*, para acceder los datos en los servidores.

Puesto que los *firewalls* protegen los datos más sensitivos, juegan un rol importante para alcanzar los servidores. Por ello, los *firewalls* son a menudo implementados en pares, mientras que uno es la unidad activa y la otra es la unidad en espera**.** En el caso de una falla la de la unidad activa, la unidad en espera pasa a estar operacional. Para asegurar que las conexiones a los servidores de aplicación y base de datos se mantengan en el caso de falla del *firewall*, los *firewall* deben ser capaces de efectuar un *stateful failover*.

2.5.1. ¿En qué consiste el *failover*?

Ambas unidades en el par de *failover* se comunican e intercambian datos sobre la identificación de las unidades primaria o secundaria, el estado de poder de la otra unidad, y otros datos necesarios para el *failover*. Las dos unidades intercambian paquetes especiales “hello” entre ellos, en todas las interfaces de red y el cable de *failover* cada 15 segundos. La característica de *failover* en los *firewall* Pix monitorea la comunicación *failover*, el estado del poder de la otra unidad, y los paquetes hello recibidos en todas las interfacews. Si dos paquetes

hello consecutivos no son recibidos dentro de un tiempo determinado, el *failover* empieza probando las interfaces para determinar que unidad fallo, y transfiere el control activo a la unidad en espera.

Cuando el *failover* ocurre, cada unidad cambia de estado. La nueva unidad activa asume las direcciones IP y MAC de la unidad activa anterior y empieza a aceptar tráfico. La nueva unidad en espera asume las direcciones IP y MAC de la unidad que previamente era la activa. Puesto que los dispositivos de red no ven cambio en estas direcciones, no cambian las entradas ARP o *timeout* en cualquier lugar de la red.

Cuando se usa *stateful failover*, el estado de las conexiones transmitido de la unidad Primaria a la unidad Secundaria. Sin un *stateful failover*, lo unidad en espera no mantendrá la información de cada conexión. Esto significa que todas las conexiones activas serán perdidas cuando el *failover* ocurra y todos los sistemas clientes deben restablecer las conexiones.

2.5.2. Stateful Failover

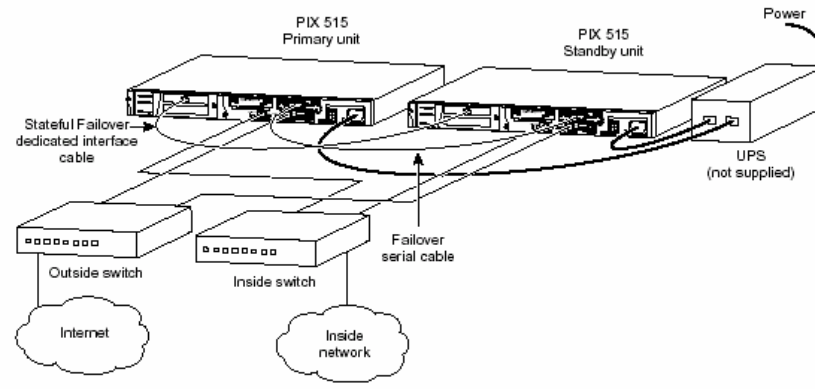
La característica *stateful failover* pasa información *stateful* por conexión a la unidad en espera. Después que el *failover* ocurre, la misma información de las conexiones está disponible en la nueva unidad activa. Las aplicaciones de usuario final no requieren reconectarse para mantener la misma comunicación de sesión. La información de estado pasa de la unidad en espera incluyendo el conjunto global de direcciones y estados, información de conexión y traducción y estados, los puertos H.323 UDP negociados, el mapa de bit de los puertos asignados para PAT (*Port Address Traslation*), y otros detalles necesarios para

permitir a la unidad en espera tomar el procesamiento cuando la unidad primaria falle.

Dependiendo de la falla, un *firewall* Pix puede tomar de 15 a 45 segundos para iniciar el cambio de unidad. Las aplicaciones que nos son manejadas por *stateful failover* requerirán tiempo para reconectarse antes que la unidad activa este completamente funcional.

En el caso de los *firewall* Pix se requiere una interfase de 100 Mbps Ethernet exclusiva para el *stateful failover*, esto servirá para que intercambien información de estado entre las dos unidades.

Figura 12. *Fire walls* en alta disponibilidad



2.5.3. Características del *stateful failover*

1. ¿Qué causa el *failover*?
   - Condición de falta de poder en el *firewall* activo
   - Reinicio del *firewall* activo.

- Enlace en el *firewall* activo baja por más de 30 segundos
- Operador activa el *failover*.
- El bloque de memoria se agota por 15 segundos consecutivos o más en la unidad activa.

2. ¿Qué información es replicada al *firewall* en espera?
   - La configuración
   - La tabla de conexiones TCP (excepto http) incluyendo información de *timeout* para cada conexión
   - Tabla de traducción
   - Tiempo que el sistema lleva arriba, el reloj del sistema es sincronizado en ambas unidades de *firewall*.

3. ¿Qué información es replicada al *firewall* en espera?
   - La configuración
   - La tabla de conexiones TCP (excepto http) incluyendo información de timeout para cada conexión
   - Tabla de traducción
   - La tabla ISAKMP y IPSec SA
   - La tabla ARP
   - Información de ruteo

2.5.4. Requerimientos de *hardware* y *software*

1. Dos unidades de *firewall* idénticas con un puerto dedicado *fast ethernet* para el *failover* requerido. Debe conectarse los puertos LAN para el

stateful failover en ambas unidades pix con un cable cruzado o a través de un *hub* o *switch*.

2. Para un mejor rendimiento se recomienda un Pix 515E para arriba.

3. Se necesita un cable de *failover* conectado a los dos puertos de *failover* en ambas unidades *firewall*.

4. Ambos *firewalls* deben correr la misma versión de *software*.

2.6. Balanceador de carga para servidores

Los balanceadores de carga para servidores ayudan a incrementar la escalabilidad de un sitio de comercio electrónico. El balanceador de carga de servidores trabaja distribuyendo las peticiones de usuario entre un grupo de servidores que aparecen como un solo servidor virtual al usuario final. Su principal función es reenviar el tráfico de usuario al servidor que "mejor" puede responder al usuario. Los balanceadores de carga de servidores usan mecanismos sofisticados para detectar el mejor servidor. Estos mecanismos incluyen encontrar el servidor con menos conexiones, la menor carga o el tiempo de respuesta más rápido. Estos también pueden detectar los servidores fallidos y automáticamente direcciona los usuarios a los servidores activos. Estos balanceadores de carga ayudan a maximizar el uso de los servidores y mejorar el tiempo de respuesta a los usuarios finales.

2.6.1. Cisco LocalDirector

El LocalDirector de Cisco Systems es una solución ideal de alta disponibilidad o *clustering* para aplicaciones *web* de misión crítica implementadas en una granja de servidores. La función crítica que efectúa el LocalDirector a una solución de *clustering* es la habilidad de automáticamente y transparentemente colocar los servidores y aplicaciones *web* en o fuera de servicio, basado en la disponibilidad momento a momento. Este identifica cuando una aplicación o contenido *web* no están disponibles y transparentemente rutea al cliente a la aplicación disponible. LocalDirector es un sistema para identificar la "salud" de los servidores. Este usa tres puntos para detectar la salud y disponibilidad de las aplicaciones y rutear a un nuevo servidor. Este sistema incluye:

1. Monitoreo de las negociaciones del protocolo control de transmisiones (TCP) entre cliente y servidor.

2. Verificación del contenido activo

3. Proveer un protocolo dinámico de retroalimentación (DFP, por sus siglas en inglés Dynamic Feedback Protocol) para la comunicación directa entre servidor y LocalDirector.

2.6.1.1. Monitoreo de la negociación TCP

El LocalDirector fisga en las negociaciones TCP/IP y el intercambio de datos entre el cliente y el servidor a nivel de puertos, analizando efectivamente la disponibilidad de la aplicación. Si este no esta disponible marca el servidor como fallido inmediatamente, lo cual reduce dramáticamente la probabilidad que un

cliente alcance un servidor “muerto”. Esta es la primera línea de defensa del administrador para asegurar que los clientes no sean enviados a un servidor o aplicación muerta. LocalDirector utiliza un método para detectar las aplicaciones fallidas de forma que el proceso es transparente y los clientes raramente son enviados a una aplicación muerta. Aun más importante, monitorea las negociaciones TCP proveyendo de un método de detección inmediata. En otras palabras, el sistema no debe esperar a que ocurra una prueba en una base periódica.

Si una aplicación o servidor no responde a una petición o esta respondiendo con TCP RST, el LocalDirector falla el servidor. Hay dos casos cuando un servidor real responde con un TCP RST:

- Un demonio sirviendo a este tipo de tráfico esta abajo (por ejemplo, el demonio HTTP en el puerto 80 fallo).
- El servidor esta muy ocupado para aceptar cualquier conexión.

LocalDirector lleva el registro de la cantidad de TCP RST. El usuario puede definir un valor límite para RST y de esta forma calibrar qué tan rápido el LocalDirector pondrá al servidor fuera de servicio. El LocalDirector también limita el número de conexiones enviadas a un servidor que no esta enviando datos. Cuando un servidor real alcanza un número de conexiones sin responder, LocalDirector chequea si hay otro servidor dentro del servidor virtual que también esté al 80 por ciento de su capacidad. Si el otro servidor está cerca de alcanzar este valor, el LocalDirector asume que el sitio está ocupado y no falla las conexiones al servidor. Similar a la forma en que se lleva registro de las RST, LocalDirector lleva el registro de los valores apropiados y puede colocar un límite para calibrar que tan rápido el LocalDirector pondrá un servidor fuera de servicio.

LocalDirector puede traer un servidor fallido a regreso a servicio inmediatamente si este responde con datos en una conexión existente. El LocalDirector coloca el servidor en un modo de prueba o transmisión, y si este responde a una nueva conexión viva, este es puesto de nuevo en servicio. Si el servidor no acepta la nueva conexión, entonces la conexión es marcada de nuevo como fallida.

2.6.1.2. Verificación de contenido

El sistema de verificación de contenido del LocalDirector pro-activamente prueba los servidores *web* con las peticiones HTTP que pueden ser definidas y personalizadas por el usuario. Este sistema puede determinar con extrema exactitud cuando las aplicaciones esta disponibles y pueden habilitar al LocalDirector para re-rutear a los clientes a los servidores disponibles.

Puede configurar el sistema de verificación de contenido para que sea ejecutado por cada servidor real asociado con un servidor virtual. Este sistema monitorea la salud de lo servidores *web*, aplicaciones, y contenido. El sistema consiste en un conjunto de pruebas de "salud" configurables que pueden ejecutarse individualmente para un servidor *web* o su contenido asociado.

Una prueba, la cual consiste en numerosos pasos, hechos a una petición http específica a el servidor, y evalúan la respuesta basada en un criterio de éxito definido por el usuario. Puede definirse con base en el resultado de la prueba, si el servidor real será llevado fuera de servicio, o un mensaje de advertencia será generado. Las pruebas pueden ser ejecutadas automáticamente por una programación definida por el usuario en todos los servidores reales bajo un servidor virtual, o estas pueden ser ejecutadas manualmente por el usuario en un

servidor real en particular. Cada paso de la prueba puede verificar las siguientes puntos en una URL en un servidor:

- Tiempo de respuesta de un servidor cuando se efectúa una petición a la URL.
- Respuesta de un servidor cuando una “galleta” en partícula son presentadas en la petición.
- Respuesta de un servidor a un agente del usuario en particular (tipo navegador)
- La operación correcta de cualquier esquema de autenticación http básico que proteja el acceso a las URL
- Cualquier re-dirección efectuado por el servidor cuando la URL es solicitada.
- Longitud del contenido regresado por el servidor
- Tipo de contenido regresado por el servidor
- Código de estado regresado por el servidor
- Presencia o ausencia de una cadena de caracteres en particular en el contenido regresado por el servidor.
- Presencia y contenido de cualquier “galleta” regresada por el servidor

Los sitios de comercio electrónico cada vez son más complejos y se múltiples capas son implementadas, esto requiere que pruebas pro-activas sean un requerimiento para cualquier sitio exitoso de comercio electrónico. Es especialmente importante en ambientes avanzados de comercio electrónico con tres capas que incluyen servidores *web*, servidores de aplicación y servidores de bases de datos. Esto puesto que la verificación activa de contenido puede identificar cuando un servidor *web* esta corriendo, pero la aplicación está muerta

o la base de datos no responde. También puede identificar cuando el contenido en cualquier capa está corrupto.

2.6.1.3. Protocolo de retroalimentación dinámico

El protocolo de retroalimentación dinámico de Cisco (DFP) es un mecanismo para que los servidores puedan proveer una retroalimentación inteligente a los dispositivos de balanceo de carga IP, como el LocalDirector. Es un mecanismo avanzado para asegurar que el LocalDirector no rutee tráfico a una aplicación muerta y que este tráfico sea enviado al servidor correcto. Agentes colocados en los servidores usan DFP para comunicar el estado de salud, disponibilidad y carga de la aplicación al LocalDirector.

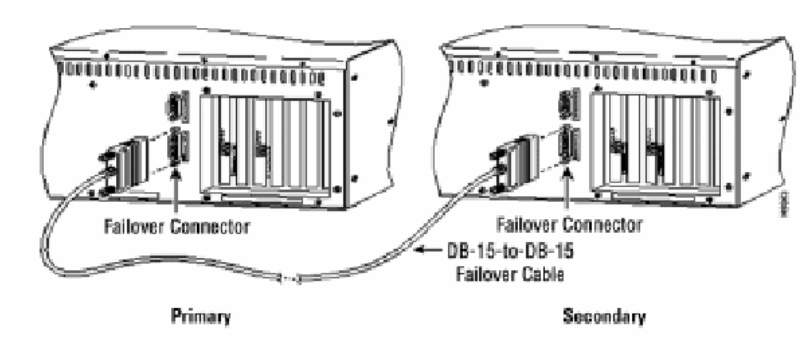
La habilidad de que múltiples agentes existan en los servidores proveen los siguientes beneficios:

- Los agentes pueden indicar a la red que un servidor esta congestionado.
- Los agentes pueden indicarle a la red que un servidor esta siendo sub-utilizado.
- Los agentes pueden informar a la red que un servidor no puede ser utilizado por un período de tiempo para balanceo de carga.
- Los agentes pueden instruir a la red que una aplicación debe tener precedencia sobre alguna aplicación genérica.
- Los agentes pueden efectuar verificación de contenido.

2.6.2. LocalDirector failover

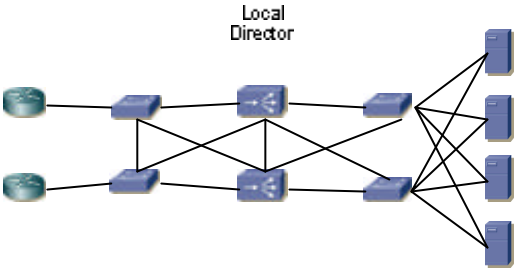
Una configuración de LocalDirector redundante consiste en dos equipos identicos que cumplen la misma funcionalidad. Las unidades son configuradas idénticamente, y son designadas como unidades “primaria” y “secundaria”. Cuando una falla ocurre, las unidades cambian de operaciones entre ellas, intercambian direcciones MAC, pero no se cambian designación “primaria” a “secundaria”, o viceversa.

Figura 13. Conexión LocalDirector en *failover*



En modo normal, la unidad primaria esta active, o es la unidad que esta procesando el flujo de datos de la red a través del LocalDirector. La unidad secundaria es la unidad en espera. Cuando ocurre una falla en la unidad activa, la designación “activa” y “en espera” se intercambian. Cuando la unidad fallida se levanta nuevamente, esta no se designa automáticamente como la unidad “activa”, aunque esta sea la unidad “primaria”.

Figura 14. Configuración LocalDirector en *Failover*



3. ALTA DISPONIBILIDAD EN LOS SERVIDORES

3.1. Componentes un Servidor

Los principales componentes de un servidor, cuyo funcionamiento es necesario para mantener la operación del mismo comprende los siguientes:

- Procesador, caché y *chipset*
- Arquitectura de memoria
- Arquitectura del bus
- Arquitectura PCI
    - Peer buses
    - NIC
- Sistema de almacenamiento

3.1.1. Procesador

El procesador es la parte de la computadora en la cual se efectúan las operaciones aritméticas y lógicas, y las instrucciones son decodificadas y ejecutadas. El procesador controla la operación de la computadora.

El procesador es el cerebro del procesador, y difieren por el conjunto de instrucciones, ancho de banda y la velocidad del reloj. La tecnología aplicada a los procesadores cambia rápidamente.

Figura 15: Procesador

La tecnología actual permite tener un procesador de respaldo fuera de línea, lo cual permitirá en el caso de una falla, reiniciar el servidor y continuar la operación con el procesador de respaldo. Aun cuando esto requiere detener la operación por varios minutos, da la capacidad de continuar la operación después de un breve lapso.

Dentro del contexto del procesador es importante también observar los modulos de poder de los procesadores (PPM por sus siglas en inglés), los cuales son los encargados de proveer poder a los procesadores y su falla, supone la falla del procesador. Por ello, debe observarse al adquirir un servidor con características de alta disponibilidad que cuente con PPM redundante

3.1.2. Chipsets

El *chipset* es un grupo de chips o circuitos integrados de la computadora, que cuando trabajan conjuntamente, manejan y controlan la computadora. Este conjunto incluye el procesador y otros chips que controlan el flujo de los datos a través del sistema. El *chipset* es el sistema nervioso central del servidor.

3.1.3. Buses de entrada/salida

Los buses de entrada/salida conectan a un dispositivo de entrada/salida al procesador; el bus estándar en la industria es el bus PCI.

El bus PCI es independiente del procesador y otras señales del bus de entrada/salida. Cualquier procesador o sistema de entrada/salida puede ser enlazado al bus PCI. El cual corre a una velocidad de 33 MHz o 66 MHz, con rutas de datos de 32 y 64 bit

Tabla III: Rendimiento del bus PCI & PCI-X

Ancho Bus /Frecuencia	Velocidad	Tipo Bus
32-bit/33MHz	133MB/s	PCI
32-bit/66MHz	267MB/s	PCI
64-bit/33MHz	267MB/s	PCI
64-bit/66MHZ	533MB/s	PCI
64-bit/133MHZ	1066MB/s	PCI-X

Los *slots PCI Hot Plug* permiten cambiar e instalar una nueva tarjeta PCI sin apagar la computadora; esta tecnología esta reservada para servidores de alta disponibilidad.

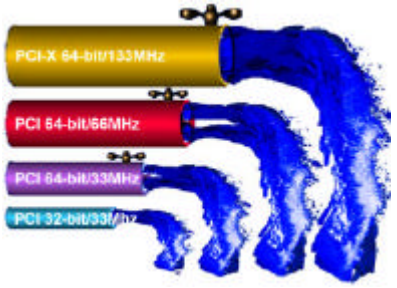
Para soportar la tecnología *PCI Hot Plug*, la tarjeta PCI debe cumplir con la especificación PCI 2.1. No se requiere ninguna alteración a la tarjeta PCI, pero se necesita modificar el *driver*. *PCI Hot Plug* y el sistema operativo trabajan juntos para permitir la acción de cambio en caliente:

- Remoción de tarjetas en caliente; permite apagar y remover permanentemente la tarjeta.

- Reemplazo en caliente de la tarjeta; permite el reemplazo de la tarjeta fallida con una tarjeta idéntica.

- Ver la información del *slot* detallada; permite ver la información detallada de cada uno de los *slot* PCI.

Figura 16. Comparación buses PCI & PCI-X



3.1.4. Ventiladores

Los ventiladores son usados para remover el calor de la computadora, si estos fallan el servidor se detendrá. Los servidores diseñados para alta disponibilidad tienen ventiladores redundantes que pueden cambiarse en caliente. Estos servidores tienen soluciones de monitoreó que notifican a los usuarios que un ventilador fallo. Los servidores en alta disponibilidad son diseñados para poder reemplazar el ventilador sin apagar el servidor.

3.1.5. Fuentes de poder

Es necesario determinar las necesidades de la fuente de poder de cada uno de los componentes instalados en la computadora. Identificar las necesidades es crítico para mantener la alta disponibilidad de la plataforma.

Un sistema de cómputo de alta disponibilidad debe tener la opción de proveer fuentes de poder redundantes en caso de una falla. El diseño para tener una fuente de poder más de lo que se requiere se refiere como *n+1*, el número de fuentes de poder necesarias más una para un respaldo. El sistema debe estar diseñado con la capacidad de remover las fuentes de poder para que puedan ser reemplazadas sin incurrir en detener el sistema.

Figura 17. Fuente de poder con capacidad de cambiarse en caliente

3.1.6. Almacenamiento

El almacenamiento se refiere a los dispositivos de entrada/salida para almacenar grandes cantidades de datos.

- Discos duros
- Arreglos de discos
- RAID
- Unidades de cintas
- Gabinetes de discos
- Sistema de almacenamiento
    - Almacenamiento de conexión directa (DAS por sus siglas en ingles *Direct Attached Storage*)
    - Almacenamiento conectado a la red (NAS por sus siglas en ingles *Network Attached Storage*)
    - Red de almacenamiento de datos (SAN por sus siglas en ingles *Storage Area Network*)

Figura 18. Sistema de almacenamiento externo

Más adelante se amplía este tema por la importancia que reviste, y el desarrollo tecnológico alrededor de él.

3.2. Tecnología para protección de memoria

Cada día más y más negocios dependen de servidores estándar en la industria para correr aplicaciones de misión crítica y que consumen recursos de memoria en forma intensiva. Esta tendencia a provocado que los sistemas operativos soporten más memoria y empujan a incrementar la capacidad de la memoria de los servidores a nuevos niveles. El sistema de memoria se ha vuelto más confiable con el transcurrir de los años debido a la mejora en los procesos de fabricación y tecnologías para protección de memoria como **ECC**. De cualquier forma, el incremento en la densidad de los componentes de memoria y la capacidad de memoria en los servidores para cumplir con la demanda, aumentan la probabilidad de que ocurran errores de memoria. Los errores de memoria pueden corromper los datos y causar una caída del servidor, resultando en perdida permanente de datos.

Para cumplir con los nuevos retos de confiabilidad, revisaremos tres tecnologías emergentes en la industria para proveer un incremento de la tolerancia a fallas para aplicaciones que requieren altos niveles de disponibilidad, las cuales son:

- *Online Spare Memory*,
- *Hot Plug Mirrored Memory*,
- *Hot Plug RAID (Redudant Array of Industry-standard DIMM) Memory*

3.2.1. Errores de memoria

Los módulos de memoria que se usan en los servidores son dispositivos de almacenamiento electrónico; por lo que, estos son susceptibles a errores. Las

computadoras usan dos tipos de dispositivos de memoria-RAM estática (SRAM por sus siglas en inglés) y RAM dinámica (DRAM por sus siglas en inglés). SRAM es usado para memoria cache puesto que es bastante rápida y retiene los datos aun cuando el poder es apagado. Los chips DRAM son instalados en módulos de memoria en línea duales (DIMM por sus siglas en ingles) de 168 pines. Cada chip DRAM almacena datos en columnas y filas de capacitares (celdas de memoria) que deben recargarse, o refrescarse continuamente, para preservar los datos. Un capacitor cargado representa un bit de datos “1” y un capacitor descargado representa un bit de datos “0”. El nivel de carga eléctrica es determinado por el voltaje de operación del dispositivo de memoria.

Si la carga de un capacitor es afectada por algún evento externo, los datos pueden ser incorrectos. Para servidores corriendo aplicaciones críticas, estos errores de memoria pueden causar una caída del servidor y pueden resultar en una perdida permanente de los datos. Los errores de memoria pueden clasificarse por el número de bits que son afectados-un solo bit o múltiples bits- y la causa del error.

3.2.1.1. Errores de un solo bit y múltiples bits

El bus de memoria es un circuito que cosiste en dos partes: el bus de datos y el bus de direcciones. El bus de datos es un conjunto de líneas que llevan los datos actuales a y desde SDRAM-cada línea lleva un bit de datos en un momento determinado. La computadoras actualmente tiene un ancho de bus de datos de 64 bit, lo cual significa que el bus transporta 64 bit simultáneamente. Estos 64 bit constituyen una palabra de datos. Un error en un bit de una palabra de datos es llamado un error un solo bit. Un error en más de un bit de una palabra de datos es llamado errores múltiples bit.

