UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES.

CAMPUS ARAGÓN

"ALTA DISPONIBILIDAD PARA DATOS DE MISIÓN CRÍTICA"

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACIÓN

P R E S E N T A N:

LIEV BASURTO TORRES
MIGUEL ANGEL CHAVEZ SANCHEZ

ASESOR:
M. EN C. MARCELO PEREZ MEDEL

MÉXICO **2005**

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Miguel Angel Chavez Sanchez

FECHA: 18/03/05

FIRMA:

Autorizo a la Dirección General de Bibliotecas de la UNAM a difundir en formato electrónico e impreso el contenido de mi trabajo recepcional.

NOMBRE: Lic. Horacio James

FECHA: 18/03/05

FIRMA:

INDICE

INDICE DE TABLAS Y FIGURAS

TABLAS

FIGURAS

OBJETIVO GENERAL

Describir el funcionamiento de la alta disponibilidad en los sistemas de cómputo.

OBJETIVOS PARTICULARES

- **Exponer las características de los sistemas de cómputo y la evolución de estos de acuerdo a las necesidades de los diferentes usuarios.**
- **Mencionar las características, mediciones y el entorno donde se desarrolla la disponibilidad.**
- **Mostrar la importancia de la alta disponibilidad en la organización de las empresas como una característica fundamental en la productividad.**
- **Señalar las diferentes técnicas utilizadas para la obtención de la alta disponibilidad en los sistemas de cómputo.**
- **Diferenciar y explicar los puntos complementarios que contribuyen a un sistema de alta disponibilidad en una mejor funcionalidad.**
- **Relacionar las características y elementos necesarios para construir e implementar un sistema de computo de alta disponibilidad**
- **Ilustrar el algoritmo para la instalación de un sistema de cómputo de alta disponibilidad.**

1. LA COMPUTACIÓN DE HOY Y SUS NECESIDADES

INTRODUCCIÓN

En esta economía globalizada y competitiva, la transformación de la información de cualquier organización en un activo de información constituye una clave para obtener ventajas sobre la competencia.

Para crear y gestionar este activo de información, las empresas necesitan almacenar, recuperar, acceder, analizar y distribuir su información operativa. Las principales herramientas de información estratégicas giran en torno a las bases de datos, que deben funcionar con soportes intermedios y herramientas para bases de datos distribuidas, acceso a Internet, negocio electrónico, reproducción de la información, asistencia a la toma de decisiones, análisis de la información, almacenamiento y extracción de datos, gestión y administración de sistemas y desarrollo de aplicaciones.

Estas herramientas de gestión de la información deben integrarse con componentes complementarios para la gestión transaccional, el trabajo en red y la comodidad y ventajas de las modernas soluciones para trabajo en grupo. Estos productos deben ir acompañados del servicio, la asistencia y las aplicaciones adecuadas para la solución global que ofrezca a las organizaciones flexibilidad y rapidez. Éstas últimas deben mantenerse por delante de la competencia, a medida que ésta descubre nuevas formas de obtener plusvalía con sus aplicaciones de información estratégica.

Anteriormente, la información fluía a través de canales más o menos previsibles: bases de datos, informes de departamentos y hojas de cálculo. Dicha información podía indexarse, recuperarse o consultarse con facilidad. Actualmente, la información fluye por varias vías distintas, correo electrónico, computadoras portátiles, software para grupos, correo hablado servicios de información en línea lo que dificulta el trabajo de los directores de sistemas de información, pues las computadoras reciben torrentes de información sin estructurar. Los directores de informática de las empresas no podrán evitar integrarse en la informática de Internet y deberán adaptarla para asistir a los directivos de las empresas.

La capacidad para analizar esta compleja información de crecimiento exponencial y para convertirla en un activo de información reviste una importancia estratégica para la mayoría de las organizaciones, ya que dicho activo les otorga una ventaja de competencia, pues les permite proveer productos y servicios personalizados en masa, adaptándose así mejor a las carencias y necesidades de sus clientes, lo que aumenta la satisfacción entre estos y su parte del mercado.

Durante las tres primeras décadas de la informática, el principal desafío era desarrollar el hardware de las computadoras de forma que se redujera el costo de procesamiento y almacenamiento de datos. A lo largo de la década de los ochentas, los avances en microelectrónica han dado como resultado una mayor potencia de cálculo a la vez que una reducción del costo.

Hoy el problema es diferente. El principal desafió es reducir el costo y mejorar la calidad de las soluciones basadas en computadoras, soluciones que se complementan con el software.

La potencia de las grandes computadoras de ayer está hoy disponible en un grupo pequeño de circuitos integrados. Las imponentes capacidades de procesamiento y almacenamiento de hardware moderno representan un gran potencial de cálculo. El software es el mecanismo que nos facilita utilizar y explotar este potencial.

Como se mencionó con anterioridad durante las últimas décadas del siglo veinte, los sistemas basados en computadoras están introduciendo un nuevo orden de cosas. La ingeniería del software y la del hardware entran dentro de la complicada categoría que llamaremos ingeniería de sistemas de cómputo. Cada una de estas disciplinas representa un intento de establecer un orden en el desarrollo de sistemas basados en computadoras. Las técnicas de ingeniería para hardware de computadoras provienen del diseño electrónico y han alcanzado un estado de relativa madurez. Las técnicas de diseño de hardware están bien establecidas, los métodos de fabricación mejoran continuamente y la fiabilidad es más una expectativa real que una modesta esperanza. En los sistemas basados en computadoras el software ha remplazado al hardware como elemento del sistema más difícil de planificar, con menos posibilidades de éxito (en tiempo y dinero) y más peligroso de manejar. Mientras los sistemas basados en computadoras continúan creciendo en número, complejidad y aplicaciones, la demanda de software continúa sin disminuir.

1.1. COMPLEJIDAD DE LOS SISTEMAS.

1.1.1 Definición de sistema

El concepto ha invadido todos los campos de la ciencia y penetrado en el pensamiento y el habla popular. El razonamiento en términos de sistemas desempeña un papel dominante en muy variados campos, desde las empresas industriales y los armamentos hasta temas reservados a la ciencia pura.

Las raíces de este proceso son complejas. Por un lado está el tránsito desde la ingeniería energética hasta la ingeniería de control, que dirige procesos mediante artefactos de baja energía y que ha conducido a las computadoras y la automatización.

La tecnología ha acabado pensando no ya en términos de máquinas independientes sino de sistemas. Un automóvil o un receptor de radio caían dentro de la competencia del ingeniero adiestrado en la respectiva especialidad. Pero cuando se trata de proyectiles o de vehículos espaciales, hay que armarlos usando componentes que proceden de tecnologías heterogéneas: mecánica, electrónica, química, etc.; empiezan a intervenir relaciones entre hombre y máquina, y salen al paso innumerables problemas financieros, económicos, sociales y políticos. O bien el tráfico aéreo, o incluso automotriz, no es sólo cosa del número de vehículos en funcionamiento sino que son sistemas que hay que planear o disponer. Así vienen surgiendo innumerables problemas en la producción y el comercio.

Así pues se hizo necesario, un enfoque de sistemas. Dado un determinado objetivo, encontrar caminos o medios para alcanzarlo requiere que el especialista en sistemas (o en la mayoría de los casos especialistas) considere soluciones posibles y elija las que prometen optimización con máxima eficiencia y mínimo costo en una red de interacciones tremendamente complejas. Esto requiere técnicas complicadas y computadoras para resolver problemas que van mucho más allá de los alcances de un matemático. Tanto el hardware de las computadoras como el software de las ciencias de los sistemas representan una nueva tecnología que ha sido llamada la segunda revolución industrial y sólo lleva unas décadas desenvolviéndose.

Semejante evolución no pasaría de ser otra de las numerosas facetas de cambio en nuestra sociedad tecnológicamente contemporánea, si no fuera por un factor significativo fácil de ser pasado por alto en las técnicas sutiles y forzosamente especializadas de la ciencia de la computación, la ingeniería de sistemas y campos afines. No sólo está la tendencia, en la tecnología, a hacer cosas mayores y mejores, sino que hay cambios en las categorías básicas del pensamiento, del cual las complejidades de la tecnología moderna no pasan de ser una manifestación, acaso ni la más importante. De uno u otro modo estamos forzados

a vérnoslas con complejidades, con totalidades o sistemas, en todos los campos del conocimiento.

La cibernética es una teoría de los sistemas de control basada en la comunicación (transferencia de información) entre sistemas y medio circundante, y dentro del sistema, y en el control (retroalimentación) del funcionamiento del sistema en consideración al medio.

La palabra sistema es posiblemente el término más sobre utilizado y del que más se ha abusado en el léxico técnico. Esta situación no se ha limitado al complejo industrial: los políticos suelen pedir que se aplique el enfoque de sistemas a problemas apremiantes, tales como la contaminación del aire y el agua, la planeación de ciudades, la delincuencia juvenil y el crimen organizado. La palabra no dice poco. Usamos el adjetivo que la describe para comprender el contexto en el que se usa.

La noción de sistema puede ser común, para el estudio de los seres vivos así como para la construcción de máquinas propositivas. Un sistema se compone de elementos y relaciones.

Se debe distinguir entre los mecanismos que requiere el sistema para poder ejecutar el comportamiento global y la meta de ese comportamiento global.

La meta es lo que da un sentido al comportamiento global, mientras que en el sentido sólo hay mecanismos. Los mecanismos son la causa eficiente de los movimientos del sistema, pero carecen de meta.

La construcción de un sistema supone primeramente definir la meta que se quiere realizar antes de encontrar el diseño de los mecanismos que permitan la ejecución de los movimientos. La descripción de los mecanismos difiere de la definición de meta.

Los mecanismos causan los movimientos, los movimientos realizan las tareas.

Nosotros definiremos a un sistema de cómputo como: un conjunto o combinación de elementos organizados que actúan conjuntamente para llevar acabo algún método, procedimiento o control mediante el procesamiento de información.

Los componentes de un sistemas de cómputo se desglosan a continuación.

a) **Software:** Los programas de computadoras, estructuras de datos y documentación asociada que sirve para realizar el método lógico, procedimiento o control requerido.

b) **Hardware:** Los dispositivos electrónicos que proporcionan la capacidad de procesamiento y los dispositivos electromecánicos que proporcionan las

funciones de almacenamiento, así como los elementos de comunicación que proporciona las funciones con el mundo exterior.

c) **Seres humanos:** Los individuos que son usuarios y operadores del software y el hardware.

d) **Bases de datos:** Una colección grande y organizada de información a la que se accede mediante el software y que es una parte integral del funcionamiento del sistema.

e) **Documentación:** Los manuales y otra información descriptiva que explica el uso y/o la operación del sistema.

f) **Procedimiento:** Los pasos que definen el uso específico de cada elemento del sistema o el contexto procedimental en que reside el sistema.

Tratar con un sistema complejo en organizaciones significa tratar con una situación problemática todavía no administrable. Se sabe que existe un problema, pero no se entiende ni se logra estructurarlo. Un sistema complejo se caracteriza por la independencia de un número grande de elementos además de que los elementos que componen un sistema pueden también representar un macro elemento de un sistema todavía mayor. Un macro elemento es un sistema basado en computadoras que forma parte de un sistema basado en computadoras, una multiplicidad de percepciones. Serán distintivos de esta clase de sistemas también la adaptación, auto-organización y propiedades emergentes.

Como se mencionó con anterioridad un sistema está compuesto de una colección de componentes en la mayoría de los casos interactivos. Un componente puede ser un sistema más pequeño aunque no necesariamente y los componentes pueden ser el resultado de algún otro sistema.

La evolución y crecimiento de los sistemas de cómputo se ha dado actualmente acompañado de una creciente complejidad. El modelo cliente-servidor ha sido punto de partida y el objetivo constante: la comunicación de los sistemas a través de las redes compartiendo información y recursos para el funcionamiento productivo acorde a las necesidades de información de una organización y sus usuarios.