3.2.1.2. Errores duros y suaves

Dependiendo de la causa los errores de memoria se conocen como errores duros o suaves. Un error duro es causado por una pieza de *hardware* rota o defectuosa, por lo que el dispositivo consistentemente retorna resultados incorrectos. Por ejemplo, una celda de memoria puede estar pegada por lo que siempre retorna un bit "0", aun cuando se escriba un bit "1" en ella. Los errores duros son causados por defectos en la DRAM, malas soldaduras, problemas conectores, etcétera. Los errores suaves son más comunes. Estos ocurren aleatoria mente cuando una perturbación eléctrica cerca de la celda de la memoria altera la carga en el capacitor. Un error suave no indica un problema con un dispositivo de memoria puesto que una vez los datos almacenados son corregidos (por ejemplo, al escribir en una celda de memoria) el error no ocurre nuevamente.

3.2.1.3. Incremento en la probabilidad de errores de memoria

Hay dos factores que van de la mano, la densidad de almacenamiento de los chips DRAM y el voltaje de operación de los sistemas de memoria. Al reducirse el tamaño de las celdas de memoria, la densidad de almacenamiento DRAM se incrementa con la sensitividad al voltaje de las celdas de memoria. Hasta hace poco, los DIMM estándar en la industria operaban a 5 voltios. Por las mejoras en la densidad de almacenamiento DRAM, el voltaje de operación decreció a 3.3 voltios y luego a 2.5 voltios para permitir a la memoria operar rápido y consumir menos poder. Ahora, en el horizonte se encuentra DRAM II con tasa de datos doble a 1.8 voltios. Puesto que la densidad de la memoria se incrementa y el voltaje de operación se acorta, hay una alta probabilidad que un

error ocurra. Siempre que un bit de datos es mal interpretado o no corregido, el error puede causa que la aplicación se caiga.

3.2.2. Métodos para prevenir errores de memoria

Hay dos formas de protegerse contra los errores de memoria: pruebas y el uso de tecnologías de detección/corrección. La calidad de los procedimientos de prueba depende de la fuente de los módulos de memoria y reduce la probabilidad de tener errores duros.

3.2.2.1. Tecnologías de detección/corrección

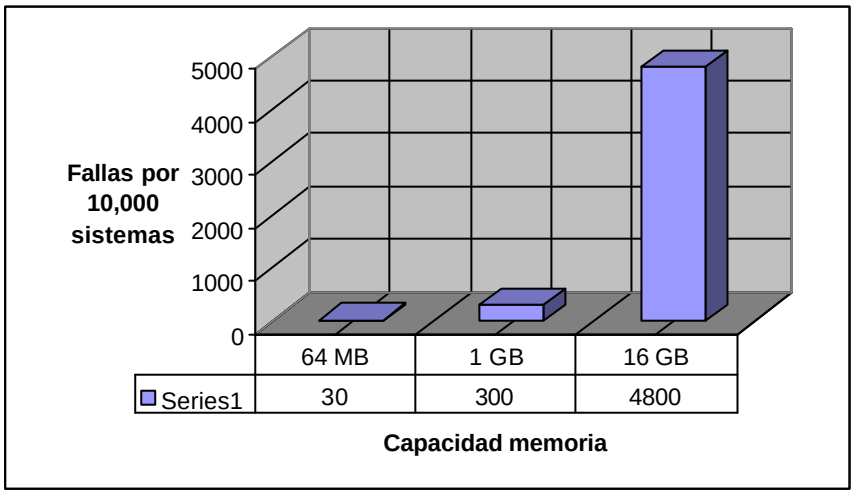
Los errores suaves no pueden prevenirse mediante pruebas realizadas por los fabricantes de los productos en sus plantas, y tomando en cuenta el incremento en la capacidad de los dispositivos de memoria, la probabilidad que ocurran errores suaves se incremento; por ello, se requieren de tecnologías de detección y corrección.

3.2.2.1.1. Memoria ECC

ECC detecta errores de uno y múltiples bits en una palabra de datos de 64 bits, y puede corregir errores de un solo bit. Además de detectar y corregir errores en un solo bit, ECC puede detectar (pero no corregir) errores en dos bits aleatorios y hasta cuatro bits dentro de un mismo chip DRAM.

La tecnología ECC puede proveer protección adecuada a muchas aplicaciones, sin embargo, la efectividad de la protección ECC decrece (las caídas de los servidores se incrementan de un 3 por ciento a un 48 por ciento) al incrementarse la capacidad de la memoria. Este hecho es significante por que hay dos factores que están dirigiendo a los servidores estándar en la industria a incrementar la capacidad de la memoria: los sistemas operativos ahora soportan grandes cantidades de memoria y el bajo costo.

Figura 19. Caídas de servidores por fallas de memoria en un año



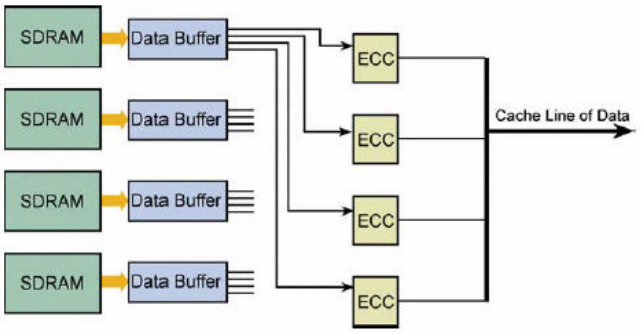
Fuente: Datos de IBM Chipkill Memory white paper, 1/99

3.2.2.1.2. Memoria ECC avanzada

Los dispositivos estándar ECC pueden corregir un error en un solo bit durante la lectura desde un DIMM. ECC Avanzado puede corregir errores de múltiples bit que pueden ocurrir en un chip DRAM y, por lo tanto, corregir la falla completa en un chip DRAM.

Un dispositivo estándar ECC puede corregir un error de un solo bit durante una lectura desde un DIMM. Mediante ECC avanzado se pueden corregir errores en múltiples bit que ocurran en un solo chip DRAM y por lo tanto podrán corregir la falla en el chip DRAM. En ECC avanzado con dispositivos de memoria de 4 bit (x4), cada chip contribuye con cuatro bits de datos a la palabra de datos. Estos cuatro bit de cada chip son distribuidos a través de cuatro dispositivos ECC (un bit por cada dispositivo ECC), por lo que un error en un chip puede producir hasta cuatro errores separados de un solo bit. La figura a continuación muestra como un dispositivo ECC recibe cuatro bit de datos desde cuatro chips DRAM.

Figura 20. Memoria ECC avanzada



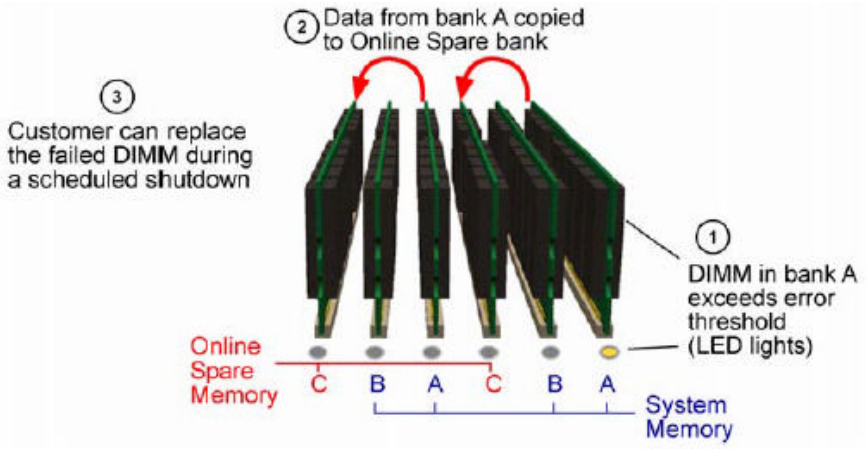
Puesto que cada dispositivo ECC puede corregir errores de un solo bit, ECC avanzado puede corregir actualmente errores de múltiples bit que ocurran en un solo chip DRAM.

3.2.2.2. Protección avanzada de memoria

3.2.2.2.1. Memoria en espera en línea

La memoria en espera en línea es muy recomendada en sitio que no tienen suficiente personal de tecnología disponible para atender una falla, no tienen siempre memoria de reemplazo a la mano, o donde los servidores no pueden ser bajados antes de lo programado. En este caso se designa un banco de memoria como memoria en espera en línea, y se designan a los bancos restantes como memoria del sistema. Cada banco de memoria puede contener dos DIMMs estándar de la industria. El banco de memoria en espera en línea entrará en funcionamiento cuando un banco de memoria del sistema alcance un límite de errores predefinido, copiando el contenido del banco de la memoria del sistema al banco de memoria en espera en línea. Esto mantendrá la disponibilidad y confiabilidad de la memoria sin requerir servicio inmediato. El DIMM que exceda el límite de errores puede ser reemplazado a conveniencia durante una parada programada del sistema. La memoria en espera en línea ofrece un mayor nivel de protección que la memoria ECC Avanzada.

Figura 21. Falla de DIMM en modo de *online spare memory*



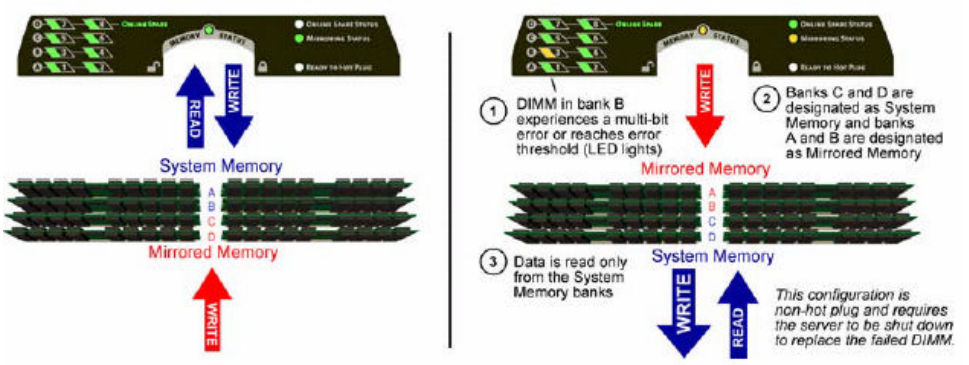
3.2.2.2.2. Memoria hot plug en espejo

La memoria en espejo *hot plug* es una opción de memoria tolerante a fallas que provee un mayor nivel de disponibilidad que la memoria en línea en espera. Si un módulo de memoria alcanza el límite predefinido de errores corregibles de memoria, existe la posibilidad de encontrar un incremento dramático de errores de *bit* y el servidor operan con un alto riesgo de falla hasta que la memoria degradada sea reemplazada. Por esta razón, la memoria en espejo *hot plug* es beneficiosa en situaciones en las cuales no puede permitirse tiempo fuera de línea y no puede arriesgarse a esperar la próxima detenida del sistema que se tiene programada.

Cuando un servidor está configurado con memoria en espejo *hot plug*, los datos son escritos a dos grupos de DIMMs estándar de la industria. Los datos

son leídos de un grupo de DIMMs mientras que el otro grupo contiene una copia exacta de los datos. Por esta razón, los dos grupos deben ser configurados idénticamente. Si un error de lectura se encuentra en un DIMM, o si el DIMM alcanza un límite predefinido, los datos son leídos del DIMM en espejo. Es recomendable utilizar esta tecnología en conjunto con el ECC Avanzado para mejorar la disponibilidad del equipo.

Figura 22. Memoria en espejo



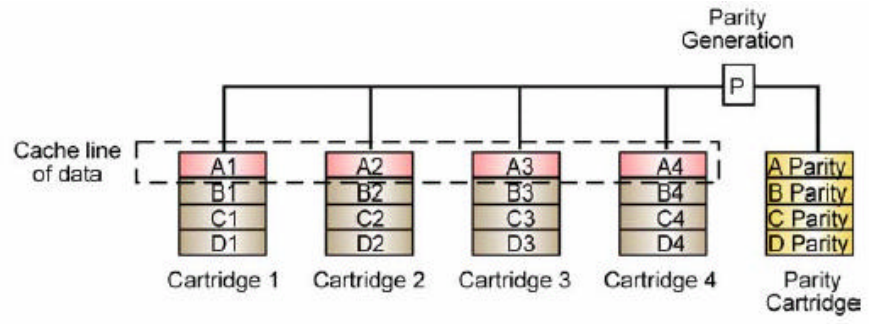
En la ilustración anterior, se muestra el modo de memoria en espejo usando una sola tarjeta de memoria en operación normal (izquierda). Si errores de múltiples *bit* ocurren (derecha), el sistema automáticamente designa al banco C y D como memoria del sistema y los bancos A y B como memoria espejo. Esto mantiene el nivel de disponibilidad del servidor.

3.2.2.2.3. Memoria *hot plug* en RAID

La memoria *hot plug* en RAID permite al subsistema de memoria continuar operando, aun cuando un dispositivo de memoria complete falle. RAID, en este caso viene del término en inglés *redundant array of industry-standard DIMMs*.

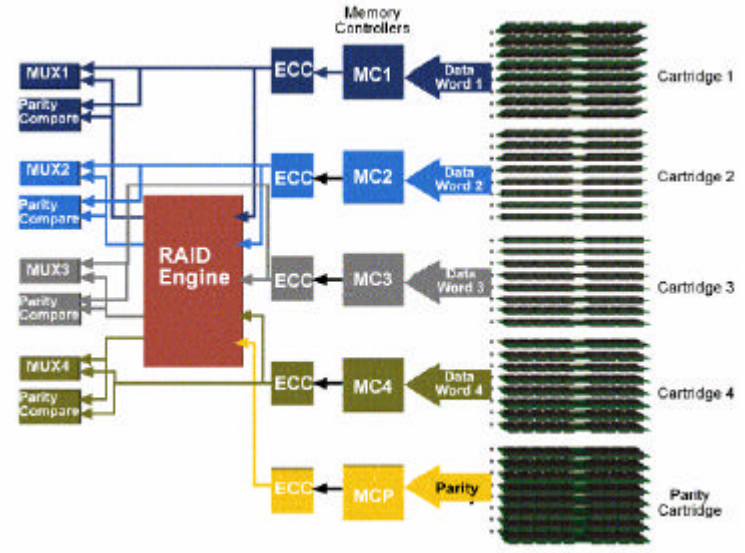
Al utilizar esta tecnología cuando un controlador de memoria necesita escribir datos a la memoria, este divide los datos en varios bloques, y lo escribe a varios cartuchos de memoria. Un proceso RAID calcula la información de paridad, y la escribe en un cartucho adicional. Con los cartuchos con los datos y el cartucho de paridad, el subsistema de datos es completamente redundante, si los datos de cualquier DIMM es incorrecta o cualquier cartucho es removido, los datos puede ser recreados de los cartuchos restantes.

Figura 23. Memoria RAID hot plug



En la ilustración anterior, la memoria RAID Compaq *hot plug* distribuye la línea de caché de datos a través de cuatro cartuchos y un motor de RAID calcula la información de paridad, la cual es almacenada en un quinto cartucho.

Figura 24. Arquitectura de memoria RAID



3.3. Tecnología para almacenamiento de datos

En esta sección examinaremos la tecnología de almacenamiento de datos disponible en la industria, y su correcta selección para una solución de alta disponibilidad. Nos concentraremos en la tecnología SCSI (por sus siglas en inglés *small computer system interface*), y HBA (por sus siglas en inglés *host bus adapter*) de fibra canal para la conexión de servidores a dispositivos de almacenamiento de datos. También cubriremos la SAN (por sus siglas en ingles *Storage Area Network*). No cubriremos la tecnología IDE (por sus siglas en inglés *integrated device electronic*), este no es recomendado para diseños de tolerancia a fallos.

3.3.1. Controladores SCSI para alta disponibilidad

Para construir un servidor en alta disponibilidad, puede usarse tarjetas controladoras SCSI para acceder a los discos SCSI. La mayoría de los servidores usan tarjetas controladoras SCSI para el sistema operativo y dependiendo del presupuesto usan tarjetas controladoras SCSI o HBA de fibra canal para el almacenamiento de los datos. Una tarjeta controladora SCSI es un dispositivo de entrada/salida que permite al sistema operativo acceder a los disco.

3.3.1.1. Guías para seleccionar una tarjeta controladora SCSI

Una tarjeta controladora para alta disponibilidad, debe presentar las siguientes características:

- **Capacidad de expansión *hot-pluggable*** esta característica permite colocar y sacar un disco del gabinete de almacenamiento mientras esta encendido el servidor.

Figura 25. Disco SCSI *hot-pluggable*

- **RAID de *hardware*** es la técnica que incrementa el rendimiento del disco en E/S, distribuyendo los datos en múltiples discos y protegiéndolos a través del uso de paridad o espejos.

- **Migración del nivel de RAID en línea** esta característica permite cambiar el nivel de RAID sin llevar los datos de producción fuera de línea, respaldándoles en cinta o reconstruyéndolos.

- **Redundancia *failover*** esta característica provee tolerancia a fallas en el acceso a los discos. Cada controladora SCSI contiene un caché espejo *writeback* y un caché espejo *prefetch* para prevenir la pérdida de datos en el caché en el evento de una falla.

- **Recuperación configuración ROM** esta característica permite respaldar y recuperar la configuración de las tarjetas controladoras SCSI.

- **Caché respaldado por batería** esta característica previene cualquier pérdida de datos que aun no han sido escritos a disco.

3.3.1.2. Ventajas y desventajas de seleccionar una controladora SCSI

Las principales ventajas de las tarjetas SCSI sobre la fibra canal son el precio, rendimiento, y compatibilidad.

- Precio

    - Los componentes SCSI requieren una inversión más baja inicial que los componentes de fibra canal.

    - Los componentes SCSI requieren una inversión menor que los componentes de fibra canal.

    - La tecnología SCSI permite compartir componentes comunes, entre las plataformas internas y externas.

- Rendimiento
    - En la evolución de la tecnología SCSI, la tasa de transferencia sea doblado entre cada generación sucesiva.

- Compatibilidad
    - La tecnología SCSI ofrece un alto nivel de ínter conectividad entre componentes del sistema.

Por otro lado sus principales desventajas son:

- La distancia máxima de los cables de 3.5 metros.
- El número máximo de dispositivos por controladora de 15 dispositivos.
- Ancho de banda es menor que la fibra.
- Solo soporta el protocolo SCSI.

3.3.2. HBA fibra canal para alta disponibilidad

La fibra canal es una tecnología de red para transmisión serial de alta velocidad.

Provee canales de alta velocidad a los dispositivos y una interfaz estandarizada entre los mismos. Dentro de sus principales características están las siguientes:

- Una nueva interfaz serial para los discos.
- Un sistema de interconexión de alta velocidad para la comunicación de servidor a servidor.
- Un medio de comunicación que es una alternativa a la tecnología de comunicación ofrecida en la red de área local, Ethernet o Gigabit Ethernet.

3.3.2.1. Guías para seleccionar un HBA de fibra canal para alta disponibilidad

En una solución de fibra canal de alta disponibilidad debe tomarse en cuenta las siguientes características:

- Tarjetas redundantes de fibra.
- *Switch* redundantes de fibra.
- Canales de fibra óptica redundantes.

3.3.2.2. Ventajas y desventajas de seleccionar HBA de fibra canal

Las ventajas de fibra canal incluyen lo siguiente:

- **Canales de alta velocidad**, con tasas de transmisión actualmente de hasta 2 Gbps por enlace en ambas direcciones.

- Diseñada para soportar discos hot pluggable simultáneos. Puede agregarse o removerse múltiples discos de un lazo activo sin interferir con la operación del lazo.

- Puede contener cientos de dispositivos usando tecnología FC-AL o SAN *switch* para crear ambientes SAN (por sus siglas en inglés *storage area network*)

- El soporte a varios protocolos de comunicación de datos incluye:

    - *Fiber distributed data interface* (FDDI)
    - *High-performance parallel interface* (HIPPI)
    - *Intelligent peripheral interface* (IPI)-3
    - *Internet protocol* (IP)
    - SCSI-3
    - *Ethernet*
    - *Token ring*
    - *Asynchronous transfer mode* (ATM)

Las desventajas de fibra canal incluyen las siguientes:

- La fibra es más frágil y requiere un cuidado extra en su implementación y mantenimiento.

- La mayoría de los gabinetes de almacenamiento con fibra canal no soportan la partición del sistema operativo. La fibra canal es usado para almacenamiento externo.

- La fibra canal no es una solución eficiente en costo para todas las situaciones de almacenamiento.

3.3.3. SAN para alta disponibilidad

3.3.3.1. Definición

La SNIA (por sus siglas en inglés *storage networking industry association*) ofrece la siguiente definición: (Refiérase a www.snia.org)

Esta es una red cuyo propósito primario es transferir datos entre los sistemas de computación y los elementos de almacenamiento y entre los elementos de almacenamiento. Una SAN consiste de una infraestructura de comunicaciones, la cual provee conexiones físicas y la capa de administración, la cual organiza las conexiones, elementos de almacenamiento y los sistemas cómputo para que la transferencia de datos sea segura y robusta.

El término SAN es usualmente (pero no necesariamente) identificado con el bloque de servicios de entrada/salida en lugar de los servicios de acceso a archivos. Un sistema de almacenamiento consiste en los elementos de

almacenamiento, dispositivos de almacenamiento, sistemas de cómputo, y/u otro dispositivo, más el *software* de control, y comunicaciones sobre la red.

La SAN esta dedicada completamente a almacenamiento y usan tecnología de interconexión de fibra canal. La SAN ofrece grandes ventajas en – mejorar el compartimiento de datos, consolidación, accesibilidad, administración, y protección de los recursos de cómputo de la organización.

La SAN (*storage area network*) es la capacidad de almacenamiento de datos que es acezada sobre una red de almacenamiento dedicada a este propósito. Esta red esta basada en tecnología de fibra canal, y descansa sobre la infraestructura de *switches*, *hubs*, y enlaces de interconexión para enlazar los componentes de almacenamiento. Típicamente conecta uno o más sistemas de uno o más fabricantes.

La SAN se compone de tres componentes principales:

- La infraestructura de tecnología fibra canal .
- Los sistemas de almacenamiento en los cuales los datos son almacenados y protegidos.
- *Software* de administración del almacenamiento y la SAN.

3.3.3.2. Guías para seleccionar un SAN para alta disponibilidad

Una SAN es recomendada generalmente en los siguientes casos:

- Sistemas heterogéneos con diferentes sistemas operativos para productos de almacenamiento de múltiples fabricantes.

- Es necesario tener rutas redundantes al sistema de almacenamiento.

3.3.3.3. Ventajas y desventajas para seleccionar una SAN

Las ventajas de la SAN incluyen las siguientes:

- Escalabilidad
- Rendimiento
- Tolerancia a desastres
- Respaldo y recuperación de datos
- Administración de datos
- Alta disponibilidad

Las desventajas de la SAN incluyen las siguientes:

- Más complejo de administrar que el almacenamiento conectado directamente.
- Mayor costo por *megabytes* que el almacenamiento SCSI.

3.3.3.4. Topología de SAN

Los diseños de topología SAN incluyen:

- Cascada
- Malla
- Anillo
- *Backbone*

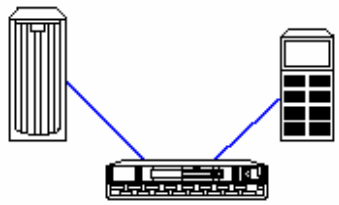
3.3.3.4.1. Topología un solo *switch*

La SAN más pequeña consiste en un solo *switch*, servidor y sistema de almacenamiento. Esta topología es un subconjunto de todas las otras topologías, y forma la base de un amplio rango de soluciones.

Un solo *switch* maximiza el rendimiento de la SAN, puesto que cada puerto en el *switch* tiene conectividad completa a cada otro puerto en el *switch*. Este diseño es también más fácil de instalar y configurar, puesto que no hay conexiones a otros *switch*.

Un ejemplo de una SAN simple con un solo *switch* se muestrá en la siguiente figura.

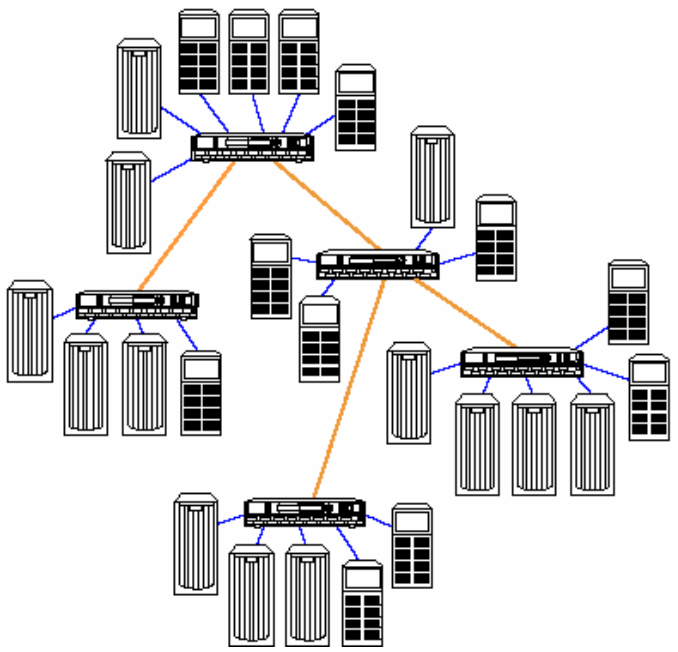
Figura 26. SAN con un solo *switch*



3.3.3.4.2. Topología en cascada

Una SAN en cascada es un conjunto de *switches* contectados entre ellos, por uno o más ISL (*inter-switch links* por sus siglas en inglés), en un arreglo de árbol. Ver la figura a continuación.

Figura 27. SAN en cascada



El diseño en cascada encaja bien en ambientes con acceso local a los datos; en estos casos, las peticiones de entrada/salida desde los servidores conectados a un *switch* son hechas por los sistemas de almacenamiento conectados al mismo *switch*. Grupos de servidores y sus sistemas de almacenamiento son conectados al mismo *switch* para proveer un alto nivel rendimiento de entrada/salida.

El diseño en cascada provee escalabilidad para la conectividad de servidores y almacenamiento adicional, y permite la centralización de la administración y respaldo, mientras mantiene un alto rendimiento de entrada/salida en el acceso local.

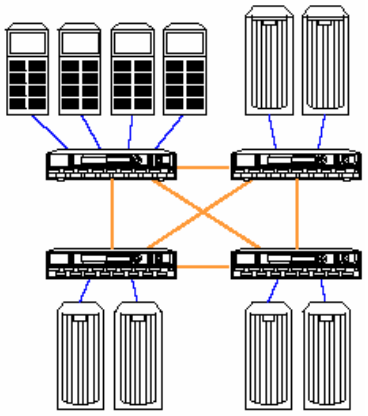
Ventajas del diseño en cascada

- Flexibilidad para acomodar diversas condiciones geográficas
- Fácil crecimiento
- Soporta respaldo compartido
- Soporta administración compartida
- El acceso local óptimo es inherente
- Más eficiente en el costo por puerto

3.3.3.4.3. Topología en malla

En un diseño en malla todos los *switch* son interconectados de tal forma que halla al menos dos rutas de cualquiera de los *switch* a cualquier otro *switch*. Este tipo de conectividad provee confiabilidad. Si un solo enlace entre los *switch* falla, las rutas se reconfiguraran automáticamente para que los datos viajen por una ruta alterna. Abajo, se muestra un ejemplo de un diseño en malla.

Figura 28. Diseño de una SAN en malla



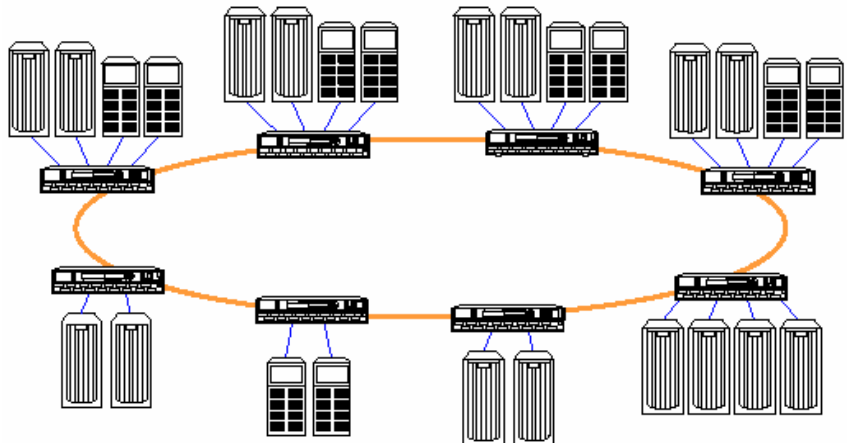
Ventajas del diseño en malla

- Puede ser configurado acceso local o cualquiera a cualquier, o una mezcla de ambos.
- Provee protección contra fallas de enlaces o puertos de *switch*.
- Crece fácilmente.
- Soporta un respaldo compartido.
- Soporta una administración compartida.
- Acceso óptimo distribuido es inherente a este diseño.

3.3.3.4.4. Topología en anillo

Un diseño en anillo es un anillo continuo de *switch* conectados entre ellos en una sola red; cada *switch* es conectado al *switch* adyacente, con el último *switch* en el anillo conectado al primero. Este arreglo de *switch* provee el mismo nivel de confiabilidad que el diseño en malla, con una conectividad completa y con al menos dos rutas internas. Ver la siguiente figura.

Figura 29. Topología de SAN en anillo



Un diseño en anillo se usa en aplicaciones donde el acceso a los datos es siempre localizado. Los servidores y el almacenamiento que es accesado esta en el mismo *switch* y la mayor parte del tráfico de datos es manejado con el *switch*.

Ventajas de la topología de anillo

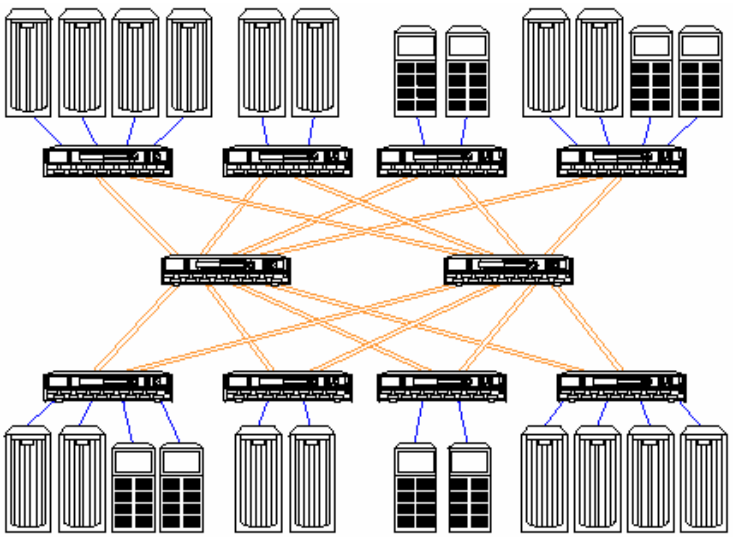
- Fácil de construir
- Crecimiento simple
- Soporta respaldo compartido
- Soporta administración compartida
- El acceso local óptimo es inherente a este diseño
- Diseño modular

3.3.3.4.5. Topología en *backbone*

Un diseño en *backbone* tiene uno o más *switch* dedicados principalmente a la conexión de otros *switch* dentro de la red. Los *switch* de *backbone* proveen alto ancho de banda y conectividad redundante a otros *switch*. Este tipo de implementaciones ofrecen la mejor conectividad "muchos a muchos".

Esta topología encaja bien en implementaciones donde el requerimiento primario es una conectividad completa "muchos a muchos" con alto rendimiento. Este es el diseño más conservador en casos donde los patrones de tráfico de entrada/salida no se conocen o varían.

Figura 30. Topología SAN en *backbone*



Ventajas de la topología en *backbone*

- Expansión eficiente de puertos: los nuevos *switch* solo necesitan ser conectados a los *switch backbone*.
- Todos los *switch* de los bordes están solo a dos saltos.
- Cuando se implementan con dos o más *switch backbone*, proveen redundancia en los *switch*.
- Pueden ser administrados centralmente.
- Conectividad completa "muchos a muchos" con ancho de banda distribuido y conectividad redundante.
- Mejora el ancho de banda con múltiples enlaces entre *switch* en paralelo.
- Ofrece una flexibilidad máxima para implementar tipos de acceso mixtos: local, distribuidos, o centralizados.
- Puede ser implementado o distribuido a través de una amplia área
- Puede ser implementado con capacidades de respaldo centralizada, reduciendo el costo de operación de respaldo y restauración
- Puede ser implementado con todos los niveles disponibles.

3.3.3.5. Disponibilidad de los datos en una SAN

La disponibilidad de los datos en una computadora esta influenciada por muchos factores, incluyendo las aflicciones de *software* y el sistema operativo en los servidores, el *hardware* de los servidores, la infraestructura SAN, y el almacenamiento primario y secundario.

En algunos ambientes, la disponibilidad adecuada de los datos es establecida por la rutina del procedimiento de respaldo en una base programada. En otros casos, el respaldo en línea dinámico de los datos a un sitio remoto es requerido; también se emplean *cluster*s de servidores y infraestructura de SAN redundante para lograr los objetivos de disponibilidad de los datos.

Cuando se considere la selección de la topología de la SAN, el número de *switches* de fibra canal y el número de enlaces entre *switches* tienen un gran efecto en la disponibilidad de los datos. El número de conexiones o rutas entre un servidor o *cluster* de servidores y la SAN, y el número de controladoras de almacenamiento o rutas a la SAN también afectan la disponibilidad de los datos.

Desde la perspectiva de la arquitectura de la SAN y la topología del diseño, la disponibilidad de la SAN puede ser clasificada en cuatro categorías o niveles; las diferentes categorías ofrecen un rango de niveles de disponibilidad que van desde el esquema de interconexión básico sin redundancia, hasta un diseño completamente redundante sin puntos de falla (NSPOF por sus siglas en inglés *no single point of failure*).

Niveles de disponibilidad

1. Una SAN no en malla / una ruta a los servidor y almacenamiento
2. Una SAN en malla o cascada / una ruta a los servidores servidor y almacenamiento
3. Una SAN en malla o cascada / múltiples rutas a los servidores y almacenamiento
4. Múltiples SAN / múltiples rutas a los servidores y almacenamiento

3.3.3.5.1. Nivel 1: una SAN no en malla / un ruta a los servidor y almacenamiento

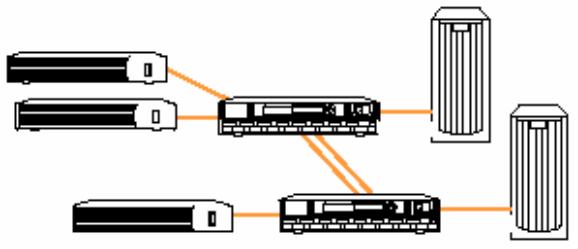
Estos diseños son implementados con un solo enlace entre cada *switch*, conectados a un SAN. El *switch* de fibra canal es dispuesto de tal forma que los servidores y almacenamiento conectados a la SAN usen una sola ruta; este tipo de diseño no provee ningún nivel de redundancia a nivel de la SAN o las rutas.

Figura 31. Nivel 1: Máxima conectividad

3.3.3.5.2. Nivel 2: una SAN en malla o cascada / un ruta a los servidor y almacenamiento

En este diseño hay más de un enlace entre *switch* y/o múltiples rutas a todos los *switch* en la SAN. Los servidores y almacenamientos conectados a la SAN usan una sola ruta. Si un solo puerto del *switch* o un enlace entre dos *switch* falla, la SAN automáticamente cambia la ruta de los datos a un enlace alterno. Los servidores no ven interrupción en el flujo de entrada/salida.

Figura 32. Nivel 2: infraestructura de SAN confiable

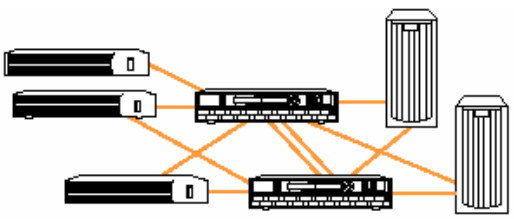


3.3.3.5.3. Nivel 3: una SAN en malla o cascada / múltiples rutas a los servidores y almacenamiento

Este diseño es el mismo que el nivel 2 con la adición de múltiples rutas de datos entre los servidores y el almacenamiento conectados en una SAN. En el caso que un *switch*, un adaptador de fibra en el servidor, o un enlace falle, los datos son automáticamente cambiados de ruta a un enlace alterno en los servidores y almacenamiento. Los servidores no ven interrupción en su flujo de entrada/salida.

Para asegurar la alta disponibilidad, cada adaptador de fibra en los servidores debe ser cableado a un *switch* diferente y debe ser configurado para acceder a una controladora diferente (en el sistema de almacenamiento) cuando se configura en modo de *failover* en múltiples bus.

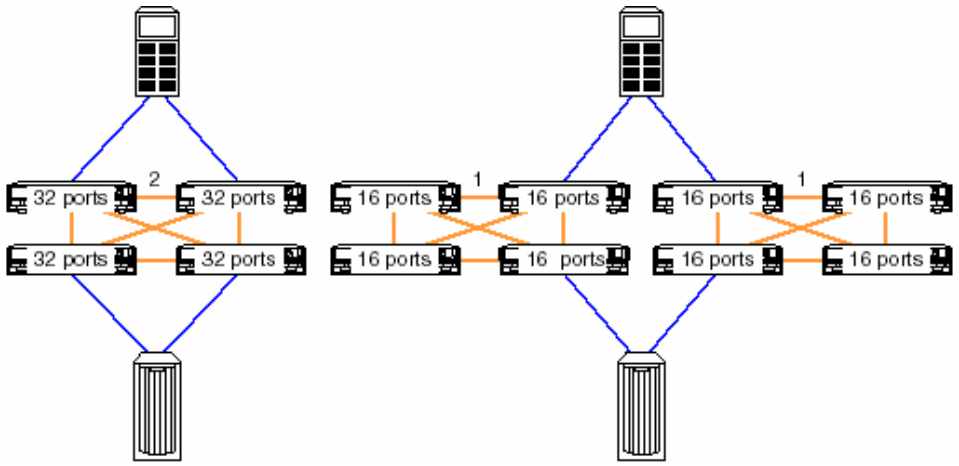
Figura 33. Nivel 3: alta disponibilidad en una SAN con múltiples enlaces



3.3.3.5.4. Nivel 4: múltiples SAN / múltiples rutas a los servidores y almacenamiento

Como el nivel 3, el nivel 4 provee múltiples rutas de datos entre los servidores y el almacenamiento, pero en el nivel 4 estas rutas son conectadas a SAN físicamente separadas. Este tipo de diseño provee el más alto nivel de disponibilidad y ofrece protección sin puntos de falla (NSPOF). Cualquier evento que pueda afectar el rendimiento o uso puede ser solucionado cambiando la ruta de datos a la otra SAN alterna. Los servidores no ven interrupción en su flujo de entrada/salida. El nivel 4 elimina cualquier vulnerabilidad a falla en la SAN.

Figura 34. Nivel 4: SAN sin puntos de falla



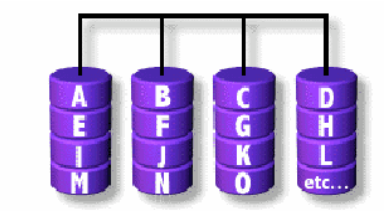
3.4. Niveles de RAID

Un RAID puede incrementar la disponibilidad mediante la protección de los datos cuando los discos fallan, y proveen gran rendimiento al acceder los datos. El RAID por sus siglas en inglés *Redundant Array of Independent Disks* es una técnica que incrementa el rendimiento de acceso a los datos por la distribución de los datos en múltiples discos y protegiendo los datos por el uso de paridad o imágenes.

Los niveles RAID son configuraciones diseñadas para optimizar el rendimiento o velocidad; cada nivel es diseñado para un acceso rápido a los datos, una mayor redundancia de datos, o una combinación de ambos; hay una variedad de niveles.

3.4.1. RAID 0: Arreglo de discos distribuido sin tolerancia a fallas

Figura 35. RAID 0



RAID 0 implementa un arreglo de disco distribuido, en el cual los datos son divididos en bloques y cada bloque es escrito a un disco separado; el nivel de RAID 0 requiere un mínimo de dos discos para ser implementado.

3.4.1.1. Ventajas

- El rendimiento de entrada/salida es altamente mejorado por la distribución de la carga de entrada/salida a través de muchos canales y discos.
- Un mejor rendimiento es logrado cuando los datos son distribuidos a través de múltiples controladoras con un solo disco por controladora.
- No involucra el cálculo de paridad.
- Un diseño muy simple.
- Fácil de implementar.

3.4.1.2. Desventajas

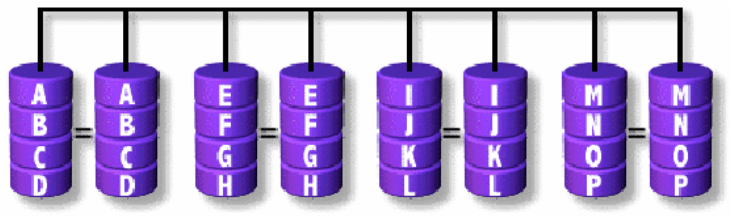
- No es un verdadero RAID puesto que no es tolerante a fallas.
- La falla en un solo disco puede resultar en la perdida de todos los datos en el arreglo.
- Nunca debe usarse en ambientes de misión crítica.

3.4.1.3. Aplicaciones recomendadas

- Producción y edición de video
- Edición de imágenes
- Aplicaciones de pre-impresión
- Cualquier aplicación que requiera alto ancho de banda, y no requiera tolerancia a fallas.

3.4.2. RAID 1: Discos Espejo

Figura 36. RAID 1



En nivel de RAID 1 requiere un mínimo de 2 discos para ser implementado. Para alto rendimiento, la controladora debe ser capaz de efectuar dos lecturas concurrentes separadas por par de controladoras en espejo o dos escrituras duplicadas por par de controladoras en espejo.

3.4.2.1. Ventajas

- Una escritura o dos lecturas son posibles por cada par en espejo
- Dos veces la tasa de lectura de un solo disco, la misma tasa de escritura que un solo disco.
- Redundancia 100% de los datos, lo cual significa que no es necesario re-construir los datos en caso de una falla de disco
- La tasa de transferencia por bloque es igual que la de un solo disco
- Bajo ciertas circunstancia, RAID 1 puede mantener fallas simultáneas de disco.
- Diseño simple subsistema de almacenamiento RAID

3.4.2.2. Desventajas

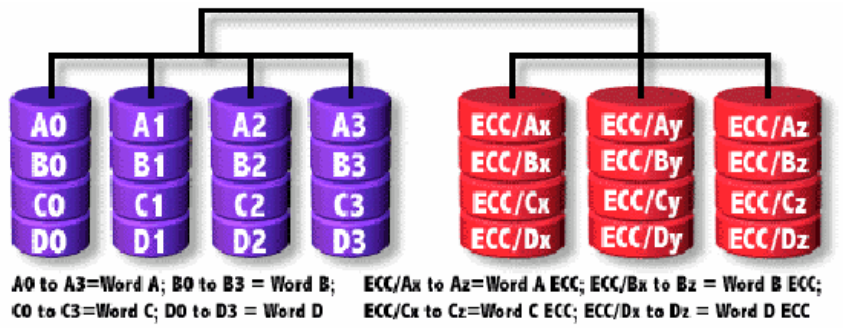
- Tiene el costo más alto en todos los tipos de RAID (100%) – ineficiente.
- La función RAID es efectuada por el *software* del sistema, lo cual carga el CPU del servidor y posiblemente degrade el rendimiento. Es altamente recomendable utilizar la implementación de *hardware*.
- Puede no soportar el cambio en caliente de un disco fallido cuando se trata de una implementación en *software*.

3.4.2.3. Aplicaciones recomendadas

- Contables
- Planillas
- Financieras
- Cualquier aplicación que requiere muy altos niveles de disponibilidad.

3.4.3. RAID 2: Código Hamming ECC

Figura 37. RAID 2



Cada *bit* de una palabra de datos es escrita a un disco de datos (4 en el ejemplo: 0 a 3). Cada palabra de datos tiene su palabra de código hamming ECC grabada en los discos ECC; en las lecturas, el código ECC verifica los datos correctos o corrige errores individuales de disco.

3.4.3.1. Ventajas

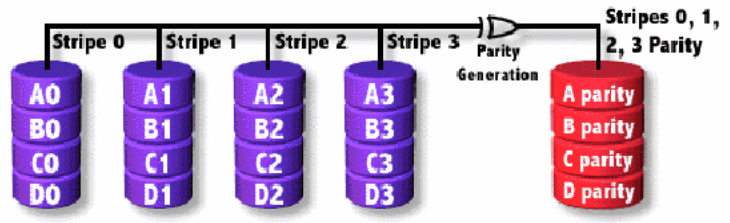
- Corrección errores de datos “en el vuelo”
- Tasas de transferencia de datos extremadamente altas
- Cuanto más alta la tasa de transferencia requerida, mejor la proporción de datos del disco para discos ECC.
- Diseño de controladora relativamente simple comparado a los niveles 3, 4, y 5

3.4.3.2. Desventajas

- Proporción muy alta de discos ECC a discos de datos con tamaños de palabras pequeñas – ineficiente
- Costo de entrada muy alto – requiere tasa muy altas de transferencia para justificarse.
- La tasa de las transacciones es igual a la de un solo disco en el mejor de los casos (con sincronización de *spindle*)
- No existen implementaciones comerciales / no es viable comercialmente.

3.4.4. RAID 3: Transferencia paralela con paridad

Figura 38: RAID 3



El bloque de datos es subdividido y escrito en los discos de datos. La paridad es generada en la escritura, y grabada en el disco de paridad y revisada en las lecturas. El nivel 3 de RAID requiere un mínimo de 3 discos para ser implementado.

3.4.4.1. Ventajas

- Tasas muy altas de transferencia para lectura y escritura
- Una falla de disco tiene un impacto insignificante en el rendimiento.
- Una proporción pequeña de discos de paridad a discos de data, lo cual significa alta eficiencia.

3.4.4.2. Desventajas

- La tasa de las transacciones es igual a la de un solo disco en el mejor de los casos (con sincronización de *spindle*)
- El diseño de las controladoras es moderadamente complejo

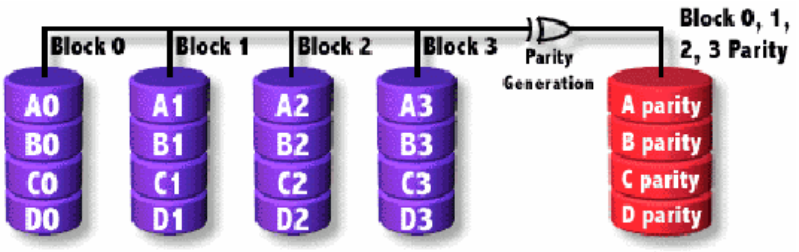
- Es bastante complicado y requiere muchos recursos cuando se implementa a nivel de *software*

3.4.4.3. Aplicaciones recomendadas

- Producción de video
- Edición de imágenes
- Edición de video
- Aplicaciones de pre-impresión
- Cualquier aplicación que requiera alto desempeño

3.4.5. RAID 4: Discos de datos independientes con disco de paridad compartida

Figura 39. RAID 4



Cada bloque entero es escrito en un disco de datos; la paridad para el mismo rango de bloques es generada en la escritura, grabada en el disco de paridad y chequeada en las lecturas. El nivel de RAID 4 requiere un mínimo de tres discos para implementarse.

3.4.5.1. Ventajas

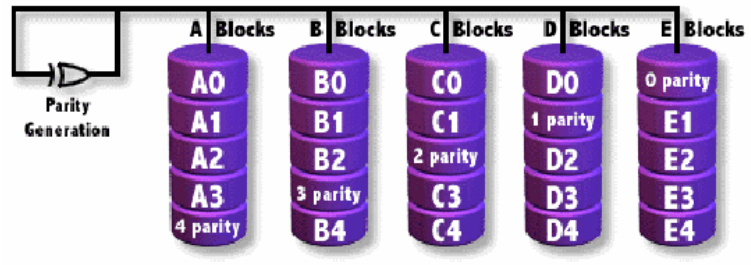
- Tasas muy altas de transferencia para lectura.
- Una proporción pequeña de discos de paridad a discos de data, lo cual significa alta eficiencia.
- Tasa de transferencia de lectura altamente acumulativa.

3.4.5.2. Desventajas

- El diseño de las controladoras es moderadamente complejo.
- Mala tasa transacciones de escritura y tasa transacciones acumulativa de escritura.
- Reconstrucción de datos difícil e ineficiente en el caso de una falla de disco.
- Tasa de transferencia de lectura de bloques igual a la de un solo disco.

3.4.6. RAID 5: Discos de datos independientes con bloques de paridad distribuidos

Figura 40: RAID 5



Cada bloque entero de datos es escrito en un disco de datos; la paridad para los bloques en el mismo rango es generado en la escritura, grabada en una ubicación distribuida y revisada en las lecturas. El nivel 5 de RAID requiere un mínimo de 3 discos para ser implementado.

3.4.6.1. Ventajas

- Tasa muy alta para transferencia de lectura.
- Tasa media para transacciones de escritura.
- Una proporción pequeña de discos de paridad a discos de data, lo cual significa alta eficiencia.
- Buena tasa de transferencia acumulativa.