Esta comunicación entre los sistemas ha hecho posible el desarrollo de grandes redes de computadoras como son las redes hechas por las organizaciones con información privada formando intranets y con información pública conectándose a la red de redes: Internet, el crecimiento va en forma proporcional a las necesidades de obtención de información y por ende a la creciente masa de datos y recursos demandados en estas redes.

Los usuarios ya no están limitados a que su estación de trabajo tenga muy poca memoria, poca capacidad en disco duro o que no cuente con alguno de estos

recursos para poder hacer su trabajo, mientras tenga una conexión a una red en la cual tendrá acceso a servidores previamente configurados hacia estos clientes, gracias a un sistema operativo como UNIX con las características siguientes:

- Multitarea – Permite al servidor ejecutar múltiples procesos simultáneamente.
- Multiusuario – Hace posible el acceso a más de un usuario al mismo recurso del sistema
- Proceso distribuido – Permite el uso de recursos compartidos a través de la red.

Son estas las características que hacen posible tener servidores de correo electrónico, licencias, aplicaciones, arranque, instalación, base de datos e impresión entre otros, dando su servicio a clientes que cuenten básicamente con un monitor, teclado, ratón e interfaz de red.

Los avances de los sistemas de cómputo y sus redes desde sus inicios han intentado cubrir la creciente necesidad de información útil y pronta de organizaciones y usuarios para una oportuna toma de decisiones acrecentando también la cantidad de datos a manejar haciendo que estos y los sistemas sean de misión crítica[1] y de aquí que los sistemas adquieran complejidad.

[1] El concepto misión crítica se explicará en el subtema 1.5

1.2 DIFERENTES AQUITECTURAS Y PROTOCOLOS DE COMUNICACIÓN EN REDES DE CÓMPUTO.

Cuando se describe una computadora, frecuentemente se distingue entre arquitectura y organización de la computadora. Aunque es difícil dar una definición precisa para estos términos, existe un consenso sobre las áreas generales cubiertas por cada uno.

La arquitectura de computadoras se refiere a los atributos de un sistema que son visibles a un programador, o para decirlo de otra manera, aquellos atributos que tienen un impacto directo en la ejecución lógica de un programa. La organización de computadoras se refiere a las unidades funcionales y sus interconexiones, que materializan especificaciones arquitectónicas. Entre los ejemplos de atributos arquitectónicos se encuentran el conjunto de instrucciones, el número de bits usados para representar varios tipos de datos (Ej., números, caracteres), dispositivos de E/S y técnicas para direccionamiento de memoria. Entre los atributos de organización se incluyen aquellos detalles del hardware transparentes al programador, tales como señales de control, interfaces entre la computadora y los periféricos y la tecnología de memoria usada.

Para poner un ejemplo, una cuestión de diseño arquitectónico es si la computadora tendrá la instrucción de multiplicar. Una cuestión de organización es si esa instrucción será implementada por una unidad especializada en multiplicar o por un mecanismo que haga un uso iterativo de la unidad de suma del sistema. La decisión de arquitectura puede estar basada en la frecuencia prevista del uso de la instrucción de multiplicar, y, por el contrario, la decisión de organización en la velocidad relativa de las dos aproximaciones, y el costo y el tamaño físico de una unidad especializada en multiplicar.

Históricamente, y aún hoy día, la distinción entre arquitectura y organización ha sido importante. Muchos fabricantes de computadoras ofrecen una familia de modelos, todos con la misma arquitectura pero con diferencias en la organización.

Para definir los diferentes tipos de arquitectura tendremos que dividirla en dos grandes aspectos, la primera en hardware y la segunda en software.

1.2.1 Arquitectura de hardware

Hoy en día, los programas cada vez más grandes y complejos demandan mayor velocidad en el procesamiento de información, lo que implica la búsqueda de procesadores (CPU) más rápidos y eficientes.

Los avances y progresos en la tecnología de semiconductores, han reducido las diferencias en las velocidades de procesamiento de los microprocesadores con las velocidades de las memorias, lo que ha reproducido en nuevas tecnologías en el desarrollo de microprocesadores. Existen dos tipos de arquitecturas la CISC y

RISC que a continuación estudiaremos. Veamos primero cual es el significado de los términos CISC y RISC:
CISC (complex instruction set computer) Computadoras con un conjunto de instrucciones complejo
RISC (reduced instruction set computer) Computadoras con un conjunto de instrucciones reducidas.

Los atributos complejos y reducidos describen las diferencias entre los modelos de arquitectura para microprocesadores solo de forma superficial. Se requiere de muchas otras características esenciales para definir los RISC y los CISC típicos.

Así, los términos complejo y reducido, expresan muy bien una importante característica definitiva, siempre que no se tomen solo como referencia las instrucciones, sino que se consideren también la complejidad del hardware del procesador.

Con tecnologías de semiconductores comparables e igual frecuencia de reloj, un procesador RISC típico tiene una capacidad de procesamiento de dos a cuatro veces mayor que la de un CISC, pero su estructura de hardware es tan simple, que se puede realizar en una fracción de la superficie ocupada por el circuito integrado de un procesador CISC.

Para aplicar una determinada arquitectura de microprocesador son decisivas las condiciones de realización técnica y sobre todo la rentabilidad, incluyendo los costos de software. Existían y existen razones de compatibilidad para desarrollar y utilizar procesadores de estructura compleja así como un extenso conjunto de instrucciones.

Aunque la tecnología de proceso de encapsulado son vitales en la elaboración de procesadores más rápidos, es la arquitectura del procesador lo que hace la diferencia entre el rendimiento de un CPU (Control Process Unit) y otra. Y es en la evaluación de las arquitecturas RISC y CISC donde centraremos nuestra atención.

1.2.1.1 Arquitectura CISC

La microprogramación es una característica importante y esencial de casi todas las arquitecturas CISC. La microprogramación significa que cada instrucción de máquina es interpretada por un microprograma localizado en una memoria en el circuito integrado del procesador.

En la década de los sesentas la microprogramación, por sus características, era la técnica más apropiada para las tecnologías de memoria existentes en esa época y permitía desarrollar también procesadores con compatibilidad ascendente. En consecuencia, los procesadores se dotaron de poderosos conjuntos de instrucciones.

Las instrucciones compuestas son decodificadas internamente y ejecutadas con una serie de microinstrucciones almacenadas en un ROM interna. Para esto se requieren de varios ciclos de reloj (al menos uno por microinstrucción).

La tendencia ha sido hacia un conjunto de instrucciones más ricos, que incluyen un número mayor de instrucciones más complejas. Dos razones principales han motivado esta tendencia:

- El deseo de simplificar los compiladores.
- El deseo de mejorar las presentaciones.

Sirvió de base a estas razones el cambio de los programadores hacia los lenguajes de alto nivel, con lo cual los arquitectos intentaron diseñar máquinas que proporcionaran mejor soporte para los lenguajes de alto nivel.

Simplificación de los compiladores. La tarea del escritor de compiladores es generar una secuencia de instrucciones máquina para cada sentencia de los lenguajes de alto nivel. Si existen instrucciones máquina que se parecen a sentencias de lenguajes de alto nivel, la tarea es simplificar.

Las instrucciones máquina complejas son con frecuencia difíciles de explotar ya que el compilador debe descubrir aquellos casos que se ajustan perfectamente a la construcción. La tarea de optimizar el código generado para minimizar su tamaño, reducir el número de instrucciones ejecutadas, mejorar la segmentación es mucho más difícil con un conjunto de instrucciones complejo.

Mejorar las presentaciones. La otra razón importante mencionada es la esperanza de que un CISC produzca:

- Programas más pequeños.
- Programas más rápidos

Examinemos los dos aspectos de esta afirmación: que los programas serán más pequeños y que se ejecutarán más rápido.

1) Los programas más pequeños tienen dos ventajas:

   a) Como el programa ocupa menos memoria, hay un ahorro de este recurso.
   b) Programas más pequeños mejoran las presentaciones, lo que ocurre por dos motivos:
      - Menos instrucciones significa que hay que captar menos bytes de instrucción.
      - En un entorno paginado programas más pequeños ocupan menos páginas, reduciendo la falta de página.

En muchos casos, el programa para el CISC, expresado en lenguaje máquina simbólico, puede ser más corto (esto es, tiene menos instrucciones), pero el número de bits de memoria que ocupa no tiene por qué ser más pequeño.

Hay varias razones para estos resultados algo sorprendentes. Hemos mencionado ya que los compiladores en los CISCs tienden a elegir las instrucciones más sencillas, de manera que la concisión de las instrucciones complejas raramente entran en juego. También, debido a que hay más instrucciones en un CISC, son precisos códigos de operación más largos, produciendo instrucciones más largas.

2) La ejecución de instrucciones fuera más rápida.

Parece tener sentido el que una operación compleja de un lenguaje de alto nivel se ejecute más rápido como una única instrucción máquina que como una sucesión de instrucciones más primitivas. Sin embargo, debido a la propensión a usar las instrucciones más sencillas, esto puede no ser así. La unidad de control completa debe hacerse más compleja, y/o el almacenamiento del microprograma de control ha de hacerse más grande, para proveer un conjunto de instrucciones más rico. Cualquiera de los dos factores aumenta el tiempo de ejecución de las instrucciones simples.

De hecho, algunos investigadores han encontrado que la aceleración en la ejecución de funciones complejas se debe no tanto a la potencia de las instrucciones máquina complejas como a su residencia en el rápido almacenamiento de control. En realidad, el almacenamiento de control actúa como un cache de instrucciones. De este modo, el arquitecto del hardware está en posición de intentar determinar qué subrutinas o funciones serán usadas con mayor frecuencia y asignarlas al almacenamiento de control implementándolas en microcódigo. Los resultados han sido poco alentadores.

1.2.1.2 Arquitectura RISC

Aunque los sistemas RISC se han definido y diseñado de diversas formas, los elementos clave compartidos por la mayoría de los diseños son:

- Un conjunto de instrucciones limitado y sencillo. A este modelo se le conoce como Load/Store (carga/almacena). Sólo las instrucciones Load/Store acceden a memoria; las demás operaciones tienen lugar en un gran conjunto de registros. Ello simplifica el direccionamiento y acorta los tiempos de los ciclos de el CPU, y además facilita la gestión de los fallos de páginas en entornos de memoria virtual. Además, permite un elevado nivel de concurrencia a consecuencia de la independencia de las operaciones de Load/Store de la ejecución del resto de las instrucciones.
- Arquitectura no destructiva de tres direcciones. Las instrucciones RISC, con tres direcciones, contienen los campos de los dos operandos y de su resultado, por lo tanto, los operandos origen como destino, son

metidos en los registros tras haber sido completada la operación. Esta arquitectura no destructiva permite a los compiladores organizar las instrucciones de modo que mantengan llenos los conductos (pipelines) del chip, y por lo tanto reutilizar los operandos optimizando la concurrencia.

- Instrucciones simples de formato fijo con pocos modos de direccionamiento. Las instrucciones simples reducen de manera muy significativa el esfuerzo para su decodificación, y favorecen su ejecución en pipelines. Las instrucciones de longitud fija, con formatos fijos, implican que los campos de código de operación y de los operandos están siempre codificados en las mismas posiciones, permitiendo el acceso a los registros al mismo tiempo que se está decodificando el código de operación. Todas las instrucciones tienen una longitud equivalente a una palabra y están alineadas en la memoria en límites de palabra, ya que no pueden ser repartidas en pedazos que puedan estar en diferentes páginas.
- Ausencia de microcódigo. El microcódigo no se presta a la ejecución en ciclos únicos, ya que requiere que el hardware sea dedicado a su interpretación dinámica. La programación en microcódigo no hace que el software sea más rápido que el programado con un conjunto de instrucciones simples. Todas las funciones y el control, en los procesadores RISC, están cableados, para lograr una máxima velocidad y eficiencia.
- Ejecución en conductos (pipeline). Las instrucciones simples, de formato fijo y ciclo único permiten que las diferentes etapas de los ciclos de ejecución (búsqueda o fetch, decodificación, ejecución, y escritura del resultado) para instrucciones múltiples, se puedan realizar simultáneamente, de un modo más simple y eficaz.
- Ejecución en ciclos únicos. El resultado directo de los conjuntos de instrucciones que ofrecen los procesadores RISC, es que cada instrucción puede ser ejecutada en un único ciclo de reloj. Este tipo de ejecución en ciclos únicos también simplifica la gestión de las interrupciones y los conductos.

1.2.2 Arquitectura de software

Un sistema operativo puede verse como una colección organizada de software que extiende al hardware y que consta de rutinas de control para operar una computadora y proporcionar un entorno para la ejecución de programas. Otros programas confían en las facilidades proporcionadas por el sistema operativo para conseguir acceder a los recursos del sistema de la computadora, tales como archivos y dispositivos de entrada / salida. Los programas invocan los servicios del sistema operativo mediante llamadas del sistema operativo. Además, los usuarios puedan interactuar con el sistema operativo directamente mediante las órdenes del sistema operativo. En ambos casos, el sistema operativo actúa como una interfaz entre los usuarios y el hardware de un sistema de computadora. En

general, el objetivo primordial de un sistema operativo es incrementar la productividad de un recurso de producción, tal como el hardware de una computadora o los usuarios de un sistema computacional.

Los sistemas operativos diseñados para computadoras de alto desempeño, estaban relacionados principalmente con producir tanto trabajo como fuera posible desde el equipo. La conveniencia del usuario y la productividad eran consideraciones secundarias. En el extremo opuesto del espectro se puede diseñar un sistema operativo para computadoras personales y que pueda servir a un usuario único. En este caso, es la productividad del usuario lo que se incrementará tanto como sea posible, con la utilización del hardware menos importante.

Un sistema operativo puede procesar sus cargas de trabajo en serie o concurrentemente. Esto es, se pueden dedicar los recursos de un sistema de computadora a un programa único hasta completarlo, o se puedan reasignar dinámicamente entre una colección de programas activos en diferentes estados de ejecución. Tales sistemas son conocidos frecuentemente como sistemas multiprogramados por su capacidad de ejecutar múltiples programas de manera intercalada.

1.2.2.1 Tipos de sistemas operativos

En esta parte discutiremos ciertas características de los diferentes tipos de sistemas operativos. En particular, discutiremos las propiedades generales y requerimientos básicos de sistemas por lotes y multiprogramados. También son presentados con detalle las variedades de sistemas de multiprogramación de tiempo compartido y tiempo real.

Además de algunos comentarios generales, cada tipo de sistema operativo se discute con la perspectiva de los siguientes aspectos:

a)Planificación del procesador.
b)Gestión de memoria.
c)Gestión de E/S.
d)Gestión de archivos.

1.2.2.1.1 Sistemas operativos por lotes

El procesamiento por lotes requiere que estén reunidos en forma de un trabajo el programa, los datos y las instrucciones del sistema apropiadas. Los sistemas operativos por lotes permiten normalmente poca ejecución debido a los retardos en el tiempo total de ejecución y a la depuración fuera de línea, el procesamiento por lotes no es muy conveniente para el desarrollo de programas.

Un sistema operativo por lotes puede servir muy bien para programas que no requieren interacción y aquellos con tiempos de ejecución largos.

Es muy simple la planificación en sistemas por lotes. Los trabajos son procesados en el orden de admisión, según el modelo primero llegado primero atendido (cola), se emplea en algunas ocasiones alguna otra forma de ordenación del trabajo para proporcionar una distribución razonable de los tiempos globales de ejecución, tal como el próximo trabajo el más corto.

También es muy simple en los sistemas por lotes la gestión de memoria. La memoria se divide normalmente en dos zonas. Una de ellas está ocupada permanentemente por la porción residente del sistema operativo y la otra se usa para cargar programas transitorios para su ejecución. Cuando termina un programa transitorio, se carga un nuevo programa en la misma zona de la memoria.

Los sistemas por lotes no necesitan ningún gestor de dispositivos de tiempo crítico, desde el momento que no puede estar en ejecución más de un programa a la vez. Por este motivo, muchos sistemas operativos serie y por lotes ordinario utilizan el sencillo método de E/S controlado por programas. La falta de disputa por los dispositivos de E/S hace trivial su asignación y liberación.

Los sistemas por lotes proporcionan frecuentemente formas simples de gestor de archivos. Como el acceso a los archivos también es en serie, se necesita poca protección y ningún control de concurrencia de acceso de archivos.

1.2.2.2 Sistemas operativos de multiprogramación

La ejecución concurrente de programas tiene un potencial significativo para mejorar el rendimiento total del sistema y la utilización de recursos con respecto al proceso por lotes y serie. Este potencial se realiza, o al menos se explota, mediante una clase de sistema operativo que multiplique los recursos del sistema de computadora entre una multitud de programas activos; tales sistemas operativos tienen normalmente en sus nombres, el prefijo multi, como multitareas o multiprogramación.

Un sistema de un programa en ejecución se llama proceso o una tarea. Un sistema operativo multiproceso, también llamado sistema operativo multitarea, se distingue por sus habilidades para soportar dos o más procesos activos simultáneamente. El término multiprogramación denota un sistema operativo que, además de soportar procesos concurrentes múltiples, permite que residan simultáneamente en la memoria primaria las instrucciones y los datos procedentes de dos o más procesos disjuntos. Señalar que la multiprogramación implica la operación de multiprocesos, pero la operación de multiprocesos (o multitareas) no implica multiprogramación. En efecto, la operación multiproceso que un sistema operativo multiprogramado emplea en la gestión de la totalidad de los recursos del sistema de computadora, incluyendo la unidad central de procesos, la memoria y los dispositivos de E/S.

Los sistemas operativos multiacceso o multiusuario, permiten acceder simultáneamente a un sistema de microcomputadora a través de dos o más terminales. Aunque asociados frecuentemente con la multiprogramación, las operaciones multiusuario no implican, ni está implícito, la multitarea o la multiprogramación. Por el contrario, los sistemas de tiempo compartido de propósito general incorporan tanto multiprogramación como operaciones multiusuario. Se pueden encontrar la operación de multiproceso sin soporte multiusuario en los sistemas operativos de algunas computadoras personales avanzadas y en sistemas de tiempo real (este tipo de sistema se explicará en párrafos siguientes).

En general, todos los sistemas de multiprogramación se caracterizan por una multitud de programas activos simultáneamente que compiten por los recursos del sistema, como el procesador, la memoria y los dispositivos E/S. Un sistema operativo de multiprogramación monitorea el estado de todos los programas activos y recursos del sistema. Se activa el sistema operativo para asignar recursos y proporcionar ciertos servicios de su repertorio cuando ocurre un cambio de estado importante, o cuando es llamado explícitamente. Como se verá rápidamente, los requerimientos del entorno específico que se va a servir influyen en la elección de los objetivos y las estrategias del sistema operativo asociado.

1.2.2.3 Sistemas operativos de tiempo compartido

El tiempo compartido es un representante popular de sistemas multiprogramados y multiusuario. Además de los entornos generales de desarrollo de programas. En contraste, los sistemas multiacceso dedicados ejecutan la mayor parte del tiempo esencialmente un programa único de una gran aplicación.

Uno de los objetivos principales de los sistemas multiusuario en general, y de los de tiempo compartido en particular, es un buen tiempo de respuesta de la terminal. Los sistemas de tiempo compartido tratan con frecuencia de proporcionar un tiempo equitativo de los recursos comunes para dar la ilusión a cada usuario de poseer la máquina para él mismo.

Se refleja esta filosofía en la elección de algoritmos planificados. En este método, los programas se ejecutan con prioridad rotatoria que se incrementan durante las esperas y cae después de que dispensó el servicio. El sistema operativo interrumpe un programa y lo pone al final de la cola de programas en espera, cuando es más largo que la fracción de tiempo definida por el sistema, para prevenir a los programas de la monopolización del procesador, este método de operación proporciona generalmente un tiempo de respuesta rápido a los programas interactivos.

El gestor de memoria en sistemas de tiempo compartido proporciona el aislamiento y la protección de los programas corresidentes, en ocasiones se proporciona algunas formas de compartición controlada para conservar la memoria y permitir el intercambio de datos entre programas. Los programas en

sistemas de tiempo compartido generalmente no tienen muchas necesidades de comunicarse con otros, debido a que se ejecutan a petición de diferentes usuarios.

En los sistemas de tiempo compartido, el gestor de entrada/salida se debe sofisticar lo suficiente para cooperar con los múltiples usuarios y dispositivos. Sin embargo, debido a las lentas velocidades de las terminales las redes y de los usuarios humanos, el procesamiento de las interrupciones de la terminal no necesitan ser críticas para el tiempo. Como en la mayoría de los entornos multiusuarios, se debe hacer la asignación y liberación de los dispositivos de forma que se preserve la integridad de los sistemas y proporcione un buen rendimiento.

El gestor de archivos en un sistema de tiempo compartido debe proporcionar protección y control de acceso, dada la posibilidad de concurrencia y de conflictos al tratar de acceder a los archivos. Esta tarea se acomoda a los requerimientos de los archivos compartidos por ciertos usuarios o clases de usuarios.

1.2.2.4 Sistemas operativos de tiempo real

Se usan los sistemas operativos en tiempo real en entornos donde se deben aceptar y procesar en tiempo breve y sin tiempos muertos gran número de sucesos, en su mayoría externos al sistema de computadora.

Proporcionar tiempos rápidos de respuesta y así hacer frente a los tiempos muertos de planificación es un objetivo primario de los sistemas en tiempo real. Son asuntos secundarios la conveniencia del usuario y la utilización de los recursos. No es raro para un sistema en tiempo real el proceso súbito de miles de interrupciones por segundo sin perder un solo suceso. Tales requerimientos no pueden abordarse normalmente con la multiprogramación en solitario, y los sistemas operativos en tiempo real usualmente ponen su confianza en ciertas estrategias y técnicas específicas para hacer su trabajo. Básicamente, se encarga un proceso separado de manejar un suceso externo único. El proceso se activa al ocurrir el suceso relacionado, señalado frecuentemente por una interrupción. Se consigue el multiproceso planificando los procesos independientemente unos de otros. Se asigna a cada proceso un cierto nivel de prioridad que corresponde a la importancia relativa de los sucesos que sirve. El procesador está normalmente asignado al proceso con más alta prioridad entre los que están listos para ejecutarse. Los procesos de más alta prioridad toman por derecho la ejecución de los procesos de prioridad inferior. Esta forma de planificación basada en la prioridad preferente, se usa por la mayoría de los sistemas en tiempo real, el gestor de memoria está comparativamente menos solicitado que en otros tipos de sistemas de multiprogramación. La razón principal para ello es que muchos procesos residen permanentemente en memoria para proporcionar tiempos de respuesta rápidos. Al contrario que los de tiempo compartido, la población de los procesos en sistemas de tiempo real está casi estática y hay comparativamente poco movimiento de programas entre el almacenamiento primario y secundario. Por otra parte, los procesos en sistemas de tiempo real tienden a cooperar

estrechamente, así necesariamente soportan la separación y compartición de memoria.

Como ya mencionamos, el tiempo crítico del gestor de dispositivos es una de las características principales de los sistemas en tiempo real. Además de las formas sofisticadas proporcionadas de gestión de interrupciones y almacenamiento intermedio, los sistemas operativos en tiempo real proporcionan frecuentemente llamadas del sistema que permiten los procesos (programas) de usuario conectarse directamente a vectores de interrupción y sucesos de servicio.