3.4.6.2. Desventajas

- La falla de un disco tiene un impacto mediano en el rendimiento.
- Un diseño más complejo de controladora.
- Reconstrucción de datos difícil en el caso de una falla de disco (comparado al nivel 1 de RAID).
- Tasa de transferencia de bloques individuales igual a la de un solo disco.

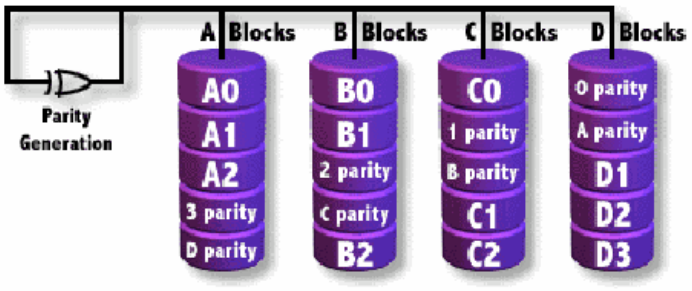
3.4.6.3. Aplicaciones recomendadas

- Servidores de archivos y aplicaciones
- Servidores de bases de datos.

- Servidores WWW, correo electrónico, y de noticias.
- Servidores de Intranet
- Nivel de RAID más versátil.

3.4.7. RAID 6: Discos de datos independientes con dos esquemas de paridad independientes distribuidos

Figura 41: RAID 6



El nivel de RAID 6 es esencialmente una extensión del nivel de RAID 5 el cual permite por la adición de tolerancia a fallas, usando un segundo esquema de paridad distribuida independiente (paridad dos dimensiones).

Los datos son distribuidos a nivel de bloque a través de un conjunto de discos, tal como un RAID 5, y un segundo conjunto de paridad es calculada y escrita a través de todos los discos; RAID 6 provee una alta tolerancia a fallas y puede mantenerse operando con fallas simultáneas en múltiples discos.

3.4.7.1. Ventajas

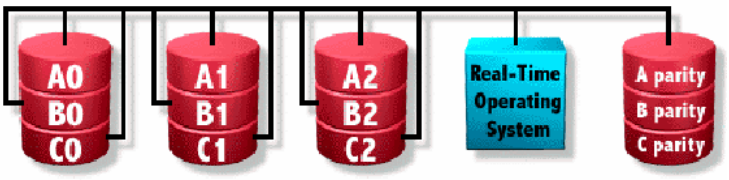
- Solución perfecta para aplicaciones de misión crítica.

3.4.7.2. Desventajas

- Diseño de controladora muy complejo.
- El costo de calcular las direcciones de paridad en la controladora es extremadamente alto.
- Rendimiento muy pobre en escritura.
- Requiere N+2 discos para implementarlo por el esquema de paridad de dos dimensiones.

3.4.8. RAID 7: Asincronía optimizada para altas tasas de entrada/salida

Figura 42: RAID 7



3.4.8.1. Características de la arquitectura:

- Todas las transferencias de entrada/salida son asíncronas, controladas y colocadas en el caché independientemente incluyendo las transferencias a la interfase el computador.
- Todas las lecturas y escrituras son colocadas en caché central vía un bus de alta velocidad.
- Un disco de paridad dedicado puede estar en cualquier canal.
- Sistema operativo en tiempo real orientado a procesos completamente implementado, residente en el microprocesador incrustado en la controladora de arreglo.
- Generación de la paridad integrada en el caché.

3.4.8.2. Ventajas

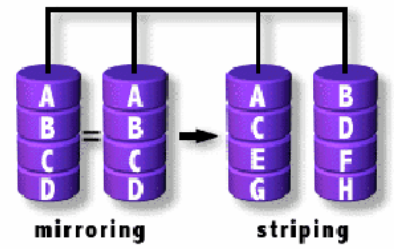
- El rendimiento total de escritura es 25% a 90% mejor que el rendimiento de un solo disco y 1.5 a 6 veces mejor que otros niveles de arreglos.
- Las interfases de la computadora pueden escalar para conectividad o incrementar el ancho de banda de transferencia al computador.
- Pequeñas lecturas en ambiente de múltiples usuarios tienen una tasa muy alta de coincidencia en el caché, resultando en un tiempo de acceso cercano a cero.
- El rendimiento de escritura mejora con el incremento en el número de discos en el arreglo.
- El tiempo de acceso decrece con cada incremento en el número actores en el arreglo.
- No requiere la transferencia de datos extra para la manipulación de la paridad.

3.4.8.3. Desventajas

- Solución propietaria de un fabricante.
- Extremadamente costosa por MB.
- Muy corta garantía.
- No puede dársele servicio por el usuario.
- La fuente de poder debe colocarse a un UPS para prevenir la perdida de datos del caché.

3.4.9. RAID 10: Muy alta confiabilidad combinada con alto desempeño

Figura 43: RAID 10



El RAID 10 es implementado como un arreglo distribuido, en el cual los segmentos son arreglos RAID 1. El RAID 10 tiene la misma tolerancia a fallas que el RAID nivel 1; el nivel de RAID 10 requiere un mínimo de 4 discos para ser implementado.

3.4.9.1. Ventajas

- El RAID 10 tiene el mismo sobre cargo para la tolerancia a fallas que un solo RAID 1.
- Tasas altas de entrada/salida pueden lograse por la distribución de segmentos RAID 1.
- En determinadas circunstancias, un arreglo RAID 10 puede mantener la operación con múltiples fallas de discos simultáneas.
- Excelente solución para sitios que de otra forma deberían usar RAID 1 pero necesitan algo que de un rendimiento adicional.

3.4.9.2. Desventajas

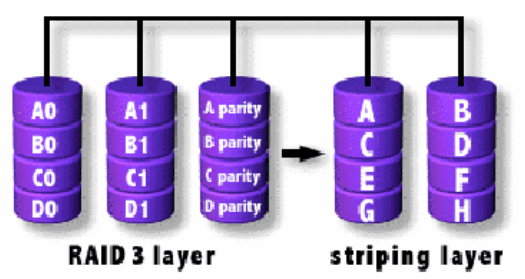
- Muy caro / alta sobre carga
- Todos los discos deben moverse en paralelo para reducir el rendimiento sostenido.
- Escalabilidad muy limitada a un muy alto costo

3.4.9.3. Aplicaciones recomendadas

- Servidores de bases de datos que requieren alto rendimiento y tolerancia a fallas.

3.4.10.RAID 53: Altas tasas de entrada/salida

Figura 44: RAID 53



El RAID 53 debería llamarse en realidad “RAID 03” puesto que es implementado como un arreglo distribuido (RAID 0) cuyos segmentos son arreglos en RAID 3. El RAID 53 tiene la misma tolerancia a fallas que el RAID 3 así como también la misma sobre carga; el RAID 53 requiere un mínimo de cinco discos para implementarse.

3.4.10.1. Ventajas

- Altas tasas de transferencia se logran gracias a los segmentos de arreglo en RAID 3.
- Altas tasas de entrada/salida para pequeñas peticiones se logran gracias al RAID 0.
- Puede ser una buena solución para sitios que de otra forma estarían con RAID 3 pero necesitan un rendimiento adicional.

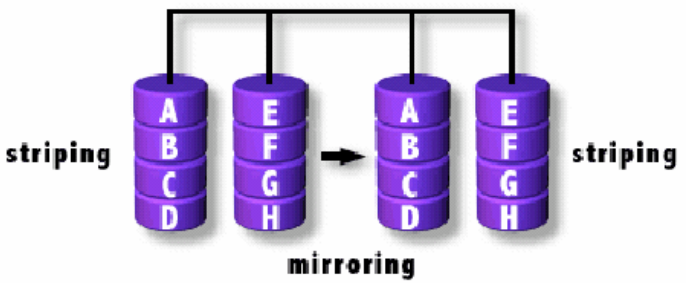
3.4.10.2. Desventajas

- Implementación costosa.

- Todos los *spindles* deben sincronizarse, lo cual limita la elección de los discos.
- La distribución de los *bytes* resulta en una pobre utilización de la capacidad formateada.

3.4.11.RAID 0+1: Alto desempeño de transferencia de datos

Figura 45: RAID 0+1



El RAID 0+1 es implementado como un arreglo en espejo cuyos segmentos son arreglos RAID 0 y el RAID 0+1 tiene la misma tolerancia a fallas que un RAID 5. El RAID 0+1 requiere un mínimo de cuatro discos para implementarse.

3.4.11.1. Ventajas

- El RAID 0+1 tiene la misma sobre carga que un RAID 1 (espejos).
- Altas tasas de entrada/salida son logradas gracias a los segmentos múltiples distribuidos.

- Una solución excelente para sitios que necesitan alto rendimiento pero no se preocupan por lograr la máxima confiabilidad.

3.4.11.2. Desventajas

- RAID 0+1 no debe ser confundido con el RAID 10. La falla de un solo disco puede causar que todo el arreglo se transforme, en esencia, en un arreglo RAID 0.
- Muy caro / alta sobrecarga
- Todos los discos deben moverse en paralelo para mantener un rendimiento sostenido
- Muy limitada escalabilidad a un alto costo

3.4.11.3. Aplicaciones recomendadas

- Aplicaciones de imágenes
- Servidor de archivos

4. SISTEMAS OPERATIVOS EN ALTA DISPONIBILIDAD

4.1. Cluster

La tecnología de *cluster* provee un ambiente redundante distribuido que aparenta y se comporta como un solo sistema virtual para los usuarios y administradores. El objetivo de un *cluster* es permitir compartir la carga computacional a través de varios servidores. No se requiere *hardware* especial para un *cluster*. La teoría es extensible a N servidores, por simplicidad para propósitos didácticos y de explicación en este capítulo se trabajan implementaciones con dos nodos.

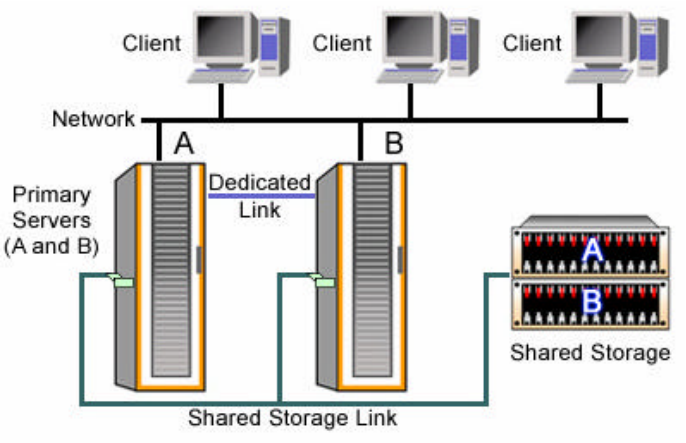
4.1.1. Componentes del *cluster*

El *cluster* es un grupo de sistemas independientes funcionando como un solo sistema, con cuatro componentes principales:

- Servidores estándar en la industria
- Rutas o interconexiones de comunicación entre nodos del *cluster* y el almacenamiento
- Almacenamiento físico compartido

- *Software* para *cluster*

Figura 46. Componentes del *cluster*



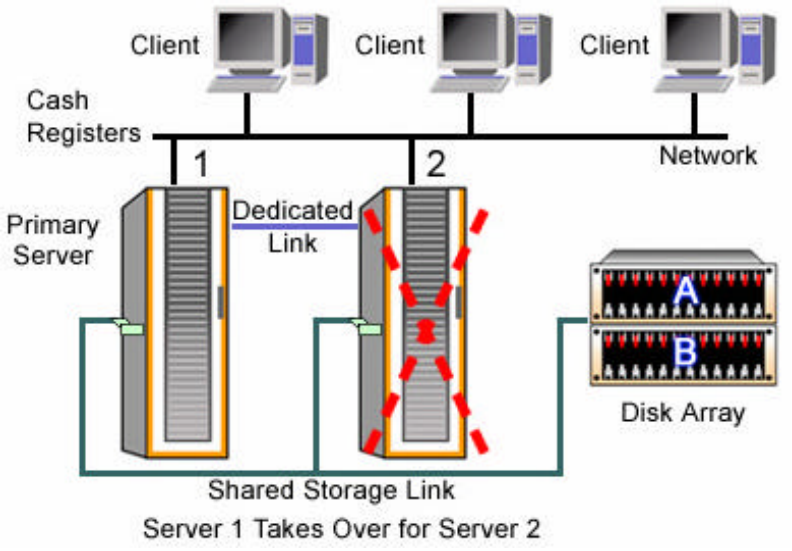
4.1.2. Beneficios del *cluster*

- **Alta disponibilidad**. El *cluster* permite incrementar la disponibilidad del sistema, al tener redundancia a nivel de nodos (computadoras).

- **Escalabilidad mejorada**. Un *cluster* permite agregar nodos adicionales al *cluster* lo cual le permite creer en su capacidad de procesamiento.

- **Balanceo de carga**. Un *cluster* con varios nodos permite distribuir la carga entre varios nodos que lo forman.

Figura 47: Ejemplo de disponibilidad de datos



4.1.3. Tecnología de *cluster*

4.1.3.1. Terminología

- **Nodos o sistemas**. Los servidores que integran un *cluster*, las computadoras individuales se conocen como nodos.

- **Servicio *cluster***. El servicio de *cluster* consiste en la colección de componentes en cada nodo que efectúa actividades de *cluster* específicas.

- **Recursos**. Los recursos son los componentes de *hardware* y *software* dentro del *cluster* que son administrados por el servicio de *cluster*.

- **Grupo de recursos**. Un grupo de recursos es una colección de recursos administrados por el servicio de *cluster*, como una sola unidad lógica.

- ***Failover***. El proceso por el cual los recursos de un nodo son transferidos a otro. El *failover* puede ocurrir automáticamente al ocurrir una falla de *hardware* o aplicación, o puede ser efectuado manualmente por la persona que administra el *cluster*. El algoritmo en ambas situaciones es idéntica, excepto que los recursos son apagados con delicadeza en un *failover* manual, mientras que son forzados a apagarse en una falla.

- **Failback**. Este proceso se produce cuando uno de los nodos regresa en línea, en ese momento el *software* que administra el *cluster* puede decidir mover algunos grupos de recursos al nodo recuperado.

4.1.3.2. Modelos de *cluster*

Hay dos modelos de *software* principales usados para *cluster*:

- Disco compartido
- Nada compartido

Ambos modelos pueden ser soportados en un mismo *cluster*.

4.1.3.2.1. Modelo de disco compartido

- Las aplicaciones y servicios que requieren solo acceso modesto a datos lectura compartido.
- Las aplicaciones y la carga son difíciles de dividir.

4.1.3.2.2. Modelo nada compartido

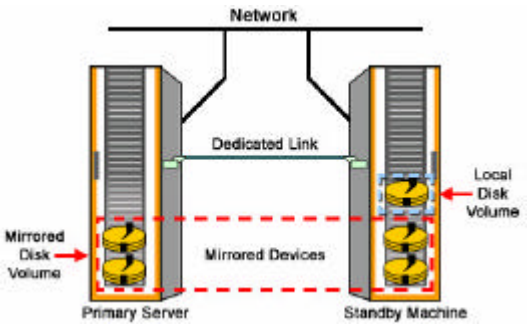
- Aplicaciones que requieren escalabilidad máxima.

4.1.4. Configuración activo/en espera

La configuración activo/en espera, incluye un servidor en espera y uno activo. El servidor en espera esta a la expectativa para efectuar un cambio en caso de falla del servidor activo; este último es el servidor primario.

Puede haber dos o más servidores activos con un servidor en espera. El proceso de *failover* es transparente. Esta configuración provee el más alto nivel de tolerancia a falla.

Figura 48. Configuración activa/en espera



4.1.4.1. Ventajas activo/en espera

- Espejo completo del servidor
- No hay degradación de rendimiento a el servidor primario
- *Failover* transparente automático
- *Hardware* estándar de la industria
- Mantenimiento en línea del *hardware*

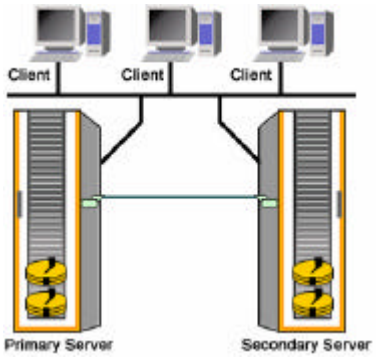
4.1.4.2. Consideraciones activo/en espera

- No es eficiente a nivel de costos
- No hay mejora en el rendimiento del sistema
- Perfil específico del usuario
- No es un *failover* instantáneo

4.1.5. Configuración activo/activo

En una configuración activo/activo cada servidor opera independientemente; si un servidor falla, los servidores restantes toman los servicios y aplicaciones del servidor que fallo. Es más eficiente a nivel de costos, puesto que todos los servidores proveen servicio; los servidores secundarios deben ser más poderosos para que puedan tomar los servicios del servidor fallido.

Figura 49: Configuración activo/activo



4.1.5.1. Ventajas activo/activo

- Usa tecnología *cluster*
- Usa todos los servidores
- Ofrece servidores pasivos que funcionan como servidores activos
- Usa *hardware* estándar de la industria
- Provee balance de carga.
- Soporta mantenimiento en vivo.
- Tiempo de *failover* cortó.

4.1.5.2. Consideraciones activo/activo

- Mayor licenciamiento requerido
- Configuración más compleja
- Degradación del rendimiento después del *failover*

4.2. *Cluster* para Microsoft® Windows®

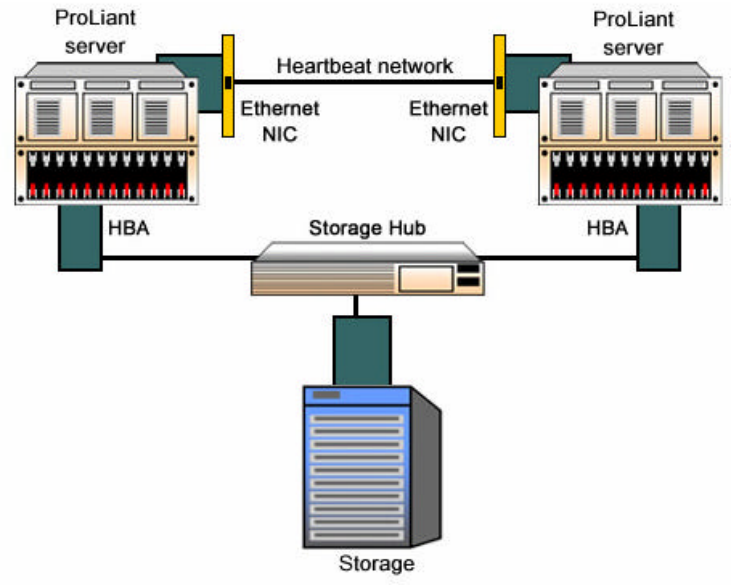
Tabla IV: diferencia sistemas operativos Microsoft® Windows® 2000

Windows® 2000	Windows 2000 Advanced Server	Windows® 2000 Datacenter Server
Sistema operativo multi-propósito para todo tamaño de negocio	Sistema operativo para aplicaciones de comercio electrónico y línea del negocio	Sistema operativo para los más altos niveles de disponibilidad y escalamiento
Escala de 1 a 4 procesadores, hasta 4 GB de memoria RAM	Escala de 1 a 8 procesadores, hasta 8 GB de memoria RAM	Escala de 1 a 32 procesadores, hasta 64 GB de memoria RAM
Confiable y disponible	Confiabilidad y disponibilidad extendida hasta dos nodos en *cluster*, 32 nodos en balanceo carga de red	Confiabilidad y disponibilidad máxima hasta 4 nodos en *cluster*, 32 nodos en balanceo carga de red

4.2.1. Configuración típica para un *cluster* Microsoft®

- La conexión *heartbeat* es única permite a los nodos detectar la disponibilidad de los otros nodos. Usa una conexión de red privada.
- Cuando un nodo falla.
    - La carga de trabajo es movida a los nodos restantes.
    - El *cluster* de discos es remontada en los nodos restantes.
    - Los clientes continúan accediendo a los recursos.

Figura 50: Configuración de un *cluster* Microsoft®



4.2.2. *Cluster* para Microsoft® Windows® 2000

Hay dos tipos de funciones de *cluster* disponibles con Windows® 2000 Server:

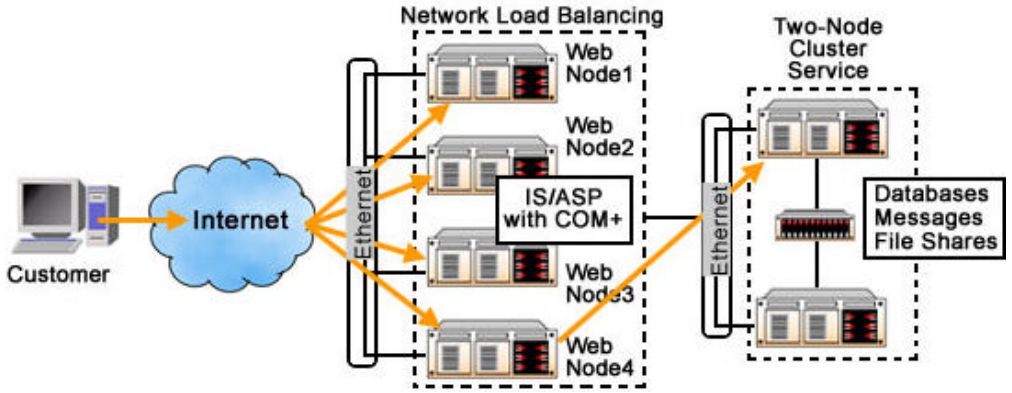
- Balanceo de carga de red
- *Cluster* de servidores

4.2.2.1. Balanceo de carga de red

- Provee una alta escalabilidad y disponibilidad para servi cios y aplicaciones basadas en TCP/IP.
- Combina hasta 32 servidores ejecutando Windows® 2000 advanced server en un solo *cluster*

- Instalado como un servicio de red estándar de Windows® 2000 advanced server o datacenter Server.
- Corre en la red existente.

Figura 51: Servicio *cluster* para balanceo carga red



4.2.2.2. *Cluster* de servidores

- Dos o más servidores que comparten recursos de almacenamiento, trabajan juntos y son administrados como una unidad.
- Los servidores ejecutan Windows® 2000 avanced server no importando si es un miembro activo o inactivo de un *cluster*.
- Crea grupos de recursos que no estan asociados a una computadora específica.
- Los grupos de recursos pueden hacer *failover* a otro nodo
- Cuando un nodo se cae, los clientes pueden usar el mismo método para acceder los recursos en otro nodo.

4.2.3. Mejoras al servicio *cluster* Windows® 2000 datacenter server

- Windows® 2000 datacenter server es recomendado para los niveles más demandantes de disponibilidad y escalabilidad.

- Datacenter derver soporta servidores con hasta 32 procesadores y hasta 64 GB de memoria RAM.

- Servicio de *cluster* que soporta la configuración de hasta cuatro servidores.

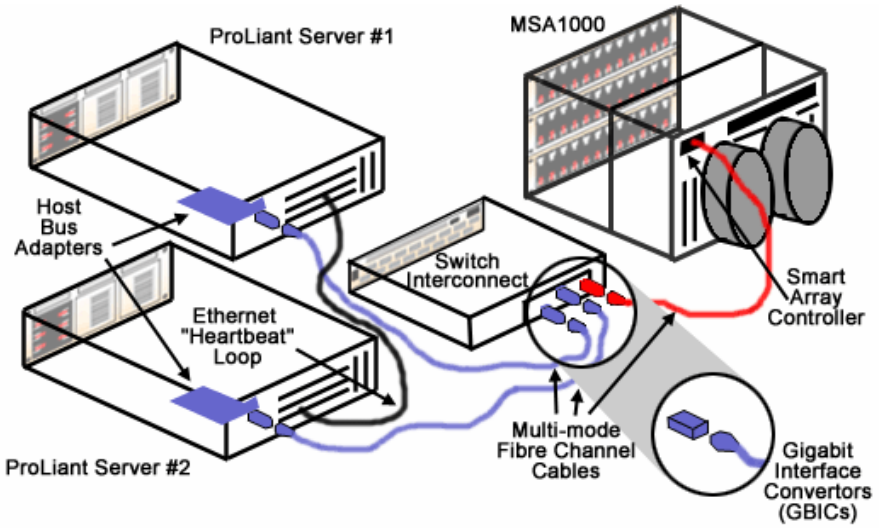
- Soporta fallas dobles o triples.

4.2.4. Servicio de *cluster* en Windows® 2000 advanced server

4.2.4.1. Solución de *cluster* con una sola ruta

En esta solución solo existe una ruta de interconexión entre el sistema de almacenamiento y los nodos del *cluster*. La falla en alguno de los componentes de comunicación (*switch* de fibra, tarjetas de fibra, cables de fibra) provocará la falla de uno de los nodos, o en el peor de los casos la falla del *cluster*, si la falla de comunicación es del sistema de almacenamiento.