El gestor de archivos se encuentra normalmente sólo en grandes instalaciones de sistemas en tiempo real. De hecho, algunos sistemas reducidos en tiempo real, tal como un controlador autopropulsado, puede no tener ningún almacenamiento secundario. Sin embargo, donde se proporcione el gestor de archivos de sistema en tiempo real, debe satisfacer muchos de los requerimientos encontrados en sistemas de tiempo compartido y de multiprogramación. Esto incluye protección y control de acceso. El objetivo principal del gestor de archivos en sistemas de tiempo real es usualmente la velocidad de acceso, más que la utilización eficaz del almacenamiento secundario.

1.2.2.5 Sistemas operativos combinados

Como hemos pensado, están optimizados, o al menos muy acoplados para servir las necesidades de ciertos entornos específicos. En la práctica, sin embargo, un entorno dado puede no encajar exactamente en ninguno de los moldes descritos. Un sistema en tiempo compartido puede soportar usuarios interactivos mientras incorpora también un monitor por lotes completamente maduro.

1.2.3 Arquitectura de redes

La comunicación de datos se ha convertido en una parte fundamental de la computación. Las redes globales reúnen datos sobre temas diversos, como las condiciones atmosféricas, la producción de cosechas y el tráfico aéreo. Algunos grupos establecen listas de correo electrónico para poder compartir información de interés común. Las personas que tienen pasatiempos intercambian programas para su computadora personales. En el mundo científico, las redes de datos son esenciales pues permiten a los científicos enviar programas y datos hacia supercomputadoras remotas para su procesamiento, recuperar los resultados e intercambiar información con sus colegas.

Por desgracia, la mayor parte de las redes son entidades independientes, establecidas para satisfacer las necesidades de un solo grupo. Los usuarios escogen una tecnología de hardware apropiada a sus problemas de comunicación. De manera más importante, es imposible construir una red universal desde una sola tecnología de hardware, debido a que ninguna red satisface todas las necesidades de uso. Algunos usuarios necesitan una red de alta velocidad para conectar máquinas, pero dichas redes no se pueden expandir

para abarcar grandes distancias. Otros establecen una red de menor velocidad que conectan máquinas que se encuentran a miles de kilómetros de distancia.

1.2.4 Protocolos de comunicación

NetWare de Novell: Se diseñó para que lo usaran compañías que deseaban cambiar mainframe por una red de PC. En tales sistemas, cada usuario tiene una computadora de escritorio que funciona como cliente. Además, varias PC de alta capacidad operan como servidores para proveer de servicios de archivos, de bases de datos y otros a una colección de clientes. En otras palabras, el NetWare se basa en el modelo cliente-servidor.

Netware usa una pila de protocolos que se basa en el antiguo Xerox Network System, XNS. Las capas físicas y de enlace de datos se pueden escoger de entre varios estándares de la industria, lo que incluye Ethernet, token ring de IBM Y ARCnet. La capa de red utiliza un protocolo de interred no confiable, sin conexión, llamado IPX. Este protocolo transfiere paquetes del origen al destino en forma transparente, aun si la fuente y el destino se encuentra en redes diferentes.

Por encima de IPX está un protocolo de transporte orientado a la conexión que se llama NCP (network core protocol, protocolo central de red). El NCP proporciona otros servicios además del transporte de datos de usuario y en realidad es el corazón de Netware. También está disponible un segundo protocolo, SPX, pero sólo proporciona transporte. Las aplicaciones pueden seleccionar cualquiera de ellos. En la capa de aplicación están presentes varios protocolos de aplicación. La clave de todo la arquitectura es el paquete de datagrama de interred sobre el cual se construye todo lo demás. Aproximadamente cada minuto, cada servidor difunde un paquete con su dirección que indica cuáles servicios ofrece. Estas difusiones usan el SAP (service advertising protocol, protocolo de publicidad del servicio). Procesos de agentes especiales que se ejecutan en las máquinas enrutadoras que detectan y recopilan los paquetes. Los agentes usan la información contenida en los paquetes para construir bases de datos que indican cuáles servidores se ejecutan y dónde.

Cuando se arranca una máquina cliente, emite una petición en la que pregunta dónde está el servidor más cercano. El agente en la máquina del enrutador local detecta esta solicitud y busca en su base de datos de servidores cuál es el mejor servidor para su solicitud. A continuación se devuelve al cliente la dirección del mejor servidor a usar. Ahora el cliente puede establecer una conexión NCP con el servidor. Mediante esta conexión, el cliente y el servidor negocian el tamaño máximo de paquetes. De aquí en adelante, el cliente puede acceder al sistema de archivos y a otros servicios usando esta conexión. También puede hacer consultas a la base de datos de servidores para buscar otros servidores (más distantes).

ARPANET: A mediados de la década de 1960, en la cúspide de la Guerra Fría, el DoD quería una red de comandos y control que pudiera sobrevivir a una guerra nuclear. Las redes telefónicas tradicionales de circuitos conmutados se

consideraban muy vulnerables, puesto que la pérdida de una línea o un conmutador ciertamente terminaría toda conversación que los estuviera usando y podría incluso partir la red. Para resolver este problema, el DoD acudió a su rama de investigación, ARPA o Advanced Research Projects Agency, es decir, la Agencia de Proyectos de Investigación Avanzados. Después de discusiones con varios expertos, la ARPA decidió que la red que necesitaba el DoD debía ser una red de paquete conmutado, que consistía en una subred y computadoras hosts.

La subred consistiría en minicomputadoras llamadas IMP (interface message processors, procesadores de intercomunicación de mensajes) conectadas por líneas de transmisión. Para lograr alta confiabilidad, cada IMP se conectaría al menos a otras dos. La subred iba a ser una subred de datagramas, de modo que si algunas líneas e IMP resultaban destruidas, los mensajes se podrían reencaminar de forma automática a través de trayectorias alternas.

Cada nodo de la red consistiría en un IMP y un hosts en el mismo cuarto, conectados por un cable corto. Un host podría enviar mensajes de hasta 8063 bits a su IMP, que entonces los dividiría en paquetes de 1008 bits a lo sumo y los reenviaría a su destino en forma independiente. Cada paquete se recibía en su totalidad antes de reenviarse, por lo que la subred fue la primera red electrónica de paquetes de almacenar y reenviar.

El software se dividió en dos partes: subred y host consistió en el extremo IMP de la conexión host-IMP, el protocolo IMP-IMP y un protocolo de IMP fuente a IMP destino diseñado par mejorar la confiabilidad.

Fuera de la subred también se necesitaba software, a saber, el extremo host de la conexión host-IMP, el protocolo host-host y el software de aplicación. Para enfrentar el problema del software de el host, ARPA convocó a una reunión de investigadores de redes, la mayoría estudiantes graduados. Los estudiantes esperaban que un experto en redes les explicara el diseño de la red y su software y luego les asignara el trabajo de escribir una parte del mismo a cada uno de ellos. Quedaron pasmados cuando no hubo expertos en redes y tampoco un gran diseño. Tuvieron que descifrar qué hacer por sí mismos.

Más tarde, el software de IMP se cambió para permitir que las terminales se conectaran de forma directa a un IMP especial llamado TIP (terminal interface processor, procesador de interfaz de terminal) sin tener que pasar por un host. Los cambios subsecuentes incluyeron el tener múltiples hosts por cada IMP, hosts que se comunicaban con múltiples IMP (para protegerse de fallas del IMP) y hosts e IMP separados por una gran distancia (para alojar a los hosts situados lejos de la subred).

ARPA financió también investigaciones sobre redes de satélites y redes de radio de paquetes móviles. En una demostración famosa, un camión que recorría California usó la red radial de paquetes para enviar mensajes a SRI, que entonces lo reenvió por conducto de ARPANET a la Costa Este, desde donde se

transmitieron a la University College en Londres mediante una red de satélites. Esto permitió a un investigador en el camión usar una computadora en Londres mientras conducía por California. Este experimento demostró también que los protocolos de ARPANET existentes no eran apropiados para funcionar en múltiples redes. Esta observación condujo a más investigaciones sobre protocolos, lo que culminó con la invención del modelo y los protocolos TCP/IP. TCP/IP se diseño de manera específica para manejar la comunicación en las interredes, algo que se volvió cada vez más importante al conectarse más y más redes a la ARPANET.

Para fomentar la adopción de estos protocolos nuevos, la ARPA concedió varios contratos a la compañía BBN y a la universidad Berkeley para integrarlos en el UNIX de Berkeley. Los investigadores de esta universidad desarrollaron una interfaz de programa conveniente para la red (sockets) y escribieron muchos programas de aplicación, utilería y administración para facilitar el trabajo con las redes.

NSFNET.: A finales de la década de 1970, la NSF (National Science Foundation, Fundación Nacional de la Ciencia de Estados Unidos) vió el impacto enorme que había tenido ARPANET en la investigación universitaria al permitir que científicos de todo el país compartieran datos y colaboraran en proyectos de investigación. Sin embargo, para introducirse en la ARPANET, una universidad debía tener un contrato de investigación con el DoD, cosa que muchas no tenían. Esta falta de acceso universal motivó a la NSF a establecer una red virtual, CSNET, centrada en una sola máquina que permitía el uso de líneas de acceso por discado y tenía conexiones con la ARPANET y otras redes. Mediante CSNET, los investigadores académicos podían hacer llamadas y dejar correo electrónico para que otras personas la recogieran más tarde.

En 1984, la NSF empezó a diseñar un sucesor de alta velocidad para la ARPANET que se abriría a todos los grupos universitarios de investigación. A fin de tener algo en concreto con lo cual empezara, la NSF decidió construir una red backbone (columna vertebral) para conectar seis centros de supercomputo. A cada supercomputadora se le dio un hermanito que consistía en una microcomputadora a la que llamaron fuzzball. Las fuzzballs se conectaron con líneas rentadas de 56 kbps y formaron la subred, la misma tecnología de hardware que usó ARPANET. Sin embargo, la tecnología de software era diferente: las fuzzballs hablaban TCP/IP desde un principio, convirtiéndose en la primera WAN de TCP/IP.

INTERNET. La cantidad de redes, máquinas y usuarios conectados a la ARPANET creció con rapidez después de que TCP/IP se convirtió en el único protocolo oficial el 1 de enero de 1983. Cuando se interconectaron la NSFNET y la ARPANET, el crecimiento se hizo comercial; se unieron muchas redes regionales y se hicieron conexiones con redes en Canadá, Europa y el Pacífico.

Nuestra definición es que una máquina está en Internet si opera con el conjunto de protocolos de TCP/IP, tiene una dirección de IP y es capaz de enviar paquetes de IP a todas las demás máquinas de Internet. La mera capacidad de enviar y recibir correo electrónico no es suficiente, pues el correo electrónico se distribuye a muchas redes fuera de Internet. Sin embargo, el asunto pierde claridad en cierta forma por el hecho de que muchas computadoras personales tienen la capacidad de llamar a un proveedor de servicios de Internet mediante un MODEM, recibir la asignación de una dirección de IP temporal y enviar paquetes IP a otro host de Internet. Tiene sentido considerar que tales máquinas están en Internet mientras están conectadas al ruteador del proveedor del servicio.

La tecnología que se ha descrito hasta este momento es un ejemplo de interconexión del sistema abierto. Se llama sistema abierto porque, a diferencia de los sistemas privados de comunicación disponible por medio de vendedores particulares, las especificaciones están disponibles públicamente. Por lo tanto, cualquier persona puede desarrollar el software necesario para comunicarse a través de una red de redes. Algo muy importante es que toda la tecnología ha sido diseñada para permitir la comunicación entre máquinas que tengan arquitecturas diferentes de hardware, para utilizar cualquier hardware de red de paquetes conmutados y para incorporar muchos sistemas operativos de computadoras.

Durante los pasados años, ha evolucionado una nueva tecnología que hace posible interconectar muchas redes físicas diferentes y hacerlas funcionar como una unidad coordinada. Esta tecnología, llamada internetworking, unifica diferentes tecnologías de hardware subyacentes al proporcionar un conjunto de normas de comunicación y una forma de interconectar redes heterogéneas. La tecnología de rede de redes oculta los detalles del hardware de red permiten que las computadoras se comuniquen de forma independiente de sus conexiones físicas de red.

Una persona no puede apreciar los detalles técnicos subyacentes de TCP/IP sin entender los servicios que proporcionan. Casi todo el análisis de los servicios se enfoca en estándares llamados protocolos. Protocolos como el TCP y el IP proporcionan reglas para la comunicación. Contienen los detalles referentes a los formatos de los mensajes, describen cómo responde una computadora cuando llega un mensaje y especifican de qué manera una computadora maneja un error u otras condiciones anormales. Un aspecto importante es que permite reflexionar sobre la comunicación por computadora de manera independiente de cualquier hardware de red de cualquier marca. En cierto sentido, los protocolos son para las comunicaciones lo que los algoritmos para la computación. Un algoritmo permite especificar o entender un cómputo aunque no se conozcan los detalles de un juego de instrucciones de CPU. De manera similar, un protocolo de comunicaciones permite especificar o entender la comunicación de datos sin depender de un conocimiento detallado de una marca en particular de hardware de red.

Hay muchos proveedores de servicios de red, cada uno con sus propias ideas acerca de cómo deben hacerse las cosas. Sin coordinación, existiría un caos completo, y los usuarios nunca lograrían hacer nada. La única manera es acordar ciertos estándares de redes.

Los estándares no sólo permiten a diferentes computadoras comunicarse, sino que también incrementan el mercado para los productos que se ajustan a la norma, lo cual conduce a la producción en masa, las economías de escala en la producción, la implementaciones de alta escala de integración en los circuitos, y otros beneficios que disminuyen el precio y aumentan la aceptación posterior.

El hacer a un lado los detalles de bajo nivel de la comunicación nos ayuda a mejorar la productividad de muchas marcas. Primero, debido a que los programas tienen que manejar abstracciones de protocolos de un nivel más elevado, no necesitan aprender o recordar tantos detalles sobre una configuración de hardware en particular. Pueden crear con rapidez nuevos programas. Segundo, con los programas hechos por medio de abstracciones de un nivel más elevado no se encuentran restringidos a una sola arquitectura de máquina o a un solo tipo de hardware de red, no se necesitan cambiar cuando se reconfiguran las máquinas o las redes. Tercero, puesto que los programas de aplicación hechos mediante protocolos de un nivel más elevado son independientes del hardware subyacente, pueden proporcionar comunicación directa entre un par arbitrario de máquinas. Los programadores no necesitan hacer versiones especiales de software de aplicación para mover y traducir datos entre cada par de máquinas posibles. Veremos por lo tanto que todos los servicios de red se encuentran descritos por protocolos.

1.3 DIFERENTES TIPOS DE USUARIOS ASÍ COMO DE SUS NECESIDADES

En una organización se tiene diferentes tipos de usuarios, la agrupación de estos dependen del tipo de organización y de recursos que dentro del sistema de cómputo se estén usando. Se podría generalizar agrupando en cuatro tipos de usuarios en una organización: el usuario final, el desarrollador, el administrador de las aplicaciones y el administrador del sistema.

El usuario final es aquel que normalmente esta conectado a través de una estación de trabajo al cual tiene instalados programas cliente para que éste pueda hacer su trabajo usualmente de captura y consulta de datos de forma restringida, es decir, sin modificar aquellos que sean esenciales en la integridad y consistencia de estos tales como la fecha y hora.

El usuario desarrollador tiene un acceso al sistema menos restringido que el anterior debido a que este es un programador quien crea, modifica y libera módulos de acuerdo a los requerimientos de los usuarios finales y de la organización, necesita los permisos suficientes para que el pueda modificar los códigos fuente de determinada aplicación que da servicio a los usuarios, ejemplos de esto son el requerimiento de nuevas operaciones en la captura y nuevas ventanas con otros colores.

El usuario administrador de las aplicaciones es aquel que recibe normalmente requerimientos de los dos anteriores, es quien da los permisos de su aplicación a determinado usuario, respalda los datos de acuerdo a un esquema determinado, recupera estos cuando es requerido y hace las modificaciones necesarias para que la aplicación se adapte a las demandas de los clientes.

Por último esta el administrador del sistema que no por ser este deja de ser un usuario, es quien da un servicio general a los tres tipos de usuario anteriores, entre sus tareas están: alta y baja de usuarios, monitoreo, actualización, respaldo, recuperación, afinación y disponibilidad del sistema entre otras tareas administrativas concernientes al mantenimiento para el buen funcionamiento de este. Por otro lado es quien debe tener un plan para la evolución de su sistema de acuerdo a la los planes de expansión de una organización. Tiene por lo general el control total y es este de quien depende en primera instancia que una organización pueda trabajar con los datos.

Es el administrador del sistema quien por sus funciones es alguien crítico, pues de sus funciones depende el sistema y por ende el accionar de todos los usuarios y hasta de la misma organización.

1.4 CRECIENTE DEMANDA DEL USO DE LAS COMPUTADORAS PARA LOS NEGOCIOS

La necesidad de cálculo en operaciones básicas fué lo que dió inicio a las primeras computadoras, de inicio fueron mecánicos sin programación alguna y con manejo manual y décadas después con el uso de la electrónica se avanzó considerablemente en la capacidad y eficiencia de las operaciones, se crearon lenguajes de programación para la simplificación de pasos en operaciones a realizar y hacer posible la ejecución de programas y teniendo una muy limitada capacidad de almacenamiento y ocupación física de espacio considerablemente grande.

Al inicio del uso de las computadoras los fines básicos de éstas eran centrados en el cálculo y organización de datos de una manera relativamente rápida y eficiente teniendo así éxito. Con la evolución de la microelectrónica no sólo disminuyó el espacio físico de una máquina de cómputo sino también su capacidad de procesamiento, automatización y almacenaje de datos, éstas se fueron convirtiendo en una herramienta atractiva para las empresas, dentro de esta evolución vino el integrar la comunicación entre una máquina y otra, dando inicios a lo que hoy son el procesamiento distribuido y las redes privadas de organizaciones: Intranets y la red de redes: Internet.

Hoy día es difícil ver en una organización que el cómputo de datos sea de manera aislada, es decir, que se realicen en una sola computadora la cual no comparte o requiere algún servicio que es posible obtenerlo una vez conectado a una red, normalmente se tendría para almacenamiento pero aún, por citar un ejemplo, una base de datos la cual no presenta cambios considerables, está hecha para ser consultada por todo aquel que esté conectado en red y tenga acceso a ésta siendo también una opción más de recurso de cómputo dentro de una conexión.

Por otro lado ya es común ver el gran desarrollo que han tenido las aplicaciones comerciales e industriales que tienen como objetivo primordial acortar tiempo y distancia en la obtención de datos procesados y almacenados, necesitando de los sistemas de cómputo una interfaz de red configurada para funcionar y cubrir las necesidades tanto de usuarios internos como externos.

El desarrollo progresivo tanto de hardware y software ha sido encaminado a tener servicios de información en red y aplicaciones multimedia donde la comunicación es un requerimiento básico.

Gracias a esto es posible la toma de decisiones oportuna para una organización sustentada en la eficiencia del manejo de datos de su red de cómputo, es posible obtener clientes quienes llegan como potenciales usuarios de los recursos que se ofrezca en una red, es decir, la comunicación eficiente de datos dentro y fuera actualmente es lo que da una parte muy importante de la solidez de una empresa.

No es posible que en el manejo de datos en la actualidad dominen aplicaciones de cómputo aisladas y hacer la comunicación a través de medios que consuman tiempo con el que no se cuenta y mucho menos tener un medio de almacenamiento sólo en papel.

Los negocios en la actualidad ya no sólo demandan el equipo de cómputo necesario para el manejo de datos, se tiene también la necesidad de aplicaciones para dar presentación a los datos y toda una infraestructura de red y tecnologías necesarias para hacer que los datos estén disponibles en los lugares y tiempos necesarios.

1.5 DATOS DE MISIÓN CRÍTICA.

Desde los inicios de la civilización, la humanidad ha necesitado información como una ayuda en la lucha por la supervivencia, así como en los intentos para administrar las organizaciones.

Antes del siglo XVIII había dos razones principales para registrar datos. En primer lugar, los hombres tenían el deseo natural de llevar la cuenta de sus propiedades y riquezas, los mercaderes babilónicos mantenían registros desde el año 3,500 a.c. es obvio que a medida que aumentó el intercambio y el comerció, los hombres necesitaron cada vez más medios para estar al tanto de los detalles y de la situación de los negocios.

La segunda razón para registrar datos antes del siglo XVIII la constituían los requerimientos gubernamentales. A medida que las tribus se transformaron en naciones, las autoridades de esas naciones (Egipto, Grecia, etc.) recopilaron investigaciones administrativas para que se emplearan en la recaudación de impuestos y en el reclutamiento de soldados.

A mediados del siglo XVIII se crearon todavía más exigencias para que se registraran los datos de manera formal. La revolución industrial había trasladado a las fábricas las tareas básicas de la producción en el hogar y los pequeños talleres. El desarrollo de estas grandes organizaciones manufactureras originó el desarrollo de otras industrias de servicios, tales como la investigación de mercados y la transportación. De esta manera, el gran tamaño y la complejidad de tales organizaciones hicieron imposible que un individuo administrara de manera eficaz una organización sin algún procesamiento de datos que proporcionara información adicional. Además, con la aparición de sistemas de grandes fábricas y las técnicas de producción más modernos requirió de inversiones mayores. Estas grandes necesidades financieras forzaron la separación entre el inversionista y la administración. Por un lado, la administración necesitaba de más información para tomar decisiones internas, y por el otro lado, los inversionistas necesitaban información referente a la organización y al desempeño de la información.

A medida que aparecían las nuevas políticas de negocios aumentaban la necesidad de procesamiento de datos.

En general, los términos datos e información se usan indistintamente, aunque se refieren a dos conceptos diferentes, los cuales están íntimamente relacionados. Desafortunadamente, esta ambigüedad en el uso de la terminología a menudo da lugar a una comunicación deficiente en lo que concierne a esas dos importantes ideas.

Los datos son flujos de hechos en bruto que representan sucesos ocurridos en las organizaciones o en el entorno físico de cualquier situación, antes de ser organizados y acomodados de tal forma que las personas puedan entenderlos y usarlos.

El término información se refiere a datos a los que se le ha dado una forma que tiene sentido y es útil para los seres humanos.

Las organizaciones hoy día depende en gran medida de los equipos de cómputo pues ahí es donde se almacena información para diferentes acciones dentro de las muy variadas organizaciones, desde recibir un saludo a través de un correo electrónico hasta la toma de decisiones para inversiones en nuevos proyectos, esto, dependiendo de los diferentes tipos de usuarios así como de sus necesidades. Es muy seguro que para cada uno de los diferentes usuarios los datos almacenados dentro de los equipos tengan un valor diferente y por tanto una prioridad semejante, al encontramos con este tipo de problemática los responsables de los equipos de cómputo en conjunto con los directivos de sus compañías deciden qué datos tienen mayor relevancia, en general dependiendo de las funciones que desempeña alguna organización es como se decide qué datos son importantes o trascendentales, no se podría comparar entre una armadora de autos y una línea aérea pues sus necesidades son totalmente diferentes.