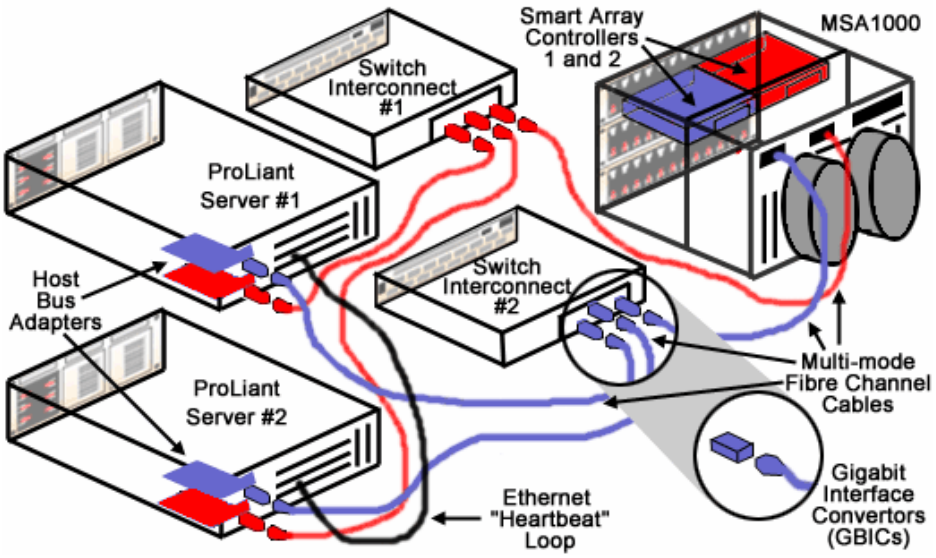
Figura 52: Solución de *cluster* con una sola ruta



4.2.4.2. Solución sin puntos de falla

En esta solución se han colocado componentes de interconexión redundantes en toda la solución. En este caso la falla de uno de los componentes de interconexión no provocará la falla del *cluster*. En el peor de los casos se requerirá redirigir la comunicación a otra ruta.

Figura 53: Solución de *cluster* sin puntos de falla



4.3. Novell® NetWare® *Cluster*

El servicio de *cluster* NetWare® es una solución de *cluster* para asegurar la alta disponibilidad y administración de recursos críticos de la red incluyendo datos (volúmenes), aplicaciones, licencias de servidores y servicios. Es un producto multi-nodo de *cluster* para NetWare® 5 y Netware® 6 que soporta *failover*, *failback* y balanceo de carga de recursos del *cluster* administrados individualmente.

Los *cluster* NetWare® soportan configuraciones de servidores mezcladas o heterogéneas. Esto lo hace excepcionalmente versátil. También protege la inversión del usuario en el *hardware* existente.

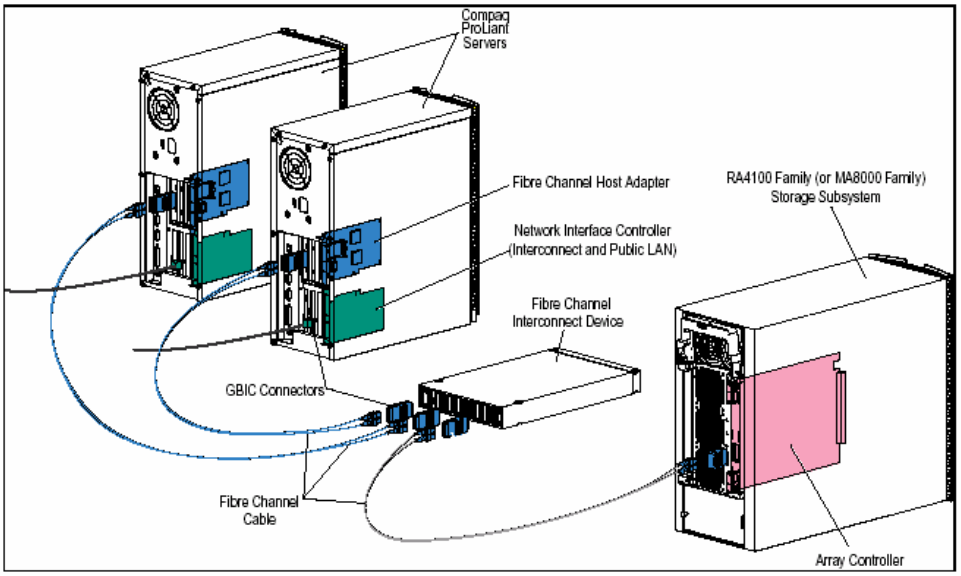
4.3.1. Componentes *cluster* Novell® NetWare®

Un *cluster* NetWare® está compuesto de un número de diferentes productos de *hardware* y *software*. Cada producto juega un rol dentro de la solución completa en el ambiente de computación.

Un *cluster* típico NetWare® consiste en los siguientes componentes:

- Dos o más servidores Intel.
- Componentes del almacenamiento compartido:
    - Uno o más sistemas de almacenamiento.
    - Adaptadores de fibra óptica para los servidores.
    - Dispositivos de interconexión (hubs, *switches*).
    - Conectores y cables de fibra óptica.
- Controladoras interfaz de red para los clientes LAN y la interconexión del *cluster*.

Figura 54: componentes principales de un *cluster* NetWare®



4.3.2. Servicio *cluster* NetWare®

El servicio de *cluster* NetWare® (NWCS por sus siglas en inglés) es el *software* implementado en los servidores para asegurar la alta disponibilidad y administración de recursos críticos de la red, incluyendo datos (volúmenes), aplicaciones, licencias de servidores y servicios.

NWCS soporta *failover*, *failback*, y migración (balanceo de carga manual) de recursos individuales del *cluster*. Provee importantes características para ayudar a asegurar y administrar la disponibilidad de los recursos de red. Estas características incluyen:

- Escalabilidad; *cluster* grandes de hasta 32 nodos son soportados por Novell®.
- Habilidad de agregar nodos al *cluster;* puede expandirse un *cluster* agregando servidores adicionales mientras el *cluster* esta en línea.
- Habilidad para agregar arreglos de almacenamiento compartido al *cluster*.
- Actualización del *software* del *cluster* en línea; puede actualizarse el *software* del *cluster* mientras el *cluster* esta en línea.

4.3.3. Novell® ConsoleOne®

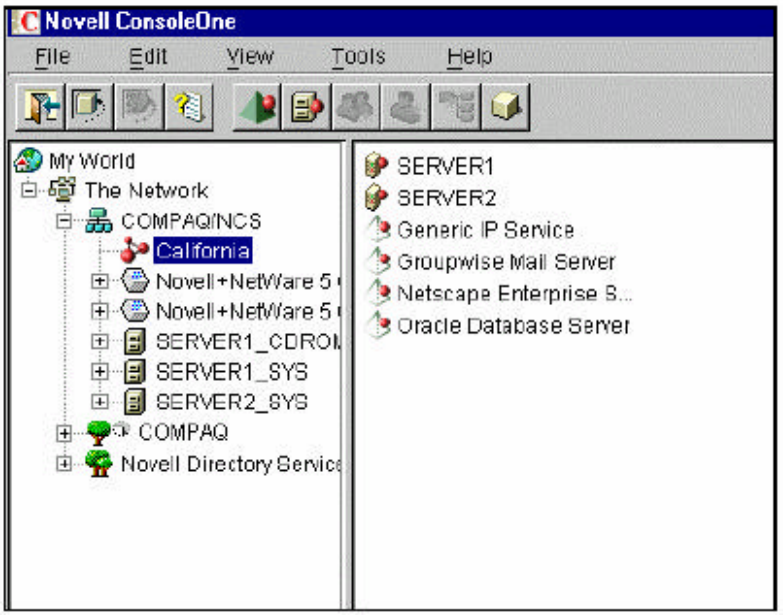
ConsoleOne® es una interfaz gráfica basada en Java que provee un solo punto de administración para la configuración, monitoreo, y manejo del *cluster* NetWare®. ConsoleOne® permite la administración remota del *cluster* desde cualquier cliente o clientes que tienen acceso a el servicio de directorio NetWare®. Las tareas que pueden efectuarse a través de ConsoleOne® incluyen:

- Reservar nodos futuros para el *cluster*
- Volúmenes disponibles para el *cluster*
- Creación plantillas de recursos del *cluster*
- Crear recursos para el *cluster*
- Configurar la carga y descarga de *scripts*
- Configurar los modos de *failover* y *failback*
- Agregar nodos a un recurso
- Ver o editar la membresía al quórum y las propiedades de timeout
- Ver o editar las propiedades de los protocolos del *cluster*

- Ver o editar las propiedades de los puertos del *cluster*
- Ver o editar las propiedades de los nodos del *cluster*
- Migrar recursos de un nodo a otro
- Identificar el estado del *cluster* y sus recursos

ConsoleOne® provee dos vistas desde las cuales se puede manejar el *cluster*: vista del estado del *cluster* y vista de la consola. En la vista de la consola muestra los servidores del *cluster*, recursos, y plantillas. Desde esta pantalla, puede ver y cambiar las propiedades del *cluster* y los recursos y crear volúmenes para el *cluster*, como en la figura a continuación.

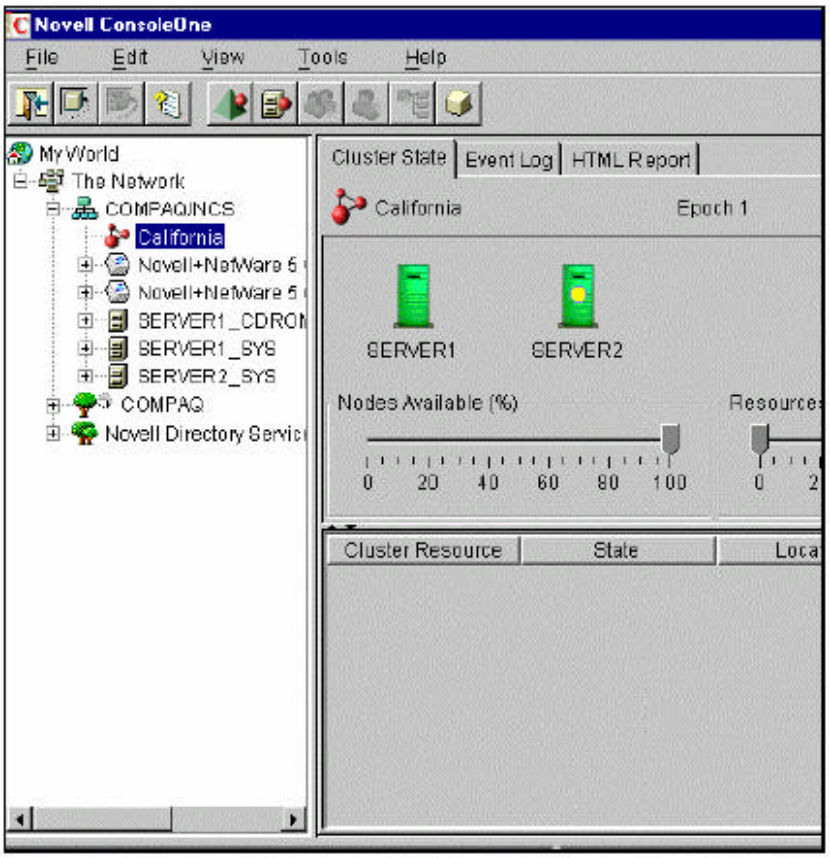
Figura 55: ConsoleOne® vista de consola



En la pantalla vista de estado del *cluster* se despliega información acerca del estado de los servidores y recursos del *cluster*. Este muestra que volúmenes

y servicios esta ejecutándose en cada servidor en el *cluster*, como en la ilustración a continuación.

Figura 56: ConsoleOne® vista estado del *cluster*



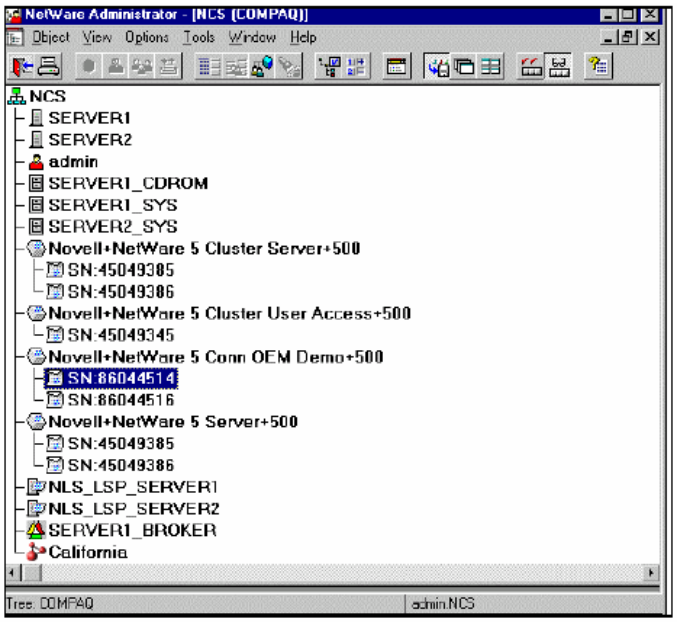
4.3.4. Servicio de directorio NetWare®

El servicio de directorio NetWare® (NDS pos sus siglas en inglés) es la herramienta de Novell para manejar el sistema en la red NetWare®. El servicio

de *cluster* NetWare® esta integrado al NDS, permitiendo que el *cluster* NetWare® pueda tomar ventaja completa de la capacidad de administración NDS.

El NDS consiste en una base de datos que contiene información acerca de los usuarios y recursos en el ambiente de *cluster* y los servicios para administrar la información. Usando NDS, puede construirse un modelo jerárquico para los nodos del *cluster*, los clientes, y la organización en una base de datos. Esta base de datos provee una vista de la red; esta base de datos puede usarse para encontrar información acerca de la red y para acceder a servicios para manejar la red. La ilustración a continuación muestra un ejemplo de una estructura de árbol NDS. Esta estructura consiste en un objeto raíz, objetos contenedores, y objetos que representan a los objetos físicos actuales en la red.

Figura 57: servicio de directorio NetWare®



4.4. *Cluster* con Linux

El sistema operativo Linux y el *software* de fuente abierta están emergiendo como una alternativa para las organizaciones buscando un mejor control de sus ambientes, mejora en el rendimiento, y reducción de costos. Linux esta a demostrado ser una plataforma confiable en muchos mercados a través de muchas aplicaciones.

Linux esta experimentando un crecimiento rápido en la industria y *cluster* de alta disponibilidad Linux están empezando a ser demandados. En respuesta a esta demanda hay varias soluciones de *cluster* en el mercado para Linux. Una de ellas caracterizada por su existo operacional es LifeKeeper de SteelEye Technology. Se desarrolla originalmente por AT&T y NCR Corporation a través de un rango de servidores y sistemas operativos, LifeKeeper esta ahora disponible a través de SteelEye Technology. A cumplido con el período de prueba y demostrado ser una solución robusta en múltiples sitios e industrias por cerca de 10 años. En el año 2000, SteelEye Technology adquirió la tecnología LifeKeeper de NCR y extendió LifeKeeper a los mayores proveedores de Linux, Red Hat, Caldera, y SuSE.

SteelEye está extendiendo el producto para cumplir las demandas de los usuarios para aplicaciones de misión crítica en ambiente Linux. LifeKeeper para Linux versión 3.01 esta certificada para operar con:

- Caldera eServer 2.3
- Red Hat Linux 6.2 y 7.0
- SuSE Linux 7.1

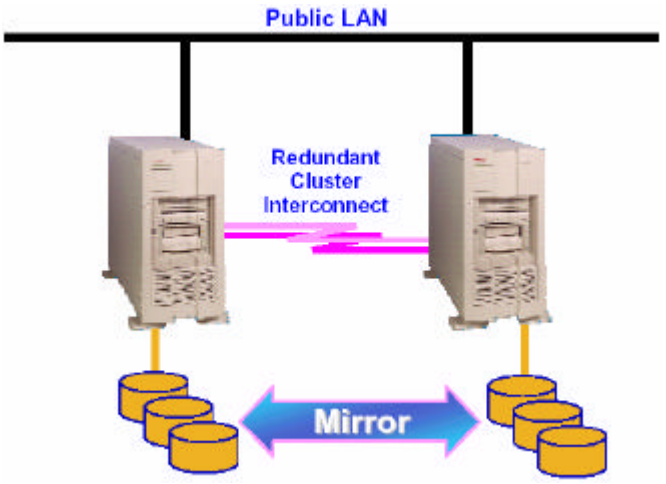
4.4.1. Lifekeeper para *cluster* Linux

LifeKeeper para *Cluster* Linux ha sido ampliamente probado en los principales fabricantes de Linux entre ellos Red Hat, SuSE, y Caldera eServer. Dentro de las configuraciones utilizadas se encuentran espejos replicados, SCSI compartido, y fibra compartida.

4.4.1.1. *Cluster* de espejo replicado

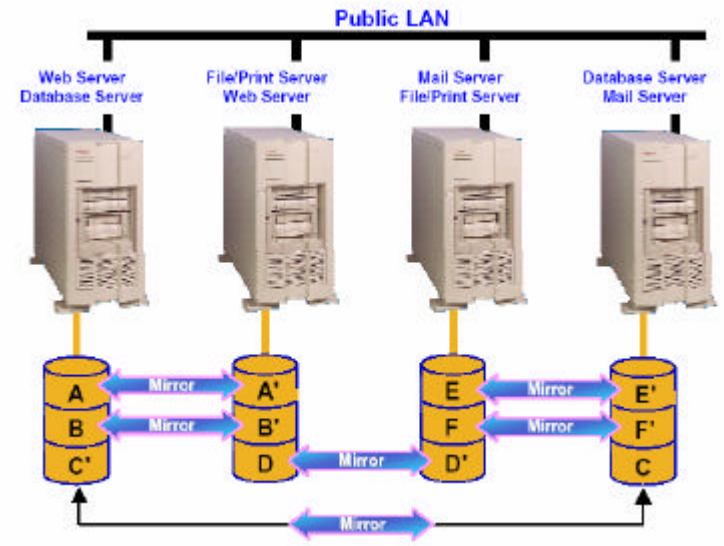
Un *cluster* de espejos replicados replica o crea una imagen de los datos en los discos internos de un nodo a otro a nivel de partición de disco. Puede ser implementado en cualquier tamaño de *cluster*. En el caso de corrupción de datos en cualquier nodo, los datos son automáticamente resincronizados completamente independientemente de la aplicación. Un *cluster* de espejos replicados puede también ser combinado con un *cluster* de fibra compartida de cuatro nodos. La imagen replicada y la fibra deben ser implementadas como un único par de nodos. El *cluster* de espejos replicados es una solución de alta disponibilidad de bajo costo puesto que no requiere componentes de almacenamiento externo; la figura a continuación muestra un *cluster* de espejos replicados de dos nodos que replican los datos de las particiones de un nodo a otro.

Figura 58. *Cluster* imágenes replicadas con dos nodos



La figura a continuación muestra un ejemplo de un *cluster* de espejos replicados que copian las particiones de disco a través a la configuración de 4 nodos.

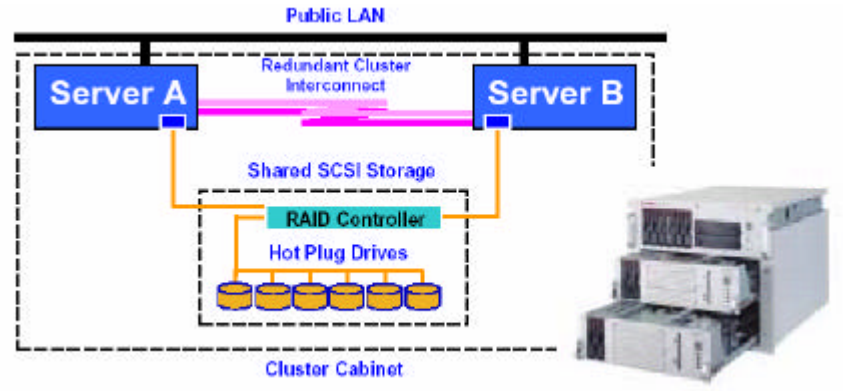
Figura 59. *Cluster* de espejos replicados con cuatro nodos



4.4.1.2. *Cluster* SCSI compartido

En esta configuración dos nodos comparten almacenamiento externo interconectado por medio de tecnología SCSI. En este caso el *failover* de un nodo a otro es mecánico y esto lo cual lo convierte en una opción un poco más lenta que la fibra óptica.

Figura 60: *Cluster* SCSI compartido



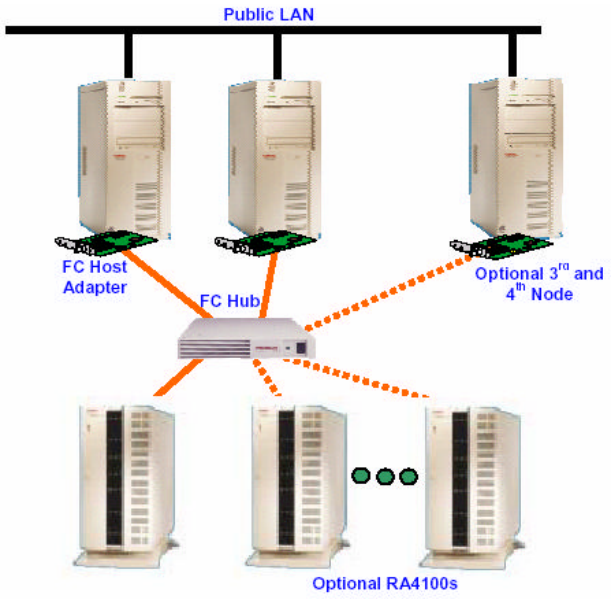
4.4.1.3. *Cluster* fibra compartida

En esta configuración los nodos del *cluster* comparten un almacenamiento externo interconectados por medio de fibra óptica.

4.4.1.3.1. Configuración estándar

En una configuración estándar al menos dos nodos deben ser conectados a un *hub* o *switch* de fibra. Nodos adicionales pueden ser agregados hasta un máximo de 4.

Figura 61. *Cluster* de fibra compartida estándar

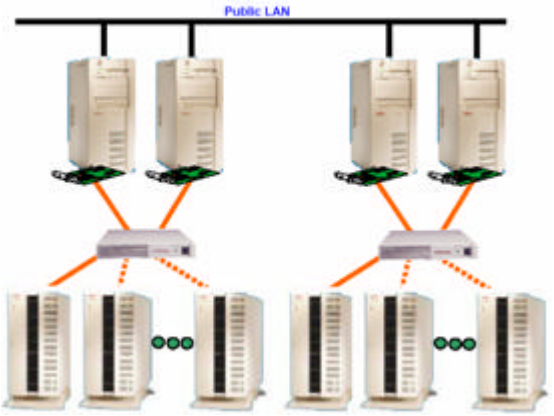


4.4.1.3.2. Configuración de fibra duplicada

En una configuración duplicada, cualquier conjunto de nodos debe ser conectado al almacenamiento de fibra; por ejemplo, los primeros dos nodos en un *cluster* de 4 nodos puede ser conectado a el almacenamiento de fibra. Por ejemplo, los primeros dos nodos en un *cluster* de 4 nodos pueden ser conectados a un almacenamiento de fibra, y este puede ser también conectado a un segundo par de nodos. La figura a continuación muestra tal configuración.

Esta configuración provee capacidad de almacenamiento adicional pero el *failover* debe permanecer en cada grupo de nodos.

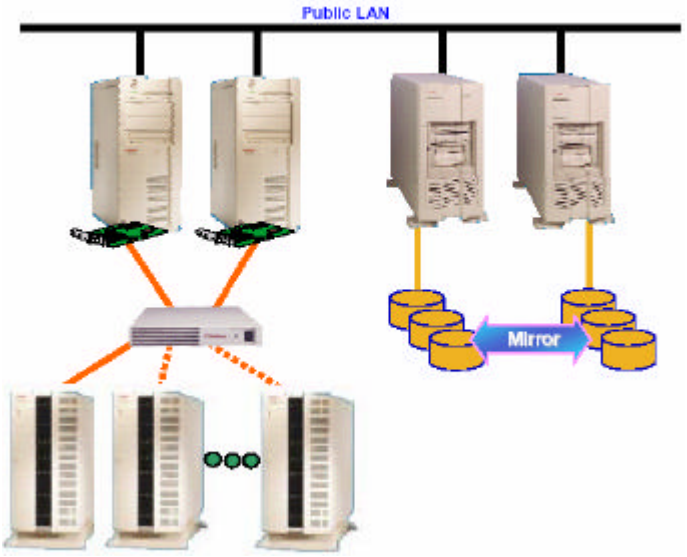
Figura 62: *Cluster* de fibra compartida duplicada



4.4.1.3.3. Configuración de fibra mixta

En una configuración de cuatro nodos, los *cluster* de fibra pueden ser combinados con un *cluster* de replicación de espejos como se muestra en la figura a continuación; cualquier par de nodos puede ser conectado al almacenamiento de fibra y los restantes dos nodos son usados para replicación de espejo; el *failover* debe permanecer dentro de cada par de nodos.