Luego entonces se puede considerar máquinas de misión crítica a las máquinas que almacenan y /o manipulan información confidencial o de suma importancia para la organización. Para ello, estas máquinas deben de contar con mecanismos de seguridad que permitan la integridad, la privacidad y la disponibilidad, así como la auditoria y control de acceso de la información, es así como, se adquiere el hardware y el software para la salvaguardia de la información. El término de sistema de misión crítica se refiere al sistema que aparte de manejar datos importantes, los resultados que da son indispensables en el momento de realizar la operación, en otras palabras, un sistema de misión crítica detendrá a la organización. Si se detiene la computadora o si la computadora o sistema se detienen por un día, se perderá más dinero que el precio de la computadora o lo que se invirtió en el sistema. De tales aseveraciones podemos definir como datos de misión crítica a todos aquellos que tengan el valor suficiente para modificar el fin para los cuales fueron almacenados y creados.

En este primer capítulo introductorio ha sido posible destacar la importancia del cómputo a través de la historia así como también del desarrollo de las herramientas necesarias para llevarlo a cabo y la importancia hoy en día de los sistemas de cómputo como algo imprescindible para el manejo y procesamiento de los datos, adquiriendo la característica de críticos debido a la importancia de contar con información.

2. DISPONIBILIDAD EN LOS SISTEMAS DE CÓMPUTO

En este capítulo veremos las formas de medir la disponibilidad en cuanto a tiempos, la perspectiva de disponibilidad dependiendo del tipo de organización de la que se hable y sus necesidades de contar con este esquema, las causas que provocan tiempo fuera de los sistemas así como los tipos de fallas que se destacan en el ambiente de cómputo, también se podrá destacar la importancia que tiene el invertir en un nivel de disponibilidad exponiendo los costos en pérdidas que representa el tiempo fuera por industria y por último cerraremos con los niveles generales de la disponibilidad complementándolos con un índice de disponibilidad el cual estará representado por los puntos convergentes que resultan entre la disponiblidad y la inversión.

2.1 MÉTRICAS DE DISPONIBILIDAD

Las organizaciones hoy en día compran sistemas de cómputo para llevar a cabo funciones de negocios asumiendo que estos pueden completar las tareas de una manera más rápida y exacta que un ser humano y con el fin de que valgan de un modo redituable el gasto que se hace al adquirir el hardware, la instalación, dispositivos periféricos, la red, aplicaciones, mantenimiento, entrenamiento al personal, etc. Es de esperarse un balance positivo debido al valor que los sistemas provean como resultado de su inversión.

Sin embargo los sistemas fallan de vez en cuando y se presentan pérdidas causadas por el tiempo que se consumió al restaurarlos, cuando una computadora falla no está dando el servicio para el cual fue comprado y es entonces cuando no dá el resultado esperado de una inversión hecha.

Los sistemas pueden ser clasificados en dos tipos: servidores y clientes. Los sistemas servidores son computadoras o procesos que proveen de servicios a otros sistemas llamados clientes y están conectados mediante una red de trabajo, los sistemas cliente son aquellas computadoras o procesos que usan los servicios que proveen los sistemas servidores.

En el ambiente mencionado es admisible que un sistema cliente tenga una falla ya sea pequeña o total, pues bastará con reemplazar el equipo sin que esto afecte a algún otro usuario, en un sistema servidor la situación es muy diferente pues de éste depende el trabajo de todo aquel que requiera su servicio, si ocurre una falla impactará directamente en el trabajo de varios usuarios.

Los equipos de cómputo siempre estarán susceptibles a sufrir alguna falla, es posible diferenciar el grado en que afecta en los ambientes de producción y desarrollo de una organización; en el ambiente de producción una falla afectará directamente a usuarios externos así como en la imagen externa lo cual se traducirá en pérdidas costosas, mientras que para el ambiente de desarrollo no es así, éste está representado por usuarios internos, programadores quienes son responsables de cambios al sistema y que una vez probados estarán en producción.

Para ejemplificar lo anterior tomemos el esquema de una empresa financiera; su ambiente de producción está representado por los servicios que proporcionan todas y cada una de sus sucursales y cajeros, la falla de alguno de estos representará pérdidas considerables pues las personas optarán por cambiar a un banco el cual tenga sus servicios disponibles cuando lo requieran, en un ambiente de desarrollo no es tan crítico pues una falla en el caso más extremo implicará el retraso de la puesta en marcha de algún cambio en producción, el cual no afectará las disponibilidad en los servicios que esta empresa proporciona externamente.

Normalmente los usuarios o líderes de proyecto dirán el 100 por ciento de disponibilidad es requerido debido a que su proyecto es muy importante y no se puede permitir en lo absoluto tiempos improductivos debido a fallas del sistema, a menudo se acaba por cambiar de parecer cuando se obtiene el costo de lo que implica tener este nivel de disponibilidad y se convierte en un asunto monetario y de un proceso de negociación.

Como se puede observar en la tabla 2.1[2], para muchas aplicaciones el 99 por ciento de disponibilidad es adecuado, si los sistemas promedian una hora y media de tiempo inactivo semanal puede ser tolerable. Claro que mucho de eso depende cuando esa hora y media ocurra; si la falla es entre 3:00 y 4:30 de la madrugada de un domingo, puede ser admitido ya que no afecta tanto como si ocurre entre 10:00 y 11:30 en cualquier día hábil de la semana.

98	2	7.3 días	3 horas, 22 minutos
99	1	3.65 días	1 hora, 41 minutos
99.8	0.2	17 horas, 30 minutos	20 minutos, 10 segundos
99.9	0.1	8 horas, 45 minutos	10 minutos, 5 segundos
99.99	0.01	52.5 minutos	1 minuto
99.999	0.001	5.25 minutos	6 segundos
99.9999	0.0001	31.5 segundos	0.6 segundos

Tabla 2.1
Porcentajes de disponibilidad de los sistemas

En la tabla 2.1 se representa el porcentaje de la disponibilidad de los sistemas, entre más disponibilidad es lograda se reducen más los tiempos de inactividad.

Un punto de negociación en cuanto a la disponibilidad pueden ser las horas que durante ese 100 por ciento es requerido, es decir, que una organización puede requerirlo sólo durante su tiempo de producción y no el resto del tiempo, haciendo que la meta de disponibilidad pueda ser lograda de una manera más fácil en un esquema 6 (días) x 12 (horas) x 317 (días), la situación se complica cuando se ofrecen servicios por Internet y algún día pueda ser necesario en caso de convertirse una empresa con sistemas de misión critica los cuales deberán

[2] Tabla tomada del libro Blueprints for high availability de Evan Marcus y Hal Stern Editorial Wiley.

tener una alta disponibilidad y será imperante tener el esquema de 7 (días) x 24 (horas) x 365 (días).

El objetivo de los sistemas será alcanzar lo más cercano al 100 % de disponibilidad lo cual representará un desembolso económico considerable y una vez hecho el cambio de visión de los directivos de una organización de que cubrir el costo de mantener disponibles a los sistemas es una inversión que se reflejará en menos caídas del sistema debido a fallas, si bien el 100 % de disponibilidad es un objetivo que en la actualidad es irreal, considerar la inversión para mantenerse arriba del 99% es algo más admisible y al alcance para las organizaciones. Tanto se va progresivamente moviendo a altos niveles de disponibilidad el costo de la implementación se incrementa rápidamente, pues para alcanzar este esquema la inversión se debe hacer a nivel sistema y recursos humanos, el incremento en costos no es lineal pues el moverse de un nivel de disponibilidad a otro mas alto costará de cinco a diez veces más respecto al anterior nivel que se tenga.

Con base en la tabla 2.1 se puede cuestionar que es lo que pasa con el número de nueves en el porcentaje para lograr una alta disponibilidad, se espera obtener lo ideal en cuanto a lo alcanzable que son los mencionados cinco nueves (99.999%) con seis minutos de tiempo abajo anual y esperar la continuidad de la disponibilidad.

Hay dos razones que son comentadas por Veritas que es una empresa que se dedica a hacer software de alta disponibilidad y que cuestionan la generalidad de asociar niveles numéricos de disponibilidad con costos específicos:

La primera es que la disponibilidad es algo más que los "nueves", es cuando un fallo total ocurre por ejemplo en un centro de cómputo de comercio electrónico en Diciembre es muy diferente a que si ocurre lo mismo en Agosto, es decir, será la percepción de disponibilidad lo que importa, de cuando es más requerido un sistema.

La segunda razón tiene que ver con algunos vendedores los cuales ofrecen garantías sin tomar en cuenta con que software y/o hardware interactuarán o si ocurrirá un fallo externo al sistema como alguna falla eléctrica, son pocas las empresas que ofrecen la información como alguna matriz de compatibilidad y hay las veces que no se percibe esto hasta que se hace la petición de probar estos elementos y su correcto funcionamiento y documentarlo así como también hasta el momento no existe sistema de cómputo ni vendedor que ofrezca una garantía de desempeño y disponibilidad que incluyan fallas eléctricas.

Es preferible considerar a la continuidad de la disponibilidad como una línea de orientación, si se hacen las cosas correctas en cuanto a los sistemas, es posible lograr altos niveles de disponibilidad, pero el obtener numéricamente los porcentajes de una forma rígida será algo fuera de la realidad.

2.2 DEFINICIÓN DE TIEMPO FUERA EN EQUIPOS DE PRODUCCIÓN (DOWNTIME) Y SUS CAUSAS.

El tiempo fuera es determinado por la falla de componentes de un sistema como son: el servidor, discos, la red, el sistema operativo o alguna aplicación, y obviamente la inactividad total, definiciones estrictas incluyen la lentitud del servidor, el desempeño de la red, la incapacidad de recuperar datos o la inaccesibilidad hacia estos.

Un usuario puede ver el sistema abajo cuando éste presenta lentitud, pues aún cuando está activo, su trabajo o la obtención del servicio no puede ser hecho con la eficiencia debida, lo opuesto ocurre con el administrador del sistema o la organización misma que de servicios a través de los sistemas, quienes dirán que el sistema está disponible y en ningún momento estuvo abajo y lo cual será reflejado en estadísticas de servicios que darán a conocer las horas de actividad del sistema pero no su lentitud. Cuando por cualquier circunstancia debida al sistema un usuario no puede llevar a cabo su trabajo su trabajo se considera que el sistema está en tiempo fuera y ésta será la definición que se tomará en el presente trabajo.

Entre las causas de tiempo fuera, la que más porcentaje abarca son los tiempos fuera planeados, estos tiempos son eventos que normalmente son por la noche cuando los administradores del sistema adicionan hardware, actualizan el sistema operativo u otro software crítico, a veces es solo un mantenimiento preventivo, un reinicio del sistema para limpiar las bitácoras, directorios temporales y memoria.

Varias de estas actividades en sistemas actuales pueden ser hechas cuando el sistema está activo; los discos pueden ser removidos o adicionados, muchas aplicaciones críticas pueden ser actualizadas e incluso en un ambiente redundante[3] el servidor primario puede ser actualizado mientras el secundario toma su lugar y la única interrupción es cuando se regresa el control al servidor que se actualizó, actualmente hay proveedores que producen arreglos de discos de almacenamiento lógico y software de manejo de volúmenes para no interrumpir en absoluto el servicio que éstos proporcionen.

El otro factor que genera tiempos fuera son causados por los usuarios y lo generan por dos razones: una es que normalmente tienen errores debido al descuido y la otra es por que no tienen bien definida la forma en cómo el sistema que usan funciona. La mejor forma de combatir este factor, es entrenando a los usuarios de aplicaciones y servicios de una forma que el sistema se les presente de una manera simple a modo de evitar errores por alguna incomprensión y mantener una documentación para ellos y estén actualizados en cuanto a términos y funcionamiento.

---

[3] La redundancia o un ambiente redundante es aquel en el que un sistema (primario) cuanta con otro igual (secundario) que contiene el mismo número y tipo de componentes y el cual se activará en caso de alguna falla detectada, este esquema de disponibilidad se verá más a detalle en el capítulo 3.