Figura 63: *Cluster* de fibra compartida mixta



4.4.1.4. Interconexión del *cluster*

LifeKeeper para Linux soporta dos o más interconexiones Ethernet TCP entre los nodos; 10/100 y gigabit Ethernet son soportados. Interconexiones seriales TTY redundantes como respaldos también son soportados para dos nodos. Al menos dos interconexiones son requeridas (una primaria y al menos una de respaldo) Esta fuertemente recomendado que la interconexión primaria sea configurada en una red privada. La interconexión de respaldo puede ser configurada sobre la red local privada; una interconexión serial TTY también puede ser usada como un respaldo en un *cluster* de dos nodos.

5. BASES DE DATOS EN ALTA DISPONIBILIDAD

5.1. Microsoft® SQL Server 2000 *Cluster*

5.1.1. Introducción

Para diseñar una solución de bases de datos en alta disponibilidad para la infraestructura *web* deben identificarse los requerimientos para la solución de base de base de datos. Una infraestructura *web* puede requerir una solución de base de datos en alta disponibilidad para almacenar: información de catálogos, información de clientes, información de compras y despachos, entre otros.

5.1.2. Tecnología de base de datos

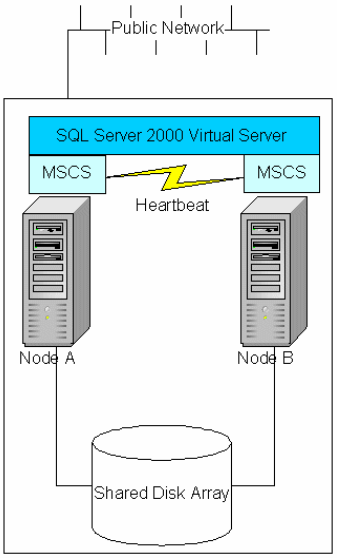
SQL Server 2000 brinda tecnología para alta disponibilidad, la cual permite:

- *Failover* automático de servidor (utilizando el servicio de *cluster* Microsoft® *cluster*)
- Entregar datos de un servidor primario a un servidor en espera (usando entrega de bitácoras).

- Publicar datos a múltiples servidores de reportes (usando replicación transaccional).

El *cluster*, entrega de bitácoras, y la replicación representan cada uno un único rol en la alta disponibilidad. Debe usarse una combinación de estas tecnologías para obtener el nivel de alta disponibilidad esperado. SQL Server 2000 esta construido sobre Windows® *Cluster* o MSCS (Microsoft *Cluster* Service) puesto que es una aplicación diseñada para *cluster*. En la figura a continuación, el servidor virtual de SQL Server 2000 esta colocado sobre la instalación de MSCS existente.

Figura 64: *Cluster* SQL Server 2000



5.1.3. Arquitectura de SQL Server 2000

El *cluster* de SQL Server 2000 requiere SQL Server 2000 Enterprise y ofrece las siguientes características:

- Sistemas de computadoras redundantes
- Detección automática falla de servidor
- *Failover* automático entre los nodos participando en el *cluster*
- Reinicio rápido
- Requiere mínima conciencia de la aplicación cliente

5.1.3.1. Servicio de *cluster*

El *cluster* de SQL server requiere el uso del servicio de Microsoft® *cluster* y uno de los siguientes sistemas operativos:

- Windows® 2000 advanced server para soportar un máximo de dos nodos.
- Windows® 2000 datacenter server para soportar un máximo de cuatro nodos.

El *cluster* de SQL server soporta dos tipos de instancias:

- Una instancia. Una instancia es configurada como un *cluster* activo/pasivo.
- Múltiples instancias. Múltiples instancias remplazan una configuración de *cluster* activo/activo.

SQL server 2000 soporta múltiples instancias por servidor, una instancia por omisión y hasta 15 instancias nombradas. SQL server puede ser instalado como una instancia por omisión o una instancia nombrada en el servicio de *cluster* en el servidor virtual. Es importante asegurarse que todos los sistemas

clientes tengan Microsoft® *data access components* (MDAC) versión 2.6 o superior para permitir el acceso a múltiples instancias de SQL Server.

5.1.3.2. Comunicación

Cuando se diseña una base de datos para alta disponibilidad usando SQL server 2000 debe prepararse cuidadosamente la configuración de la comunicación para la solución. Cuando una computadora con SQL server 2000 es parte de una solución de n capas, deben considerarse los tipos de canales de comunicación para configurar SQL Server. La solución de n capas puede incluir *firewalls* u otros dispositivos de seguridad para proteger el servidor que esta ejecutando SQL server de un ataque externo. También se requieren canales encriptados entre la base de datos y sus clientes para proteger los datos, y su estructura.

Para seleccionar la configuración de la comunicación para una solución de SQL server, debe tomarse en cuenta:

- La posibilidad de ataques contra la computadora con la base de datos y el impacto que tal ataque puede tener en la confiabilidad de los datos. Pueden prevenirse estos ataques tomando en consideración:

    - Usar conexiones TCP/IP entre los clientes y el servidor que esta ejecutando SQL Server.

    - Incluir políticas de encripción para proteger los datos transferidos por la red.

- Minimizar el número de cuentas de usuarios que son usados para proveer acceso a la base de datos.

- Usar conexiones poll para proveer un mejor rendimiento con un mínimo número de conexiones requeridas.

- Las conexiones de red que son usadas para tráfico de consultas son separadas de la red que es usado para tráfico de administración, mantenimiento, y *log shipping*.

Es recomendado que siempre se use un servidor monitor separado en la arquitectura para facilitar la recuperacion después de una falla catastrófica y pérdida de la base de datos fuente. El servidor de monitoreo controla el flujo de las transacciones de datos de un servidor fuente activo a un servidor destino durante la actividad de *log shipping*.

5.1.3.3. Estrategia de red

Uno de los factores para percibir la disponibilidad de la solución *Web* es el tiempo que toma regresar los datos requeridos por una petición del cliente. Si los datos no son retornados rápidamente, el retardo puede causar tiempo fuera del lado del cliente o tiempo fuera en el código de la página ASP o tiempo fuera en el objeto que esta solicitando los datos de la base de datos SQL server.

En la estrategia de red hay que considerar:

- Minimizar la posibilidad de retardos producidos por la congestión de red que causan los dispositivos *firewall*, NAT y servidores de publicación. Puede reducirse el número de saltos entre la fuente de la consulta y el servidor ejecutando SQL Server e idealmente colocar las interfaces en la misma subset.

- Minimizar el retardo producido por la congestión de red que es causada por la administración, *log shipping* y tráfico de replicación. Puede separarse las consultas en una red aislada o subset.

5.1.3.4. Arquitectura de disco

La forma en la cual se utiliza una base de datos en una solución *Web* puede influenciar la arquitectura del *hardware* requerido. Grandes bases de datos que soportan consultas con un alto desempeño en las transacciones y conexiones requiere mucha memoria y un subsistema de disco de alta velocidad tolerante a fallas. Para la alta disponibilidad, los discos deben ser parte de un arreglo externo tolerante a fallas. Debe utilizarse discos SCSI o de fibra canal. Esta última es más recomendable por su alto desempeño y capacidad y que soporta grandes distancias de cableado.

Cuando se selecciones la arquitectura de disco para la solución SQL server, los datos, bitácoras, y tempdb (base de datos temporal) deben ser colocadas en discos separados. De ser posible en canales de entrada/salida diferentes.

5.2. Oracle® real application *cluster*

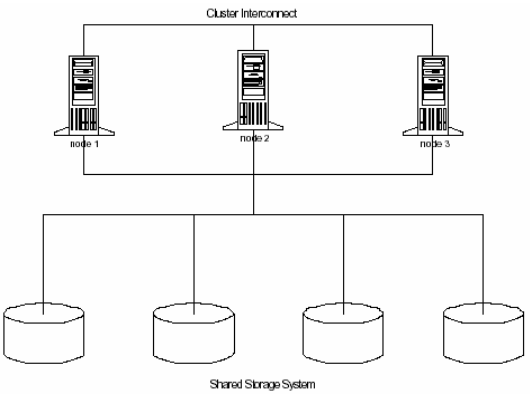
Oracle® real application *Cluster*s (RAC) es una extensión de múltiples nodos a los servidores de base de datos Oracle. Este permite al comercio electrónico construir un servidor de base de datos que es altamente disponible y altamente escalable.

5.2.1. Arquitectura de real application *cluster*

5.2.1.1. Configuraciones típicas soportados por RAC

RAC se ejecuta sobre un *cluster* de *hardware*; un grupo de servidores independientes que cooperan como un solo sistema; el componente primario del *cluster* son los servidores, la interconexión del *cluster*, y el subsistema de almacenamiento compartido. Los servidores comparten el acceso al sistema de almacenamiento y recursos que administran los datos.

Figura 65. Elementos del *cluster*: servidores, interconexión, y disco compartido



5.2.2. Beneficios del *cluster*

El beneficio de construir una base de datos con un *cluster* de pequeños servidores, en lugar de un solo servidor grande son:

- Flexibilidad y eficiencia en costos en la planeación de la capacidad del sistema, de forma que un sistema puede escalar a la capacidad deseada.
- Tolerancia a fallas para fallas parciales con el *cluster*, especialmente fallas de los servidores.

5.2.2.1. Escalabilidad

La base de datos Oracle le da al usuario la flexibilidad de agregar servidores al *cluster* según la demanda por la capacidad se incremente, el crecimiento incremental para ahorrar costos en la inversión de capital y la eliminación de la necesidad de remplazar pequeños nodos individuales con grandes nodos.

5.2.2.2. Alta disponibilidad

Otra ventaja principal de la arquitectura de *cluster* es la tolerancia a fallas que proveen múltiples nodos. Puesto que cada nodo físico funciona independientemente, la falla de uno o más nodos no deben afectar a los otros nodos en el *cluster*. En el caso extremo, un *cluster* puede continuar disponible aun cuando solo un nodo sobreviva, haciendo el sistema basado en *cluster* altamente disponible. Esta arquitectura también permite que un grupo de nodos

se lleven fuera de línea para mantenimiento mientras el resto del *cluster* continua proveyendo el servicio.

5.2.3. Estructura de disponibilidad del *cluster*

Para tomar una completa ventaja de la tolerancia a fallas que permite la arquitectura de *cluster*, la base de datos en Oracle® permite al servidor de base de datos Oracle® funcionar en varios escenarios de falla en el *cluster*. Es más, la base de datos en *cluster* de Oracle® es capaz de recuperar nodos fallidos mientras el servidor de la base de datos esta en línea.

Antes de entrar en detalle de la estructura de disponibilidad para un sistema de *cluster*, deben comprenderse las diferencias entre disponibilidad entre un solo nodo y dentro de un *cluster*. En un sistema de base de datos Oracle en un solo nodo, la disponibilidad se refiere a la habilidad de sobrevivir a fallas en la aplicación y operación en la instancia de la base de datos. En el caso extremo de la falla del nodo, la disponibilidad se refiere a la habilidad de recuperar la base de datos a un estado de transacciones consistente tan rápido como sea posible.

Para un sistema *cluster*, aparte de manejar los escenarios de falla de un solo nodo, este necesita manejar los escenarios asociados con un nodo, un grupo de nodos o la red, mientras provee el rendimiento requerido.

La flexibilidad en un sistema de *cluster* para que cada nodo funcione relativamente independiente viene con algunos problemas únicos con los cuales el sistema de *cluster* tiene que tratar cuando la falla ocurre.

5.2.3.1. Aislamiento de la falla

El sistema de *cluster* mantiene una imagen consistente del sistema todo el tiempo, especialmente durante la falla individual de nodos o en la interconexión del *cluster*. El gran reto para un sistema de *cluster* es que sea capaz de aislar las fallas rápida y confiablemente, y que tome las acciones correspondientes.

Por ejemplo, en el caso de una falla de red, un *cluster* con varios nodos puede terminar como un grupo aislado de nodos conectados. El sistema de *cluster* debe ser capaz de decidir que esta condición es provocada por una falla de red y debe decidir que grupo de nodos conectados (sub-*cluster*) debe continuar operando para el *cluster*, y cual debe ser temporalmente retirado del *cluster*. Esta decisión es crítica para prevenir que el sistema del *cluster* desarrolle el síndrome de “cerebro separado”, en el cual diferentes grupos aislados de nodos conectados (sub-*cluster*) todos afirman representar al *cluster* completo y todos trabajan en el mismo conjunto de datos, sin importar que cada uno de los otros continúe existiendo.

5.2.3.2. Recuperación

Puesto que el nodo o nodos fallidos pueden contener información global de todo el *cluster*, el sistema de *cluster* debe ser capaz de reconstruir la información tan rápido como pueda; este puede mantener un repositorio en espera en caliente para la información global, o puede re-crear la información de los nodos “vivos” y la información almacenada en el sistema de almacenamiento.

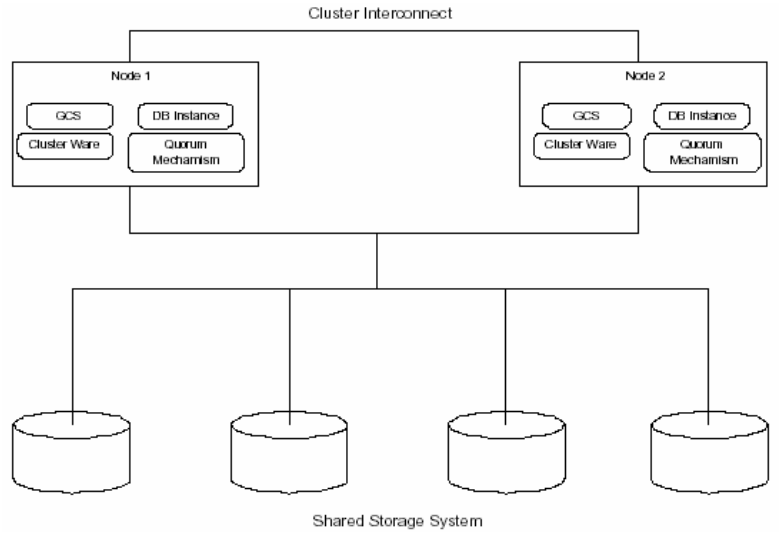
5.2.4. Estructura de disponibilidad de ORAC

5.2.4.1. Arquitectura

En la figura a continuación se describen los componentes de *software* de la base de datos Oracle en *cluster*. Entre los cuales se encuentra:

- *Cluster aware*, es el componente que provee las funciones genéricas de *cluster* a nivel del sistema operativo. El *cluster aware* monitorea el *cluster* y maneja la membresía a los grupos relativas a los eventos, tal como cuando se quiere incluir un nodo dentro o excluir un nodo del *cluster* cuando un nodo inicia o falla.

Figura 66: Componentes de *software* de la base de datos en *cluster* Oracle®



- El mecanismo de *quorum* es usado por la base de datos en *cluster* Oracle® para extender la detección de falla a nivel de la base de datos; por ejemplo, *cluster aware* no conoce si la instancia de la base de datos se cae. *cluster aware* en conjunto con el mecanismo de quórum proveen una detección de falla confiable y aislamiento a nivel de ambos sistema operativo y base de datos.

- GCS (*global caché service*) se asegura que una sola imagen consistente de la base de datos se mantenga; esta mantiene la consistencia de la base de datos a nivel de *cluster*. Los bloques de la base de datos accedidos concurrentemente por los nodos del *cluster* tienen los recursos GCS para asegurar que los mismos bloques de dato no sea actualizados sin coordinación por diferentes nodos; cuando la información se pierde durante la falla de un nodo, este debe ser reconstruido antes que cualquier otra cosa.

5.2.4.2. Recuperación de la base de datos en un ambiente ORAC

La instancia de la base de datos depende de todos estos componentes para implementar la recuperación de instancia para instancias fallidas, en adición a manejar las operaciones normales de la base de datos. Cuando una instancia de base de datos o un nodo del *cluster* falla, la base de datos para *cluster* Oracle® necesita hacer una recuperación para la base de datos de la misma forma que lo hace par aun instancia individual de la base de datos. Puesto que los otros nodos en el *cluster* están aun proveyendo servicio a los clientes, la base de datos de *cluster* debe hacer la recuperación en el menor tiempo posible.

El GCS mantiene el estado de los recursos globales para asegurar la consistencia de la base de datos. Si un nodo falla, este necesita reconstruir la información GCS. La recuperación no puede empezar hasta que el GCS termine de reconstruir la información. Efectivamente, la base de datos completa es "congelada" durante este tiempo. Desde que los recursos del cache para los bloques de la base de datos son distribuidos a través de los nodos del *cluster*, el tiempo necesario para reconstruir la información del GCS es minimizada. Solo los recursos del cache que residen o son dominados por el GCS en el nodo fallido necesitan ser reconstruidos o re-dominados. Esta fase toma solo unos pocos segundos en promedio. Es más, Oracle® 9*i* usa un esquema de recuperación de base de datos de dos fases, donde el primer fase de rastreo de la bitácora redo decide que bloques de datos recuperan y entonces en la segunda fase solo accede los bloques marcados para acelerar la recuperación de una instancia con problema. Oracle9*i*, puede iniciar la primera fase del proceso de recuperación simultáneamente con el proceso de reconstrucción GCS. Después de la primera fase, la base de datos es colocada disponible para los bloques de datos que no fueron impactados por la falla. Oracle® 9*i* también le da la habilidad de especificar la cantidad de tiempo que la recuperación tomará, lo cual elimina problemas potenciales causados por la incertidumbre acerca del tiempo necesario para la recuperación.

5.2.4.3. Recuperación de falla sobre conexiones TCP/IP

Para asegurar una rápida recuperación de las conexiones de los clientes a un nodo/instancia fallida, los clientes de la base de datos pueden conectarse al *cluster* de la base de datos a través de Oracle® Net usando balanceo de carga y recuperación de falla a nivel de aplicación. El balanceo de carga de las conexiones clientes distribuye estas a todos los nodos del *cluster*, aliviando el

impacto de la falla de un nodo. Con la opción Transparent Applicatin *Failover* (TAF), Oracle Net podrá reconectar las conexiones fallidas a otro nodo dentro del *cluster* sin que el cliente se percate de la falla. Hoy, la mayoría de conexiones cliente son hechas usando el protocolo de red TCP/IP.

5.2.4.4. Configuración en línea

Para minimizar el impacto de la configuración del *cluster* de la base de datos Oracle® en alta disponibilidad, los nodos pueden ser agregados o quitados del *cluster* de la base de datos Oracle® sin que el servidor de base de datos deba ser apagado.

5.2.5. Construyendo estructura de disponibilidad de ORAC

Comprender las características de disponibilidad del *cluster* de base de datos Oracle® coloca las bases para entendimiento de las tareas involucradas en la construcción de un servidor de base de datos Oracle® 9*i* en alta disponibilidad. Esta sección discute los aspectos prácticos de la construcción del sistema de base de datos usando la base de datos *cluster* Oracle®.

5.2.5.1. Requerimientos de disponibilidad

Los pasos más críticos en la construcción de un servidor de alta disponibilidad de base de datos son establecer objetivos claros, para crear la expectativa correcta, y recolectar los requerimientos correctos.

La base de datos Oracle® es ideal para proveer disponibilidad cuando un sitio debe ser operacional aun si algunos nodos del *cluster* son llevados fuera de línea por un mantenimiento o falla.

Un *cluster* de la base de datos Oracle® es ideal para proveer alta disponibilidad al sistema cuando un sitio debe permanecer operacional aun cuando algún nodo del *cluster* sea puesto fuera de línea por un mantenimiento o una falla. Almacenes en línea, sitios *web*, bases de datos corporativos, y la mayoría de portales basados en *web* encajan en esta categoría. Si el requerimiento es para un sistema que sobreviva a desastres, un *cluster* de la base de datos Oracle® no es suficiente por si sola. Un sistema de respaldo en espera que no esta ubicado físicamente en el mismo lugar puede proveer una protección extendida a los desastres. Un sistema *cluster* de base de datos Oracle actúa como el servidor principal mientras que otro sistema de base de datos Oracle esta en espera en otra ubicación. Estos sistemas están *loosely coupled* a través de operaciones *standby* de la base de datos pueden sobrevivir a desastres.

5.2.5.2. Consideraciones instalación sistemas *cluster*

Los sistemas de *cluster* requieren componentes de *hardware* y *software* adicionales a los de un solo nodo, los usuarios son aconsejados a tener atención especial durante la instalación. Adicionalmente a los nodos y el *software* del sistema requerido por una sola instancia de base de datos Oracle®, un *cluster* requiere una red de interconexión para enlazar los nodos del *cluster* y el sistema de almacenamiento cuyo acceso es compartido por todos los nodos en el *cluster*.

Para asegurar una instalación sencilla, Oracle® trabaja con los fabricantes para certificar las plataformas del *cluster* como también los sistemas de almacenamiento que son soportadas por el *cluster* de base de datos Oracle®.

5.2.5.3. Conexiones clientes

La conexión de clientes a un *cluster* de base de datos Oracle® es la misma que la conexión a una base de datos individual Oracle®. Ningún cambio se necesita para mover los clientes de una base de datos individual a un *cluster*.

6. HERRAMIENTAS DE ADMINISTRACIÓN DE RED

6.1. Características y protocolos administración red

La administración de la red es un especto importante de las redes modernas. Los administradores de red necesitan instrumentos para monitorear la funcionalidad de los dispositivos y conexiones de red. Cada dispositivo administrado en la red tiene varias variables que cuantifican el estado del dispositivo. Leyendo los valores de estas variables, pueden monitorearse los dispositivos administrados, y escribiendo valores en estas variables, pueden controlarse.

6.1.1. ¿Qué es administración de red?

La administración de red es la habilidad de tener un solo punto de control para efectuar las tareas de administración de la red; la administración de red provee un conjunto de herramientas para que el administrador de red monitoree y administre múltiples sistemas de red y aplicaciones desde una sola representación gráfica de la red.

Hay una necesidad de administración de la red por las siguientes razones:

- Avances en la tecnología de red
- Complejidad y tamaño de la red
- Uso de componentes de múltiples proveedores

- Uso de productos de *hardware* y *software* de múltiples proveedores

6.1.2. Objetivos de la administración de red

Hay dos objetivos principales de la administración de red:

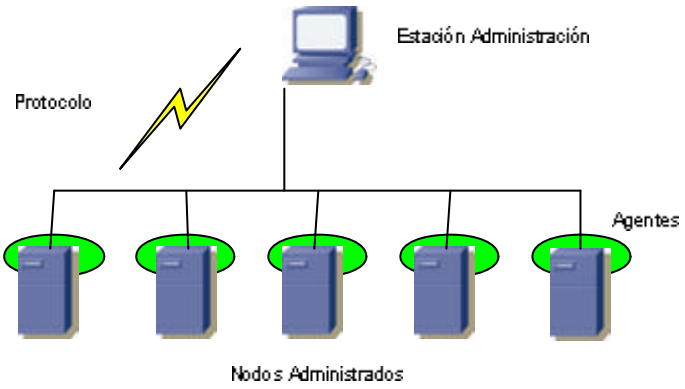
1. Proveer servicios de administración de red a un costo reducido
   - Fácil de usar
   - Mejorar la seguridad
   - Mejorar la contabilización de la utilización de los recursos
   - Control y reporte centralizado
   - Reducción del tiempo en las tareas para los administradores de la red

2. Proveer un rendimiento y servicio de red confiable y consistente
   - Downtime reducido
   - Servicio sin interrupciones
   - Rápida detección, aislamiento y corrección de problemas
   - Anticipación a problemas, carga y demanda
   - Habilidad registrar la información para un diagnosticar a futuro
   - Análisis histórico de tendencias
   - Habilidad de reaccionar cuando un umbral se ha alcanzado

6.1.3. Modelo de administración de red

El modelo de administración de red consiste de una o mas estación de administración de red, nodos administrados cada uno con agentes de administración, y un protocolo de administración de red.

Figura 67: Modelo de administración de red



La estación de administración de red es un sistema ejecuta la aplicación de administración de red y soporta el protocolo de administración de red. La estación de administración de red, ejecuta operaciones de administración de red, la cual monitorea y controla los nodos administrados a través de agentes de administración. Esta estación de administración soporta la carga de la administración de la red, minimizando el impacto de los nodos administrados.

Un nodo administrado es un dispositivo, u objeto, como un computador, un ruteador, un puente, o una impresora de red; la variedad de nodos

administrables es extensa; el factor común de los nodos administrables es que ellos tienen un conjunto de capacidades de red.