Como es mostrado en la figura 2.1 la más sorprendente causa es la de hardware con 10%, esto significa que los mejores discos y las redes más redundantes sólo pueden prevenir este porcentaje de tiempo fuera puesto que en estas fallas se incluyen las de procesador, memoria, fuentes de poder y el sistema interno de enfriado, aunque lo que más tiende a fallar son los discos duros.

La causa más obvia está en las fallas de software que son responsables de un 40%, los errores en los módulos de código fuente del software llamados bugs, son la fuente más difícil de controlar y que predominan como causante de tiempo fuera de un sistema. Cuando el hardware sea más confiable, los métodos de tiempo fuera planeados sean más confiables y las técnicas de refinamiento de código (debugging) sean más sofisticadas el porcentaje de fallas se reducirá considerablemente.

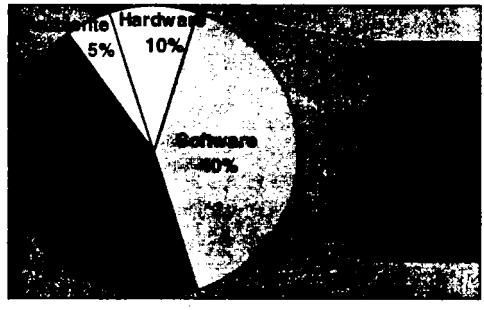


Figura 2.1
Porcentaje de causas de tiempo fuera

La más pequeña causa de fallas son las ambientales, que de alguna manera llegan a ser más controlables, están representados por todo aquello que envuelve la atmósfera de un centro de cómputo: cableado eléctrico, cableado de red, sistema de aire acondicionado, piso falso, sistemas de energía eléctrica de respaldo y personal de limpieza, este último a diferencia de los demás elementos es normalmente un factor de descuido típico pues hay componentes a los cuales se debe tener especial cuidado al limpiar y muchas veces accidentalmente desconectan de la energía eléctrica a los sistemas o crean ruido que afectan el funcionamiento de las computadoras al conectar aspiradoras.

Actualmente hay empresas que llevan a cabo mantenimiento preventivo y correctivo de la infraestructura de un centro de cómputo y que hacen posible minimizar los puntos de falla que rodean físicamente a los sistemas, por otro lado se necesita implementar políticas para los empleados de limpieza y hacer de su conocimiento los cuidados que se deben llevar a cabo cuando hagan su trabajo dentro del centro de cómputo.

2.3 TIPOS DE FALLAS EN LOS SISTEMAS DE CÓMPUTO

A continuación veremos los tipos de fallas, lo que puede estar mal en un sistema de cómputo y que son causantes de tiempos fuera. Algunos de ellos, en especial el de hardware, pueden parecer increíblemente obvios pero otros no.

2.3.1 Fallas de hardware.

Los puntos de falla en hardware son los más obvios, son las fallas que la gente de sistemas piensa en primer instancia cuando se les pregunta acerca de éstas. Como se vio anteriormente son los que hacen el 10% de causa de inactividad del sistema. Es posible que cuando se tiene esta falla la inactividad pueda prolongarse en tiempo si no se cuenta con un ambiente redundante pues de ser necesario cambiar alguna parte no se tendrá disponible en el momento y por lo tanto habrá que esperar a que llegue.

Los componentes que con más frecuencia causan fallas son las partes movibles, especialmente aquellas asociadas a la alta velocidad, baja tolerancia y complejidad, siendo los discos los primeros candidatos. Estos discos tienen también tarjetas controladoras y cableado que pueden romperse o fallar, hay arreglos de discos los cuales tienen componentes adicionales propensos a fallas, tales como memoria para caché o hardware para espejo o striping[4] (segmentación).

Dispositivos de cinta, en especial los DLT[5], tienen muchas partes movibles, motores que arrancan y se detienen y son extremadamente bajos en tolerancia a fallas y por ende dejan de funcionar.

Los ventiladores internos son otro componente con partes movibles y las fallas de un ventilador tal vez no causen un efecto inmediato en el sistema como ocurre con los discos duros, pero cuando el ventilador de una máquina falla los efectos son impredecibles. Cuando el procesador y la memoria se sobrecalientan los sistemas pueden fallar de diferentes maneras, muchos sistemas no tienen un monitoreo de su sistema de enfriado siendo estos un punto de falla que normalmente se da por sorpresa.

Las fuentes de poder pueden resultar también puntos de falla potenciales causando la baja del sistema o un deterioro gradual siendo éste un desagradable causante de fallas en procesador, memoria o la tarjeta madre completa, las fallas de la fuente de poder son causadas por varios factores incluyendo la variación de voltaje y el acentuado cambio de estados entre apagado y encendido.

---

[4] El striping o segmentación es un método de la tecnología RAID 0 la cual consiste en almacenamiento y escritura de datos en paralelo, este método y la tecnología RAOD se tratarán más a detalle en el capítulo 3.
[5] DLT es un tipo especial de unidad manejadora de cinta de alta capacidad desde los 70 a los 120 GB.

2.3.2 Fallas físicas y ambientales.

Las fallas de un sistema pueden ser tanto internas como externas, hay varios componentes ambientales que pueden causar la baja de un sistema aún cuando son considerados rara vez como puntos de falla potenciales. Muchos de estos están relacionados con centros de cómputo pero varios de éstos pueden impactar en otros servidores según su ubicación y en varios casos tener un servidor de respaldo no es suficiente en estas situaciones.

El problema ambiental que más frecuentemente se presenta es la falla de energía eléctrica debido a las fuentes de poder de respaldo o algún fusible que pueden causar la baja del sistema, la desconexión debido a la limpieza del lugar por conectar una aspiradora e inclusive sobrecargar un circuito debido a su encendido.

Un sistema de aire acondicionado puede fallar causando un masivo sobrecalentamiento en los otros sistemas que se encuentren en el mismo cuarto.

Muchos de los centros de cómputo tienen varios cables debajo de un piso falso y en las cercanías de un concentrador, el rompimiento de uno de éstos puede causar una gran falla sobre todo los que conducen energía eléctrica a los sistemas.

Muchos de los centros de cómputo tienen sistemas de protección contra el fuego, como por ejemplo, el gas halon, que a consecuencia de un incendio es primero una ayuda y después resulta ser un factor de falla de los sistemas, lo mismo pasa con un sistema contra incendio a base de agua, con el paso del tiempo estos sistemas han mejorado para no ser un factor de falla de los sistemas, controlando la aspersión en donde solo hay fuego y una vez extinguido se detiene el riego, el otro polo seria no tener estos sistemas la buena noticia seria que no habría residuos de humedad y la mala es que el alguna vez un centro de cómputo podría ser solo un cuarto de residuos de sistemas incendiados en el peor de los casos.

Otro problema potencial de ambiente es la falla estructural debido a un componente de soporte de una cabina o estante de computadora que pueden venirse abajo cuando no están bien construidos o ensamblados, si el estante no está bien asegurado en la cabina se puede afectar a los que están debajo de éste.

2.3.3 Fallas de red.

Las redes son normalmente susceptibles a fallas debido a que tienen varios componentes que son afectados por la configuración de cada uno, se necesita tener la correcta información en el momento de hacer algún cambio para evitar la interrupción debido a la desconexión del cable de red incorrecto.

Los problemas de configuración normalmente se deben a información incorrecta de ruteo, máscara, nombres y direcciones de red incorrectos o duplicados, todo esto causando que una red pueda venirse abajo.

Cuando se trata de conectar una red confiable a una red sin manejo e inconfiable, se corre el riesgo ser sujeto de una negación del servicio o hasta un ataque de la propia red.

Finalmente, las redes usan una variedad de servicios básicos, sistemas de nombres como NIS[6] o DNS[7], servidores de autenticación y seguridad, o servidores de configuración requiriendo de DHCP[8] para levantar el sistema y que por una mala configuración serán destinados a ser un punto de falla.

2.3.4 Fallas de sistemas de base de datos.

Tal como cualquier aplicación compleja, los sistemas de base de datos tienen varios componentes; el corazón de un sistema de base de datos es el proceso servidor o el motor de la base de datos quien es el principal componente de la encargado de leer y escribir en el disco, manejando la ubicación de los datos y respondiendo a sentencias que requieren de información, si este proceso se detiene todo usuario que accesa al sistema no podrá trabajar con la base de datos.

También entre los usuarios y el servidor de base de datos se encuentra el proceso de escucha llamado listener, éste toma las sentencias provenientes del usuario y los cambia a una forma que el servidor de base de datos pueda procesar y entonces, cuando el servidor responde, el listener envía la respuesta al usuario que hizo la petición.

Los usuarios en sus estaciones de trabajo corren sus aplicaciones de usuario final la cual para el caso de base de datos es casi siempre es usado el SQL[9]. La aplicación de usuario final traduce la petición del usuario en sentencias de SQL las cuales son enviadas a través de la red al proceso de listener.

La falla de alguno de los componentes en la cadena del funcionamiento de la base de datos causará la indisponibilidad para los usuarios. Posibles fallas pueden incluir:

Aplicación caída. La aplicación se detiene completamente, dejando un mensaje de error que habilita al administrador del sistema determinar la naturaleza del problema.

Aplicación colgada. El problema más incidente de bases de datos y otros sistemas que tienen una interacción significante con el sistema operativo, es

---

[6] NIS Network Information Service por sus siglas en inglés es un servicio de nombres distribuido para identificar y localizar objetos y recursos de la red.
[7] DNS Domain Name Service por sus siglas en inglés es un servicio de nombres de dominio el cual traduce nombre de computadoras en su correspondiente dirección IP y especifica el dominio al que pertenece.
[8] DHCP Dinamic Host Configuration Protocol por sus siglas en inglés es un protocolo usado para la asignación dinámica de direcciones IP.
[9] SQL Estándar Query Language por sus siglas en inglés es el lenguaje estándar usado en bases de datos.

cuando un proceso componente tal como el listener, el proceso lector-escritor o el kernel del sistema operativo se cuelga esperando por el recurso del sistema que sea liberado por otro proceso.

Déficit de recursos. El más común déficit de recurso en el ambiente de bases de datos, es el espacio en disco asignado, cuando éste se llena causa que se detenga sin aceptar sesión alguna ni peticiones, los factores causantes son las bitácoras generadas por la aplicación y las mismas actualizaciones de datos que gradualmente ocupan espacio y degradan la respuesta del proceso servidor.

Corrupción de índices de base de datos. Un servidor de base de datos puede manejar terabytes de información, para encontrar datos de manera rápida en los discos se usa un arreglo complejo de punteros y ligas pero estos pueden convertirse en corruptos, los datos pueden ser erróneos o pero aún, el tratar de obtener datos de un espacio ilegal puede provocar que la aplicación se corrompa completamente inactivándose. El problema de corrupción de datos son algo inusual debido a que muy buenos RDBMS[10] tienen verificadores de consistencia, los cuales revisan la base de datos cuando ésta levanta para estar activa.

Errores de software (bugs). Se asegura que si en los programas extensos se tomára un módulo normalmente habría una línea con error, es decir, un bug. El software es hecho por humanos y muchos de nosotros de vez en cuando cometemos errores. Los bugs pueden impactar un sistema de varias maneras desde una simple escritura incorrecta en una bitácora hasta uno fatal que causa la baja de la aplicación o del mismo sistema.

2.3.5 Fallas de servidor Web.

Varios servidores web son parte de aplicaciones cliente-servidor que hacen uso de los servidores de base de datos para dar servicio a peticiones de clientes mediante páginas web así que algo que afecte el servidor de base de datos directamente afecta al servidor web. Hay también otras fuentes de bugs incluyendo CGI[11], Perl, Java, o Active/X que es código que manejan las páginas web, si por alguna circunstancia un programa CGI se cicla, la página web puede nunca desplegarse y causará que el usuario elija otro sitio.