El agente de administración de red es la implementación del protocolo de administración de red el cual intercambia información con la estación de administración de red; Cada nodo administrado es visto con varias variables; el nodo administrable mantiene registro de los valores actuales de las variables administrables; el nodo administrable es monitoreado mediante la lectura de estas variables; el nodo administrable es administrado mediante el cambio de estas variables.

El protocolo de administración de red es usado por la estación de administración y los agentes de administración para intercambiar información de administración. El protocolo de administración de la red permite el monitoreo y administración de los nodos administrables.

6.1.4. Protocolo de administración de red - SNMP

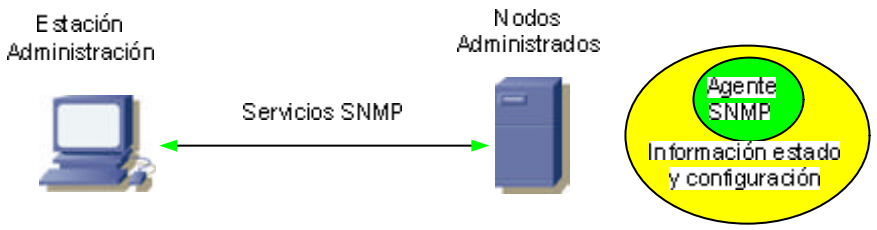
El protocolo *simple network management protocol* (SNMP) es usado para intercambiar información de administración y para el monitoreo y administración de los nodos administrados.

SNMP es basado en el modelo de administración por agentes. En este modelo, una estación de administración (la cual ejecuta el *software* de administración) puede consultar y modificar la información de estado y configuración en cada dispositivo administrado. Esto lo hace haciendo solicitudes al agente corriendo en el dispositivo administrado. El agente tiene

todas las rutinas específicas del dispositivo para consultar y modificar los varios estados y configuración de los componentes.

La información de estado y configuración de los componentes es también llamada *management information base* (MIB). La MIB es definida como un conjunto de variables las cuales son ejecutadas como parte de la actividad involucrada en una variable MIB particular.

Figura 68: Protocolo SNMP



SNMP es un estándar que se actualizado durante el tiempo las últimas versiones son: SNMP2c, definido en el RFC 1901, y la más reciente SNMP3, definido en el RFC 2571 – 2575; donde se han hecho muchas mejores sobre todo en la parte de seguridad.

6.1.5. Base de administración de información (MIB)

La base de administración de información (MIB) define la estructura y tipo de variables de administración que existen en cada nodo administrado SNMP. La organización estándar tiene organizadas las MIB SNMP en una jerarquía con muchos sub-niveles (tal como la estructura de árbol de un sistema de archivo UNIX o MS-DOS); esta organización permite a un objeto particular MIB para ser definidos únicamente de acuerdo a donde es encontrado en la jerarquía MIB.

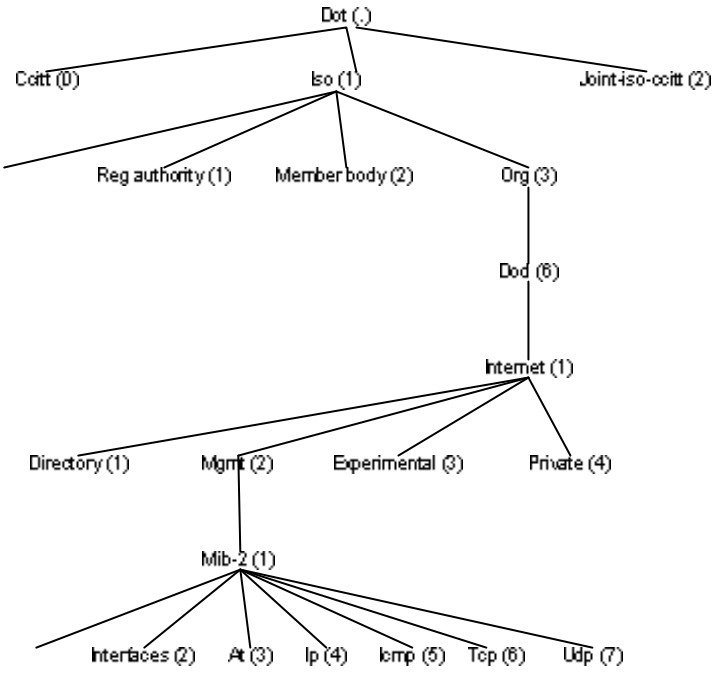
La MIB es

- Definida por estándar
- Organizada lógicamente por la información recolectada de los sistemas administrados
- Jerárquicamente definida con ramas accesibles por nombre o número

6.1.5.1. Nombre de los objetivos MIB

Un nombre de objeto MIB completamente calificado, o ID objeto, es expresado en lo que es llamado notación de punto, con un punto (.) separando cada nivel en una ruta desde el tope de la jerarquía MIB hacia abajo al actual objeto MIB. Los objetos MIB son conocidos como hojas del árbol MIB, mientras todos los nombres en los niveles intermedios son ramas. Cada rama y hoja tienen ambos un nombre numérico y un nombre ASCII, con nombres numéricos frecuentemente usados por brevedad. El siguiente ejemplo muestra las dos diferentes formas de nombrar el mismo objeto MIB: .1.3.6.1.2.1.1.1 e .iso.org.dod.internet.mgmt.mib-2.system.sysDescr

Figura 69: Jerarquía MIB



Todos los nombres de las ramas abajo del nivel de una compañía (u otra empresa) son asignados por una organización estándar para preservar la unicidad. Además del estándar de Internet objetos MIB-II, muchos fabricantes de *hardware*, tal como Hewllett-Packard, Cisco Systems, Wellfleet, y Novell tienen desarrolladas extensiones para sus propios productos.

6.2. Áreas funcionales de administración de red

El modelo de administración de red de la *international organization for standardization* (ISO) define cinco áreas funcionales de la administración de red: administración de fallas, administración de configuración, administración de la utilización de los recursos, administración de rendimiento y administración de seguridad.

6.2.1. Modelo funcional

El modelo de administración de red define cinco áreas funcionales de administración de red:

- **Administración de fallas**, esta función incluye herramientas de monitoreo del estado de la red, alarmas, alertas, y predicción. Esta función es responsable de la detección, aislamiento, y control situaciones anormales en la red.

- **Administración de configuración y cambios**, esta función tiene la capacidad de llevar la pista de la configuración de la red y los dispositivos. Esta función también incluye un control central de la configuración.

- **Administración de utilización de recursos**, esta función obtiene información estadísticas de la red usada por los usuarios. Además, incluye la recolección y procesamiento de datos relacionados con la utilización de recursos de la red.

- **Administración de rendimiento**, esta función asiste a los administradores y operadores en la optimización del rendimiento de la red a través de la recolección y análisis de datos acerca de la red (tiempo de respuesta, carga, y errores e n los subsistemas de red).

- **Administración de seguridad**, esta función protege la red, sus interconexiones, sistemas e información de administración de acceso no autorizado, uso y otros daños.

6.2.2. Administración de fallas

La administración de fallas esta diseñada para manejar condiciones de error que causan que los usuarios pierdan la completa funcionalidad de los recursos de la red. El objetivo de la administración de fallas es detectar, registrar, notificar a los usuarios, y automáticamente corregir el problema de red manteniendo la red corriendo efectivamente. La administración de fallas es efectuada en cinco pasos:

- Determinación de la falla
- Diagnostico
- Sortear y recuperar
- Solucionar
- Registrar y controlar la falla

La determinación de la falla consiste en detectar una falla y completar los pasos necesarios para empezar el diagnóstico de la falla, tal como aislar la falla en un subsistema particular. El diagnóstico de la falla consiste en determinar la causa precisa de la falla y la acción requerida para resolverla. El sortear y recuperarse consiste en los intentos de sortear (evadir) la falla, ya sea en forma parcial o completa. Este provee solo una solución temporal. La resolución de la falla consiste de un esfuerzo por eliminar la falla. Ésta comienza después de que el diagnóstico de la falla se completa e involucra acciones correctivas, tal como el reemplazo de *hardware* o *software* dañado. El registro y control de las fallas consiste en llevar registro de cada falla hasta el final de su solución. La información vital describiendo la falla se graba en la base de datos de fallas.

6.2.3. Administración de configuración

La administración de configuración es una colección de procesos y herramientas que promueven la consistencia de la red, registran los cambios en la red, y proveen documentación y visibilidad de la red actualizada.

La administración de configuración identifica, ejerce control, recolecta datos, y provee datos a los sistemas abiertos con el propósito de preparar, iniciar, empezar y proveer para la operación continua, y terminar los servicios de interconexión; la administración de configuración incluye las siguientes funciones:

- Parámetros que controlan las operaciones de rutinarias de los sistemas
- Asocia nombres con los objetos administrados y con los conjuntos de objetos administrados
- Inicia y cierra los objetos administrados
- Recolecta información en demanda acerca de las condiciones actuales de los sistemas abiertos
- Obtiene anuncios de cambios significativos en las condiciones de los sistemas abiertos
- Cambia la configuración de los sistemas abiertos.

El objetivo de la administración de la configuración es monitorear la configuración de la red y los sistemas de información, permitiendo que las operaciones en la red de varias versiones de elementos de *hardware* y *software* sean registradas y administradas.

Estos objetivos incluyen:

- Bajo costo de soporte de decremento en las tareas de soporte reactivo.

- Bajo costo en la red derivado de la identificación de componentes no usados, utilizando herramientas de registro de dispositivos, circuitos y usuarios.

- Mejorar la disponibilidad de la red derivado del decremento en el costo del soporte reactivo y la mejora en el tiempo de solución de problemas.

Una pobre administración de configuración puede resultar en tareas como:

- Inhabilidad para determinar el impacto en la red de los cambios en los usuarios finales

- Incremento en las tareas de soporte reactivo

- Incremento en el tiempo de solución de problemas

- Alto costo de la red derivado de componentes no utilizados

6.2.4. Administración utilización de recursos

El objetivo de la administración de la utilización de recursos es medir la utilización de los parámetros de la red permitiendo regular el uso de la red individual o en grupo; tal regulación minimiza los problemas de red y maximizan la imparcialidad en el acceso a la red de todos los usuarios.

El primer paso para una propia administración en la utilización de los recursos es medir la utilización de todos los recursos importantes de red; el análisis de los resultados provee información sobre los patrones de utilización actuales, permitiendo que el uso de cuotas sea establecido; en ocasiones, ajustes deben realizarse para obtener el uso óptimo; adicionalmente, las medidas de utilización de los recursos pueden utilizarse como información de tarificación así como también información usada para optimizar el uso de los recursos.

6.2.5. Administración de rendimiento

La planeación de capacidad es el proceso de determinar los recursos de red requeridos para prevenir un impacto en el rendimiento o disponibilidad de las aplicaciones críticas del negocio. La administración de rendimiento es la práctica de administrar el tiempo de respuesta de los servicios de red, consistencia, y calidad de los servicios individuales y en su conjunto.

El objetivo de la administración de rendimiento es medir y hacer disponibles varios aspectos del rendimiento de la red, permitiendo el mantenimiento de un rendimiento aceptable de la red.

La mayoría de organizaciones recolectan información relacionada a la capacidad y trabajan consistentemente para resolver los problemas, cambios de plan, y la implementación de nuevas funcionalidades en capacidad y rendimiento. Las organizaciones no efectúan rutinariamente análisis de tendencia y ¿Qué sucede si? Un análisis ¿Qué sucede si? es el proceso de determinar el impacto de los cambios en la red. Tendencias es el proceso de capturar mediciones base de la capacidad y rendimiento de la red para identificar las tendencias de la red para futuros requerimientos de actualización. La administración de la capacidad y rendimiento también incluyen administración de excepciones (donde las tareas consisten en identificar y resolver antes que los usuarios llamen) y administración de QoS (*quality of services*-calidad de servicio) (donde los administradores de la red planean, administran, e identifican asuntos de rendimiento relacionados con servicios individuales.

La administración de capacidad y rendimiento tiene limitaciones en estas áreas:

- CPU
- *Backplane* o entrada/salida
- Memoria y *buffer*
- Tamaño de las interfases
- Encolamiento y latencia.
- Velocidad y distancia,
- Características de las aplicaciones.

6.2.6. Administración de seguridad

El propósito de la administración de seguridad es soportar las políticas de seguridad de la aplicación, incluyendo: la creación, eliminación, y control de mecanismos y servicios de seguridad; la distribución de información relevante de seguridad; y el reporte de eventos de seguridad relevantes. La administración de seguridad controla el acceso a los recursos de la red, y previene sabotajes a la red (intencionales o no) y acceso no autorizado a información sensitiva.

La administración de seguridad ayuda a los administradores a crear un ambiente de red seguro. Esto incluye partir los recursos de red en áreas autorizadas o no, mapeo de grupos de usuarios en dichas áreas, y monitoreo, aplicación de políticas y registro del acceso de usuarios a recursos en dichas áreas.

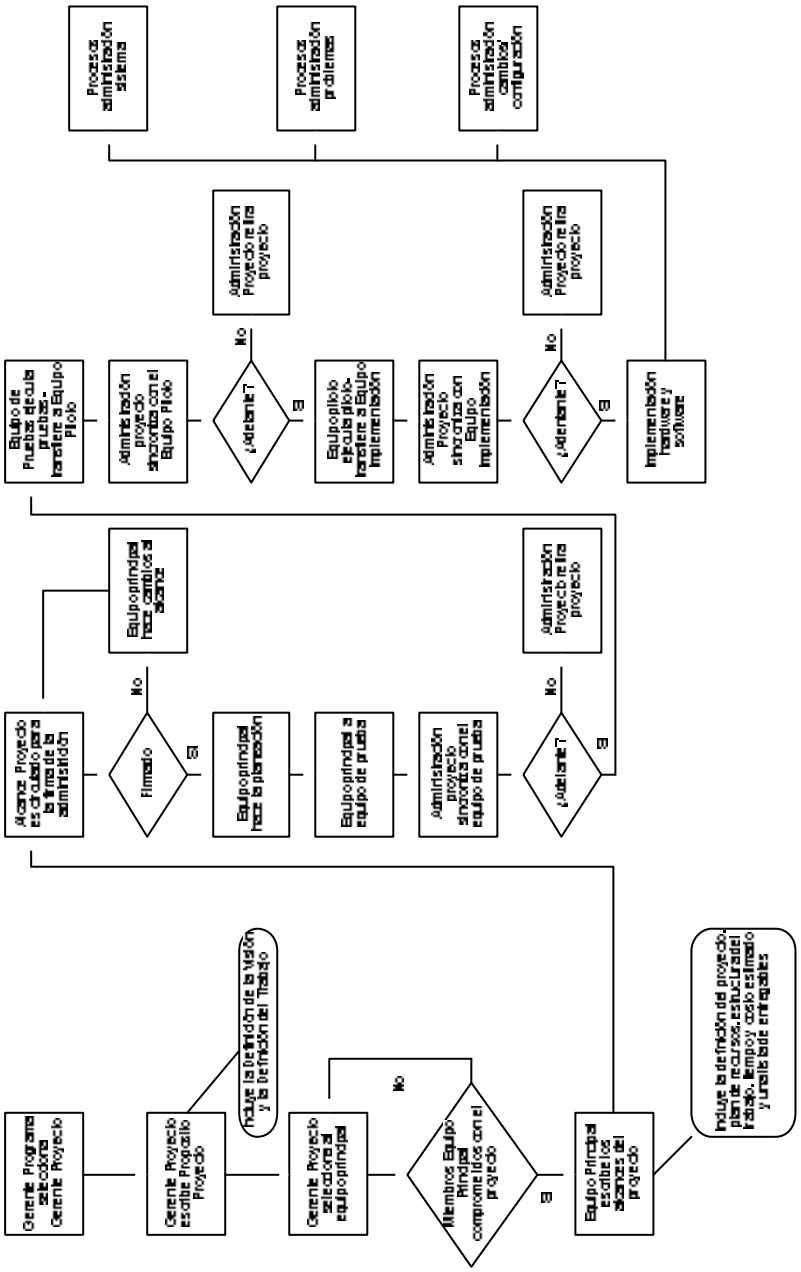
7. PROCEDIMIENTOS PARA ASEGURAR LA ALTA DISPONIBILIDAD

Este capítulo describe lo mejor de los procesos y procedimientos usados en la industria para crear y mantener la alta disponibilidad de los sistemas.

7.1. Introducción

La alta disponibilidad es más que tecnología; la alta disponibilidad se logra solo con la combinación de procesos probados e iterativos, personal entrenado y conocedora, y tecnología de punta planeada. El siguiente diagrama ilustra lo “mejor de lo mejor” en el flujo de procesos utilizados en la industria para crear un sistema de alta disponibilidad

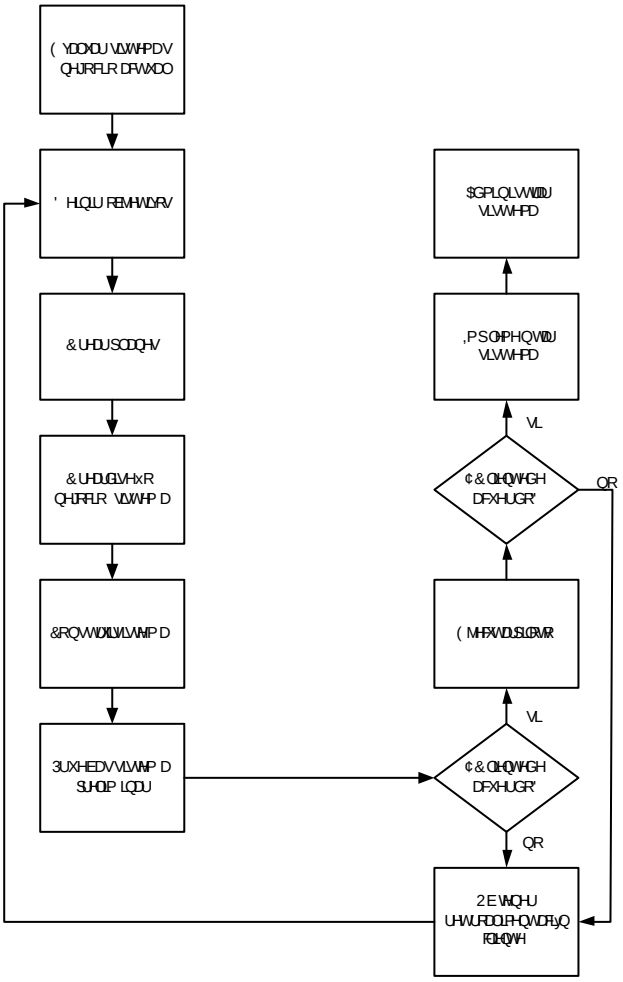
Figura 70. Mejores prácticas en la planeación y administración de alta disponibilidad



7.2. Mejores prácticas comunes

Estos son los bloques fundamentales usados para crear, y continuar la disponibilidad. Los bloques pueden coincidir con el diagrama presentado en la introducción excepto para los entregables, los cuales son una parte esencial de todo proceso, puntos de revisión, los cuales son parte de cada proceso, y lecciones aprendidas, las cuales son parte de todos los bloques de proceso sincronización.

Figura 711. Bloques fundamentales



7.2.1. Evaluar negocio/sistema actual y definir objetivos

Primero se debe definir la administración del proyecto; esta se divide en partes manejables: las personas, procesos y tecnologías necesarias para crear un sistema de alta disponibilidad.

7.2.1.1. Propósito del proyecto

Este define el problema o oportunidad que el proyecto resuelve o capitaliza; también define quién es el patrocinador del proyecto y quién es el propietario del proyecto. Es fácil distraerse o perder el centro. Regresando a la definición del propósito puede ayudar a dirigir los esfuerzos al objetivo.

7.2.1.1.1. Mejores prácticas

- Definición de la visión-una vista sin límites de lo que puede ser la solución, sin ver los límites del proyecto. (El sueño)

- Definición de trabajo-una o dos oraciones claras explicando que es lo que debe hacerse.

- La definición de trabajo y la definición visión deben ser firmadas por todos las partes responsables significando que están de acuerdo y entienden.

- Análisis caso negocio – un simple caso de negocio definiendo un contexto, dirección, estrategia de implementación, estrategia de negocio, asunciones, asuntos claves, análisis de opciones y recomendaciones.

7.2.1.2. Equipo proyecto

Los miembros del equipo deben trabajar junto sinérgica mente. Los roles y responsabilidades son asignados a títulos de trabajo desde un conjunto de recursos de acuerdo con las habilidades disponibles y el foco de trabajo. No tiene sentido asignar roles opuesto a los mismos miembros del equipo.

Cada proyecto tiene un equipo principal; los miembros del equipo principal tienen un compromiso de tiempo completo mientras los expertos son llamados cuando son necesitados.

La documentación describe a cada rol de los miembros del equipo y sus responsabilidades incluyendo una definición clara de la cantidad de tiempo requerida por cada miembro del equipo.

Una definición clara de los roles y responsabilidades asigna responsabilidad. Esto también ayuda a los miembros del equipo a conocer cuando una transición necesita ocurrir.

7.2.1.2.1. Mejores prácticas

- El equipo no debe ser mayor de quince personas, incluyendo al equipo principal y expertos.

- El equipo técnico es *empowered* para tomar decisiones técnicas.

- El equipo de análisis del negocio es *empowered* para tomar decisiones relacionadas con asuntos del negocio.

7.2.1.3. Alcance proyecto

El alcance le da al proyecto un principio y un fin. Una definición clara de lo que debe hacerse, quién lo hará, los procesos de negocio a soportar, y que sistemas existentes se tomará la entrada y salida.

El alcance también identifica un comité responsable de aprobar los cambios y procesar los cambios presentados al comité.

Una definición de alcance bien escrita puede ayudar a definir nuevos requerimientos descubiertos como una parte que “debe tener” o “sería conveniente que tuviera”. La parte que “debe tener” puede ir a través de la administración de cambio para ser incluido en el alcance. Mientras, “sería conveniente que lo tuviera” debe ser priorizado como un requerimiento para una extensión o mantener en una lista para ser considerado después.

7.2.1.3.1. Mejores prácticas

- Identificar y definir el trabajo a ser hecho empezando a un alto nivel y terminando a nivel de tarea; abarca recursos y entregables. Procesos de cambio – define como los cambios en el alcance suceden.

7.2.1.4. Entregables

La entrega de los servicios es parte del propósito de un sistema de alta disponibilidad; los entregables soportan el propósito y alcance del proyecto.

Los entregables son una prueba que el proyecto esta en agenda, que el sistema esta corriendo como se espera, y que el equipo esta trabajando según el objetivo. Estos también son la evidencia documental de quién es responsable, y de que son ellos responsables.

7.2.1.4.1. Mejores prácticas

- Firmar- clientes, miembros del equipo principal, y administración deben firmar los entregables para indicar que están de acuerdo y entienden.
- Equipo principal debe tener una copia en papel de toda la documentación.
- La documentación incluye definición de mojones con sus fechas de finalización y puntos de decisión.

7.2.2. Creación de planes

La planeación es una forma sistemática de asegurar un resultado esperado. La planeación es la única forma de alcanzar un objetivo específico en el tiempo, con todos los componentes necesarios y dentro de presupuesto.

7.2.2.1.1. Mejores prácticas

- Cada proyecto tiene un período de inicio en el cual el equipo principal es identificado, los expertos son identificados y documentación se reúne.
- Entrevistas se mantienen con los expertos para preparar los alcances.
- La fase de planeación es la más larga.
- El inicio incluye sesiones de lluvia de ideas documentadas.

- Identificación de un repositorio central para todos los documentos del equipo.
- Cada servidor tiene un rol específico. Los servidores con múltiples roles fallan más frecuentemente.

7.2.2.2. Registro proyecto

El registro del proyecto lista todas la fases del proyecto, todas las áreas a ser realizadas, asignación de recursos, puntos de revisión, y entregables. También se deben registrar las funciones más importantes y esta es un aviso temprano de ajustes a los procesos y/o recursos necesarios.

7.2.2.2.1. Mejores prácticas

- Expandir las tareas listadas en los trabajos a realizar para incluir tiempo para completar, y los recursos necesarios.
- Cada sub-proyecto tiene su propia agenda ejecutada dentro de la agenda del proyecto principal.
- Reuniones de seguimiento semanales o bi-semanales obligatorias.

7.2.2.3. Plan de comunicación

La comunicación con cada y todos los miembros del equipo es esencial en la construcción y mantenimiento de un sistema de alto disponibilidad. El plan incluye frecuentes reuniones del equipo, donde la documentación debe mantenerse, cuanta información debe ser comunicada y quién es el responsable por la comunicación.

La comunicación rápida y fácil accesible permite informar a quién la necesita; los miembros del equipo solo puede tomar acción cuando ellos conocen que hay un asunto o problema.

7.2.2.3.1. Mejores prácticas

- Colocar toda la documentación del proyecto en la intranet.
- *Net meeting*.
- Semáforo – listar todos los proyectos precedidos por un rojo (fuera de programación), amarillo (agenda pausada), verde (en programa).
- El plan incluye mediadas que deben tomarse en caso el proyecto se retrase.

7.2.2.4. Plan de recursos

El plan de recursos quién efectuara las tareas, los materiales necesarios para efectuar las tareas y donde las tareas serán efectuadas.

Se debe identificar los roles y responsabilidades entre aquellos y las personas con las habilidades deseadas, que aseguran el proceso será llevado cabo como se espera; también se debe identificar qué necesidades deben ser cubiertas y quién debe hacerlo pues ésta es la base del proyecto. Sujetar los recursos hace que la transición de una fase a otra sea más sencilla.

7.2.2.4.1. Mejores prácticas

- El plan incluye un compromiso directo escrito para los miembros del equipo, el cual es firmado por los gerentes apropiados.
- Un compromiso escrito por los recursos asegurado. Incluyendo recursos humanos, y recursos materiales como: espacio, capacidad de máquina, requerimientos de poder y *software*.

7.2.2.5. Plan de escalación de problemas

El plan de escalación de problemas indica a las personas a las cuales hay que llamar, empezando con los niveles bajos continuando con los niveles altos. Este también hace coincidir los tipos de problemas o situaciones que aplican a cada nivel de contacto.

7.2.2.5.1. Mejores prácticas

- Siempre tener una jerarquía de escalación.
- Una persona es designada como un solo punto de contacto para cada área técnica y negocios.
- El plan de escalación y la persona de contacto es publicado en el *Web*.
- Los problemas son marcados con tiempos, y documentados con una descripción detallada.

7.2.2.6. Plan de entrenamiento

El plan de entrenamiento identifica cuando y como un conjunto de habilidades requeridas deben ser adquiridas. Estas también son creadas cuando se introducen nuevos procesos y tecnologías.

El personal entrenado trabajando en un proyecto comete menos errores, lo cual resulta en períodos de prueba más cortos, y una mayor probabilidad que el proyecto se entregue a tiempo y dentro del presupuesto. El entrenamiento ayuda también al equipo a sentirse más cómodos con el cambio.

7.2.2.6.1. Mejores prácticas

- Identificar las necesidades de entrenamiento para los miembros del equipo e incluir tiempo para ello dentro del programa del proyecto.
- Crear material de entrenamiento para el personal que hará las pruebas y los usuarios considerados esenciales.
- El material de entrenamiento es colocado en el *Web*.

7.2.2.7. Plan de pruebas

El plan de pruebas incluye para ejecutar aplicaciones, *hardware* y periféricos combinados como un sistema para asegurar el rendimiento esperado; este también tiene escenarios de prueba para verificar todos los requerimientos para el sistema.

Las pruebas son la única forma de asegurar que todo el conjunto de requerimientos son funcionales. Esta la forma de asegurar que las aplicaciones afectarán positivamente a la empresa.

7.2.2.7.1. Mejores prácticas

- Plan de pruebas incluye ejecutar *software* para asegurarse que trabaja, y los escenarios que cubren todos los requerimientos.
- Las bitácoras se mantienen y son revisadas por el equipo de prueba.
    - Cada individuo técnico tiene su propio plan de prueba.
    - El equipo valida y comparte el plan individual con sus colegas.
    - El equipo une los planes individuales en un plan.
- El plan de prueba incluye escenarios para funcionalidad pasada para asegurar que nada se pierda con la actualización o instalación de la nueva aplicación.
- El plan también incluye instrucciones para respaldar la actualización o instalación en caso el nuevo *software* tenga un impacto negativo en el sistema existente.

7.2.2.8. Plan de piloto

El plan piloto refleja el ambiente de producción y se mantiene por un período de tiempo en que el sistema esta corriendo sin interrupción. Este período de tiempo es iniciado a cero cada vez que el sistema necesita ser interrumpido; el período de tiempo necesita ser lo suficientemente largo para producir datos y reportes, 30 días o más.

El plan piloto también incluye un proceso de selección de participantes. Inicialmente los participantes son representativos de las unidades de negocio estratégicas y expertos selectos dentro de esas unidades.

El plan piloto prueba que el sistema de alta disponibilidad y todos sus periféricos, sistemas de entrada, sistemas de salida y sistemas de administración, funcionan como se necesita.

7.2.2.8.1. Mejores prácticas

- Un piloto corre por 30 días o más sin interrupción. Si una situación causa que el piloto se interrumpa, todo lo hecho debe ser documentado y el reloj se re-inicia.
- Los grupos de foco de usuarios son consultados, antes, durante y después del piloto para obtener ideas, asuntos a resolver, mejores prácticas, y lecciones aprendidas.
- Limitar el número de participantes en el piloto a aproximadamente 5 a 10 por unidad de negocio. Agregar 5 a 10 nuevos usuarios después de un período de tiempo utilizando a los usuarios previos para soportar a los nuevos usuarios.

7.2.2.9. Plan de capacidad

El plan de capacidad considera el *software* y equipo que permite la comunicación y transferencia de datos. Este incorpora planeación de capacidad, red y rendimiento para el sistema completo de punta a punta; esto asegura la disponibilidad de los datos y comunicaciones.

7.2.2.9.1. Mejores prácticas

- Establecer líneas base:
  - Tomar un lineamento base del sistema – el sistema corriendo sin actividad de transacciones y sin componentes adicionales.
  - Tomar una nueva medida cada vez que se agregue un componente estableciendo un lineamento base con esas adiciones.
  - Tomar medidas variando la carga a un rendimiento operacional.
- Monitorear y registrar el rendimiento del sistema durante la operación de producción o piloto, tomando "fotos" y comparando con el rendimiento operacional conocido.
- El grupo de red trabajara mano a mano con los desarrolladores de la aplicación.
- Obtener una visibilidad total de la red. Para resolver los problemas primero debe identificarse donde se originan.
- Usar sistemas de análisis expertos.
- Invertir tiempo y dinero en dimensionar correctamente los servidores, ancho de banda, ruteadores, etc. El dimensionamiento debe incluir los requerimientos de utilización actuales y la utilización esperada dentro de 18 meses.

7.2.2.10. Plan de recuperación de desastres

Debe planearse qué hacer cuando un desastre ocurra; los negocios que sobreviven tienen un plan para reanudar operaciones. ¿Cuánto tiempo puede el negocio sobrevivir sin servicios críticos?

7.2.2.10.1. Mejores prácticas

- El plan de recuperación de desastres tiene equipo especial para pruebas, y pruebas para efectuar a intervalos regulares pruebas de simulación de desastres.
- El plan incluye asegurarse de que hay suficiente capacidad para restaurar una unidad fallida mientras la operación normal de respaldo continua.
- La administración superior decide que aplicaciones son de misión crítica y que tanto tiempo los servicios pueden estar abajo.

7.2.2.11. Punto de revisión

Los puntos de chequeo son puntos para detenerse que le permiten al equipo mirar alrededor y definir si aún el objetivo y si el proyecto es aun relevante para el negocio.

Estos puntos ayudan a eliminar “proyecto que nunca terminan” donde nada se produce. Puntos de chequeo intermedios dan la oportunidad de ajustar la dirección de los equipos.

7.2.2.11.1. Mejores prácticas

Los puntos de chequeo son reuniones planeadas como parte del programa del proyecto.

7.2.3. Creación del diseño para el negocio/sistema, construir el sistema, y pruebas preliminares

El diseño del sistema es un componente clave en la creación de un sistema de alta disponibilidad; toma los requerimientos del cliente y los organiza en un portafolio de servicios que deben proveerse. Los ingenieros y arquitectos definen como se entregan los sistemas. Todo el esfuerzo invertido en la planeación e implementación es un desperdicio si el sistema no trabaja. Un buen trabajo de ingeniería asegura los resultados.

7.2.3.1.1. Mejores prácticas

- Identificar la capacidad y limitaciones del *hardware* del sistema en el proceso de diseño.
- Crear el portafolio de servicios recabando requerimientos y definiendo los servicios principales a entregarse.
- Construir un borrador para unificar nivel de servicio y un proceso de medición mientras se crea la arquitectura del sistema.
- Crear la arquitectura del los procesos y herramientas de administración del sistema como un paso del proceso de diseño.

7.2.4. Pruebas y piloto

Las pruebas y piloto son el primer vistazo que los clientes tendrán de algo que se está produciendo. Por lo tanto, es importante obtener la aceptación de

ellos. La forma como las fases de prueba y piloto son ejecutadas es un indicador al cliente de lo serio que el equipo del proyecto toma el cumplir con los requerimientos.

El sistema juzga mecanismos puestos en funcionamiento para responder preguntas y tratar con problemas. Una rápida respuesta a un problema y soluciones que son comunicadas al equipo, clientes y los individuos que los reportan son esenciales para una satisfacción continua del cliente. Para asegurarse que todos los componentes del sistema trabajan juntos sin interrupción del ambiente de producción se recurre a pruebas y pilotos.

7.2.4.1.1. Mejores prácticas

- Bitácoras – en cualquier momento que el *hardware* o *software* se toca, la persona que lo toca registra que fue hecho en una bitácora. Las bitácoras son publicadas en su formato original en la intranet como parte de la documentación del proyecto.
- Procesos sumisión y reporte de problemas – crear un fólder público que las personas que prueban puedan utilizar para dejar problemas para su revisión.
- Las pruebas son ejecutadas con cuentas y actividades del mundo real.
- Equipo SWAT – un grupo incluyendo desarrolladores, administradores del sistema, expertos, administradores de red, administradores de clientes, administradores proyecto, que son *empowered* para tomar el control cuando son llamados a remediar una situación.

7.2.5. Lecciones aprendidas

Esta es una acumulación del conocimiento recogido durante el camino, conocimiento que, en el pasado, estaba perdido. Los métodos usados para recabar este conocimiento toman muchas formas, reuniones departamentales, reuniones del proyecto, reuniones con colegas, etc. La clave es mantener esto documentado en una forma reusable. Por ejemplo, formas *post-mortem*.

Las lecciones aprendidas, o un post-mortem, son contribuciones a la base de conocimientos usados para comunicar el conocimiento a la audiencia.

7.2.5.1.1. Mejores prácticas

Consiste en reuniones durante el proyecto y al final para compartir problemas, obstáculos, y sus soluciones; las lecciones aprendidas son parte de la documentación del proyecto y son parte de la guía de implementación.

7.2.6. Estrategia de implementación

Poner el sistema en sitio requiere la cooperación de muchos niveles, administración superior, el equipo de instalación, el equipo de entrenamiento, mandos medios, administración del sistema, administración de la red y los usuarios. La coordinación de todos estos niveles y los mecanismos para implementar un sistema de alta disponibilidad esta incluido en la estrategia.

Un sistema de alta disponibilidad funciona solo cuando todos los componentes y soporte están sincronizados; la estrategia de implementación afina y sincroniza el sistema desarrollado.

7.2.6.1. Mejores prácticas

- Una guía de implementación es creada que incluye las lecciones aprendidas del proyecto, la configuración del equipo y *software*, los problemas encontrados con sus soluciones, información soportada, contactos, e información de instalación.

- Un análisis de impacto y ambiente es efectuado; el estudio incluye respuestas a preguntas como:
  - ¿Existe la infraestructura necesaria para soporta el nuevo sistema?
  - ¿Existen las habilidades necesarias para correr y mantener el sistema?
  - ¿Cuál es el impacto en la operación?
  - Un análisis de costos determina la factibilidad de la implementación a lo largo de la empresa.
- Comunicación en tiempo para introducir los cambios y solicitando retroalimentación de los usuarios.
- Un equipo que se especializa en los escritorios se debe instalar para la implementación.
- Hay un proceso de escalación definido y comunicado para la solución de problemas.
- Un análisis de casos de negocios es efectuado para probar la necesidad de la implementación.

7.2.7. Operación día a día

Diligencia en la administración del sistema es la única forma de mantener un sistema de alta disponibilidad.

7.2.7.1. Administración sistema

La administración del sistema incluye carga *software* a las computadoras personales, autenticación de los usuarios, inventario de equipo, des-instalación de *software*, llevar la cuenta de los volúmenes y revisar el rendimiento; básicamente, un afinamiento al minuto de cada aspecto del sistema.

La administración del sistema tiene buen sentido cuando se consideran los ahorros que se pueden recibir; esto se produce en áreas como el costo total de propiedad, retorno de la inversión, distribución de *software*, *help desk* entre otros.

Una revisión y monitoreo proactivo de la salud del sistema mantiene la disponibilidad del sistema.

7.2.7.1.1. Mejores prácticas

- Un equipo o personas dedicadas a tareas de administración específicas, tal como: autenticación, políticas y rendimiento.
- Un equipo o personas dedicadas a tareas de instalación específicas, tal como: instalación y mantenimiento de *hardware* y *software*.
- Un equipo o personas dedicadas solo a respaldo y recuperación.

- El equipo o persona encargada de respaldo y recuperación debe programar periódicamente pruebas de recuperación.
- *Software* para la automatización de tareas administrativas como carga de *software* y monitoreo de sistemas.
- Equipos SWAT son formados por profesionales, y son llamados para corregir anomalías en los sistemas; los miembros del equipo SWAT, cuando son llamados, abandonan sus deberes regulares hasta que el problema es resuelto; ellos tienen poder absoluto en la corrección de la situación.
- Una jerarquía documentada y fácilmente accesible, está disponible para la solución de problemas.
- Un buen *software* de monitoreo para la red y para los servicios entregados es fundamental.

7.2.7.2. Comunicación

La comunicación es el envío y recepción de datos a través de una red y la transferencia de conocimientos de persona a persona; ambas son fundamentales para mantener los sistemas disponibles. Los sistemas en alta disponibilidad no permiten la interrupción del envío y recepción de datos. Para que esto ocurra, un conocimiento en tiempo real tiene que tomar lugar entre las personas y las máquinas.

7.2.7.2.1. Mejores prácticas

- Publicaciones en el *web* de
    - Documentación

- Programación proyecto
- Estadísticas de rendimiento y estado de los servidores, progreso del proyecto y más
- El *web* es bien conocido dentro de la compañía como un repositorio de información

7.2.7.3. Administración de cambios/configuración

La administración del cambio es un proceso de aprobar, priorizar, llevar registro, e implementar los cambios en los procesos de negocio, aplicaciones, y *hardware* que hacen al sistema estar arriba; la administración de la configuración es el proceso de aprobar, priorizar, llevar registro, e implementar cambios al estándar en *hardware* o *software* del sistema.

La administración del cambio es el reto más grande en la tecnología de la información; los rápidos cambios son necesarios para mantenerse en el negocio, mientras, los cambios a los procesos, personas y tecnologías deben ser orquestados con simetría para presentar una operación sin fallas. Administrar el cambio y las configuraciones es lo que hace una operación sin fallas.

7.2.7.3.1. Mejores prácticas

- Mantener una bitácora de cambios en servidores y *software* – todas las actividades son registradas.
- Un consejo de administración de cambios – todos los cambios necesitan ser revisados y aprobados por el consejo.

- Puntos de bloqueos – estándares dictados para todo el *software* y *hardware*. Requerimientos por cambios a los puntos de bloqueo son presentados, revisados a intervalos pre-determinados, y rechazados o aprobados. Los cambios son agregados y probados antes que sean certificados como parte de un punto de bloqueo.

7.2.7.4. Administración de problemas

La administración de problemas lleva registro de los problemas desde su notificación hasta su resolución. Es otro contribuyente a la base de conocimientos. Los problemas son reportados al escritorio de ayuda, resueltos o dirigidos a un experto para su solución; el problema y su solución son registrados en una base de datos que esta disponible como referencia. Todas las personas, procesos, y la tecnología involucrada en la administración de problemas son la base de referencia de la base de conocimientos.

7.2.7.4.1. Mejores prácticas

- Los reportes de estado del escritorio de ayuda son enviados a los gerentes apropiados en forma diaria y semanal.
- Procedimientos detallados de solución de problemas, escalación y contactos.

CONCLUSIONES

1. La arquitectura de la infraestructura de una solución de comercio electrónico puede dividirse en capas de servicio. Las cuales consisten en las capas de servicios de red, sistema operativo, bases de datos y aplicaciones. Para conseguir un nivel de alta disponibilidad en cada una de ellas, hay que trabajar en la redundancia de componentes en menor o mayor medida, considerando la variable costo/beneficio.

2. En los servicios de red la alta disponibilidad se coloca alrededor de sus componentes, entre los que se encuentran balanceadores de carga geográficos, ruteadores, caché de contenidos, *switch*, *firewalls*, y balanceadores de carga a nivel de servidores. La alta disponibilidad se logra al colocar redundancia en los elementos mencionados.

3. Como parte de los componentes vitales dentro de una solución de comercio electrónico se encuentra los servidores, computadoras especializadas que cumplen una función en la arquitectura de comercio electrónica. El avance tecnológico ha permitido que la arquitectura de los servidores cada día brinde nuevas opciones de alta disponibilidad a un menor costo, haciéndolo accesible a una gran cantidad de usuarios.

4. El *cluster* de servidores ha sido una tecnología de alta disponibilidad que ha existido por varios años en el mercado. Esta ha alcanzado madurez que ha permitido que se coloquen este tipo de soluciones en ambiente de

misión crítica. Existiendo diversas opciones en el mercado de fabricantes como Microsoft®, Novell®, Linux®, y varias de ambientes Unix.

5. Las bases de datos son un componente muy importante dentro de la arquitectura de comercio electrónico y la disponibilidad es crítica para la solución. Dentro de las opciones que existen y se utilizan ampliamente en el mercado está el *cluster* de Microsoft® SQL server, y real application c*luster* de Oracle®.

6. Para garantizar la disponibilidad del sitio de comercio electrónico es necesario un conjunto de herramientas de administración que permitan tomar acciones correctivas y preventivas para mantener la solución trabajando en forma continua.

7. La solución de alta disponibilidad no se puede completar si no se cuenta con un conjunto de políticas y procedimientos que garantice que los planes y diseños propuestos operen de acuerdo con sus especificaciones.

RECOMENCACIONES

1. Evaluar el nivel de disponibilidad que requiere y en base a esto determinar el nivel de redundancia que se colocará en cada uno de los componentes del mismo; al construir un sitio de comercio electrónico.

2. Considerar los sistemas de administración del sitio electrónico como un componente vital de la solución, para garantizar los niveles de servicio que requiere el mismo. Estas herramientas nos ayudan a medir el nivel del servicio que se esta entregando, nos recomiendan mejoras que pueden hacerse a la arquitectura del sitio, para actuar en una forma preactiva y nos alertan a fallas en el sistema, que nos permite reaccionar rápidamente.

3. Definir políticas y procedimientos que permitan garantizar los niveles de servicio comprometidos.

BIBLIOGRAFÍA

1. Ali, Beaulieu, y Chauhan. **Business considerations for uptime.** Draft Version 0.8. Microsoft Corporation. Abril, 2000. Estados Unidos de América.

2. Cisco Systems. **Cisco 7500 series router high-availability initiative beat the downtime.** 2001. Estados Unidos de América.

3. Cisco Systems. **Cisco 7500 series router increased performance with enhanced high-availability and networking features.** 2001. Estados Unidos de América.

4. Cisco Systems. **Hot standby router protocol (HSRP) features and functionality.** Julio 09, 2002. Estados Unidos de América.

5. Cisco Systems. **Cisco content engine data sheet.** 2000. Estados Unidos de América.

6. Cisco Systems. **Cisco content engines for the internet content delivery network.** 2001. Estados Unidos de América.

7. Cisco Systems. **Cisco cache engine 500 series.** 2000. Estados Unidos de América.

8. Cisco Systems. **Catalyst 6000 family – multilayer *switches*.** Septiembre, 2002. Estados Unidos de América.

9. Cisco Systems. **High availability for the catalyst 6500 series.** 2001. Estados Unidos de América.

10. Cisco Systems. **Understanding and configuring spanning-tree protocol (STP) on catalyst.** Diciembre 11, 2002. Estados Unidos de América.

11. Cisco Systems. **Configuration guide for the cisco secure PIX *firewall* version 5.0**. Estados Unidos de América.

12. Cisco Systems. **LocalDirector: high availability through application-aware systems.** 1999. Estados Unidos de América.

13. Cisco Systems. ***Failover* configuration for LocalDirector** 2000. Estados Unidos de América.

14. Cisco Systems. **Designing for cisco internetwork solutions** Volume 3, Version 1.0 Student Guide Cisco Systems, 2002. Estados Unidos de América.

15. Compaq Corporation. **Technology profile: compaq hot plug RAID memory technology.** Octubre, 2002. Estados Unidos de América.

16. Compaq Corporation. **New challenges drive compaq advanced memory protection strategy.** Marzo, 2001. Estados Unidos de América.

17. Compaq Corporation. **Technology profile: compaq hot plug RAID memory technology.** Octubre, 2000. Estados Unidos de América.

18. Compaq Corporation. **Compaq proliant 4-node *cluster*s for microsoft windows 2000 datacenter.** Septiembre, 2000. Estados Unidos de América.

19. Compaq Corporation. **Microsoft *cluster*ing: principles of operation, applications and management.** Mayo, 2000. Estados Unidos de América.

20. Compaq Corporation. **Planning considerations for compaq proliant c*luster*s using microsoft *cluster* server.** Primera Edición. Diciembre, 1997. Estados Unidos de América.

21. Compaq Corporation. **Introduction to compaq netware *Cluster*s.** Abril, 2001. Estados Unidos de América.

22. Compaq Corporation. **Executive brief: lifekeeper for linux *cluster*s on proliant.** Abril, 2001. Estados Unidos de América.

23. Ching, Aaron, y Kirk, Steve. **Building a highly available database c*luster*.** Microsoft Developer Network. Diciembre, 2000. Estados Unidos de América.

24. Hewllet-Packard. **Quickspecs HP proliant ML570 generation 2 (G2).** Diciembre, 2002. Estados Unidos de América.

25. Hewllet-Packard. **Technology brief: Compaq advanced memory protection technologies.** Junio, 2002. Estados Unidos de América.

26. Hewllet-Packard. **Technical Whitepaper: Choosing the Right Disk Technology in a High Availability Environment.** Mayo, 1996. Estados Unidos de América.

27. Hewllet-Packard. **QuickSpecs – HP StorageWorks SAN.** Julio, 2002. Estados Unidos de América.

28. Hewllet-Packard. **Hp storageworks – SAN design reference guide.** 4ta edición. Agosto, 2002. Estados Unidos de América.

29. Hewllet-Packard **The winning combination: compaq proliant servers and microsoft windows 2000.** Febrero, 2001. Estados Unidos de América.

30. Hewllet-Packard **What´s new in microsoft windows 2000 *cluster* server** Enero, 2000. Estados Unidos de América.

31. Hewllet-Packard. **HP openview training course network node manager on windows NT/2000 operating systems.** Version D.01 Mayo, 2001. Estados Unidos de América.

32. Infante, German. **Hp proliant server basic training.** Hewllet-Packard. Octubre, 2001. Estados Unidos de América.

33. Infante, German. **Hp proliant server basic training.** Hewllet-Packard. Octubre, 2002. Estados Unidos de América.

34. Infante, German. **Hp proliant server second level training.** Hewllet-Packard. Agosto, 2002. Estados Unidos de América.

35. Infante, German. **Hp proliant server advanced training.** Hewllet-Packard. Noviembre, 2002. Estados Unidos de América.

36. Johns, Paul y Ching, Aaron. **Bulding a highly available and scalable w*eb* farm.** Microsoft Corporation. Diciembre, 2000. Estados Unidos de América.

37. Microsoft corporation. **Cisco and microsoft e-commerce framework architecture.** 2000. Estados Unidos de América.

38. Microsoft corporation. **Course 2088 – designing a highly available *web* infraestructure.** Septiembre, 2001. Estados Unidos de América.

39. Microsoft corporation. **A+ certification training kit.** 3era edición. 2001. Estados Unidos de América.

40. Microsoft corporation. **Implementing windows 2000 *cluster*ing.** Diciembre, 2000. Estados Unidos de América.

41. Microsoft corporation. **Windows 2000 *cluster*ing technologies: *cluster* service architecture.** 2000. Estados Unidos de América.

42. Microsoft corporation. **Designing a highly available *web* infrastructure.** Course number: 2088A. Noviembre, 2001. Estados Unidos de América.

43. Microsoft corporation. **SQL server 2000 *failover cluster*ing.** 2002. Estados Unidos de América.

44. Microsoft corporation **Best practices for end-to-end high availability** Mayo, 1999. Estados Unidos de América.

45. Oracle corporation. **Building highly available database servers using oracle real application *cluster*s.** Mayo, 2002. Estados Unidos de América.