Un servidor web implica una colección de aplicaciones: el httpd o servidor web que maneja las peticiones en una página web y responde con archivo HTML o archivos de imagen, los CGI ejecutados para generar las páginas web para los clientes y una base de datos o servidores de archivos que contienen los datos usados en el sitio web. La falla de cualquiera de estos componentes resulta en una falla del servidor web.

---

[10] Relational Data Base Management System por sus siglas en inglés es el sistema manejador o gestionador de una base de datos relacional.

[11] CGI Common Gateway Interface por sus siglas en inglés

2.4 RELACIÓN COSTO-RIESGO

La única manera de convencer a la gente que maneja los recursos monetarios en una organización, de dar valor a los sistemas y datos, es el tener y mostrar la perspectiva de valuar el costo real de lo que se está hablando.

El más obvio costo de inactividad de un sistema es probablemente la que menos valor tiene, esto es que se pierda la productividad del usuario, el actual costo de ésto es dependiente del trabajo que dejen de hacer los usuarios mientras un sistema se vea afectado por alguna falla.

Si los usuarios son desarrolladores entonces, tal vez, el costo se verá en sólo el tiempo y el acarrear con el costo de aquellos que estén desocupados, desde luego para una organización grande de programadores o usuarios finales los costos de inactividad serán bastante significativos.

Teniendo como ejemplo una organización la cual da servicio de ventas por Internet, una falla, la que sea que provoque que un usuario conectado no pueda obtener un servicio para adquirir algún articulo, muy probablemente afecte al sistema proveedor de servicio un tiempo considerablemente mayor al que pudiese tener en el caso de haber invertido en algún esquema de alta disponibilidad, no se consideró en los costos desembolsar para obtener los elementos que aseguren el menor tiempo posible de tiempo abajo del sistema lo cual de manera inmediata se considere tal vez como un ahorro, ésta consideración deja de ser aceptada cuando ocurre la falla expuesta y la perspectiva cambiará de un gasto a una inversión.

El usuario que demanda el servicio y no le es proporcionado de la manera y tiempo deseado obviamente se ira a otro sitio el cual ofrezca de manera efectiva lo que busca, teniendo este esquema tal vez no represente un caso de pérdida considerable más sin embargo, este usuario al ser consultado por otros para recomendar sitios será claro que no lo hará por el sitio el cual no le pudo ofrecer el servicio, aunado a esto, en el momento de la falla no sólo pudiera haber sido un usuario el que quiso acceder sino cientos dando como resultado una gran pérdida en ventas para la organización proveedora del servicio de ventas de artículos por Internet.

Otro ejemplo puede ser el que en una junta de directivos dispuestos a mantener lazos estrechos de comercio tales como unión de servicios e incluso de las mismas empresas, estos requieran de información de la más actualizada para llevar a cabo las negociaciones y tomar la más adecuada toma de decisiones en ese momento, será importante en esos instantes tener las información, cual requerida que esta sea y resida en los sistemas, disponible y sin ningún contratiempo más que el que se lleva en instrucciones para obtenerla y darle el formato correspondiente. El tiempo que se consuma en la obtención de los datos será crítico dentro de los momentos de negociación, el tomarse mucho tiempo debido a alguna falla del sistema puede ser algo que impacte seriamente el

proceso de negociaciones que se pueden venir abajo debido a que no se contó con la información a tiempo para llegar a buenos términos convenientes a nivel directivo.
Si bien el costo de obtener un sistema con el más bajo tiempo de inactividad posible es costoso y a primera vista no es posible considerar pérdidas en caso de alguna falla, tampoco es permisible actualmente el no tener una perspectiva de inversión y esperar hasta que ocurra una falla que se traduzca en degradación del servicio que se proporcione tanto como de la misma organización y por ende corte de personal debido a las pérdidas representadas por fallas que dieron como resultado la baja recomendación de un servicio ofrecido o la pérdida de negociaciones.

Tomar el riesgo de no tener la visión de invertir en los sistemas para aminorar en lo posible el tiempo fuera debido a fallas llegará a ser bastante alto en costos e impactará en la sobrevivencia de una organización. El riesgo es determinado por los eventos que tendrán efectos, en nuestro caso son las fallas, cada evento o falla tendrá tres factores a considerar: probabilidad, duración e impacto.

Probabilidad. Será el número de veces de un fallo considerado como esperado en el tiempo restante de la vida de un sistema, puede haber desde un fallo muy frecuente hasta uno muy raro como un desastre total, en estos casos es recomendable examinar el historial del sistema, hablar y mantener contacto con gente que maneje los mismos sistemas y usar la propia experiencia.

Duración. Es el periodo de tiempo que el usuario se verá imposibilitado a trabajar debido a al indisponibilidad del sistema más el tiempo que se tome para tener disponible al sistema, puede ser desde recuperarse de un fallo de hardware y cambiar la parte afectada hasta la corrupción de un sistema de archivos el cual será necesario restaurar desde un respaldo.

Impacto. Estará representado por el número de usuarios que se afectó debido a la indisponibilidad del sistema causado por algún fallo, éste puede variar por la hora en la que el sistema esté indisponible, será un gran número si es en horas de oficina y será bajo si es en horas y días inhábiles.

Los factores a considerar en los costos por tiempos fuera son los siguientes:

- Acarreo de costos por empleados desocupados
- Retraso de proyectos
- Insatisfacción de clientes
- Mala publicidad
- Pérdida del negocio
- Pérdida clientes a corto y largo plazo
- Pérdida de competitividad
- Costos de pérdida de oportunidades
- Penalizaciones

2.4.1 Costos de tiempo fuera por industria.

En la tabla 2.2[12] puede verse el costo por hora de los tiempos fuera en diferentes industrias y su promedio, estos costos son dependientes de la tecnología usada y de los datos, la vulnerabilidad de una organización en cuanto a la indisponibilidad y/o pérdida de datos no sólo es de impacto monetario, se incluyen situaciones como pérdida de clientes, de confiabilidad y posiblemente hasta del negocio.

Industria	Costo del tiempo fuera por hora (dólares)
Operaciones de corredores de bolsa	$6,450,000.00
Energía	$2,817,846.00
Autorizaciones en ventas por tarjetas de crédito	$2,600,000.00
Telecomunicaciones	$2,066,245.00
Manufactura	$1,610,654.00
Instituciones financieras	$1,495,134.00
Tecnología de información	$1,344,461.00
Aseguradoras	$1,202,444.00
Ventas de menudeo	$1,707,274.00
Farmacéuticos	$1,082,252.00
Bancos	$996,802.00
Procesamiento de alimentos y bebidas	$804,192.00
Productos al consumidor	$785,719.00
Química	$704,101.00
Transportación	$668,586.00
Servicio público	$643,250.00
Sector salud	$636,030.00
Recursos naturales y metalúrgicos	$580,588.00
Servicios profesionales	$532,510.00
Electrónica	$477,366.00
Contrucción e ingeniería	$389,601.00
Medios de información	$340,432.00
Viajes y hospedaje	$330,654.00
Televisión de pago por evento	$150,000.00
Ventas por televisión	$113,000.00
Ventas por catalogo	$90,000.00
Reservaciones de vuelo	$90,000.00

[12]Fuente: IT Performance Engineering and Measurement Strategies: Quantifying Performance and Loss, Meta Group, Oct. 2000; Fibre Channel Industry Association.

Industria	Costo del tiempo fuera por hora (dólares)
Ventas por tele-ticket	$69,000.00
Envío de paquetes	$28,000.00
Pagos en punto de venta	$14,500.00
Promedio	**$944,395.00**

Tabla 2.2
Pérdidas en dólares por industria

2.5 NIVELES DE DISPONIBILIDAD.

Los niveles de disponibilidad que a continuación se presentan no son de ninguna manera discretos entre cada uno de ellos, puede haber un buen número de pasos incrementales y combinaciones de tecnologías que se puedan adicionar o en algunos casos decrementar la disponibilidad general de un sistema. Los siguientes niveles son arbitrarios pero representan escenarios reales que van desde el nivel más básico hasta el más completo.

Nivel 1. Disponibilidad regular:

Es el nivel más básico de protección del sistema, no hay medidas especiales a tomar, en este nivel se cuenta sólo con respaldos de datos sin haber planes de contingencia si el sistema se viene abajo. Si se tiene una falla del disco al final del día se habrá perdido todo el trabajo hecho pues el sistema será recuperado con un respaldo del día anterior, este nivel muy probablemente no satisfaga las necesidades de las empresas con grandes masas de datos y transacciones.

Nivel 2. Disponibilidad incrementada:

El nivel 2 tiene la diferencia con el anterior nivel, en algunas medidas de protección de datos, esto significa emplear tecnología RAID[13] y entonces la falla de un disco no será pérdida de datos debido a que la información es almacenada en más de un disco físico, tal vez haya la inactividad del sistema pero con el refuerzo de los respaldos el más critico recurso del sistema que son los datos estarán protegidos por algún esquema implementadode RAID en lso discos duros.

Nivel 3. Alta disponibilidad:

Este nivel es normalmente llamado HA[14] y en esta configuración se tomarán 2 servidores que junto con sus componentes formarán un conjunto actuando como un solo sistema (cluster) y combinado con discos protegidos por RAID, el sistema tendrá componentes duplicados si uno falla automática o manualmente serán activados para reemplazar el que esté fallando.

Aún en estos sistemas existe tiempo fuera debido a la inactividad, pero en muchos casos este tiempo es limitado en este nivel puede ser alcanzada la disponibilidad de un 99.98% e incrementarlo implementando protecciones adicionales en otros puntos de falla o SPOF[15].

---

[13]Redundant Array of Inexpensive (or Independent Disks por sus siglas en inglés, es una tecnología usada para arreglos de discos y hacer que tengan un mejor desempeño y sean menos propensos a fallas, este tema es tratado más a detalle en el capítulo 3.

[14]High Availability por sus siglas en inglés es el término empleado para referirse a la alta disponibilidad.

[15]Single Point of Failure por sus siglas en inglés es el término usado para referirse a un punto de falla.

El implementar HA requiere de un costo considerable debido a que cada componente tendrá su redundante lo cual resultará tener dos sistemas y sólo uno de ellos será el activo, mientras el otro estará a la espera de la señal de una falla y por mientras improductivo.

Nivel 4. Recuperación en desastres

Éste es el nivel de disponibilidad más alto y caro de protección de un sistema, cuando este esquema es implementado se estará protegiendo en contra de la pérdida total del sitio de operaciones teniendo un sitio alterno a distancia del original y éste necesitará tener todo lo necesario para operar de igual forma en caso de desastre.

Previamente se habrá probado y tomado tiempos mediante un simulacro de recuperación de desastres con la ayuda de los respaldos de datos hechos en el sitio original y restaurándolos en los sistemas del sitio alterno y probando que todo el sistema funcione como el siniestrado.

Como se comentó al principio de este capítulo los pasos entre uno y otro nivel de disponibilidad hay pasos incrementales y combinaciones de tecnologías así como también el incremento de la inversión para lograrlo, en la figura 2.2 muestra el índice de disponibilidad de datos de Veritas[16]

Los niveles presentados de abajo hacia arriba son:

- Buenas prácticas de administración e infraestructura
- Respaldos grupales y locales
- Manejo flexible de archivos y volúmenes
- Respaldos empresariales
- Manejo de SAN de hardware independiente
- Manejo de recursos compartidos
- Manejo de bases de datos
- Manejo avanzado de bases de datos
- Cluster local
- Cluster metropolitano
- Replicación de sitio a sitio
- Cluster WAN

Los ejes horizontal y vertical están representados por la inversión y la disponibilidad respectivamente.

[16]Tomado del documento The value of availability de Evan Marcus

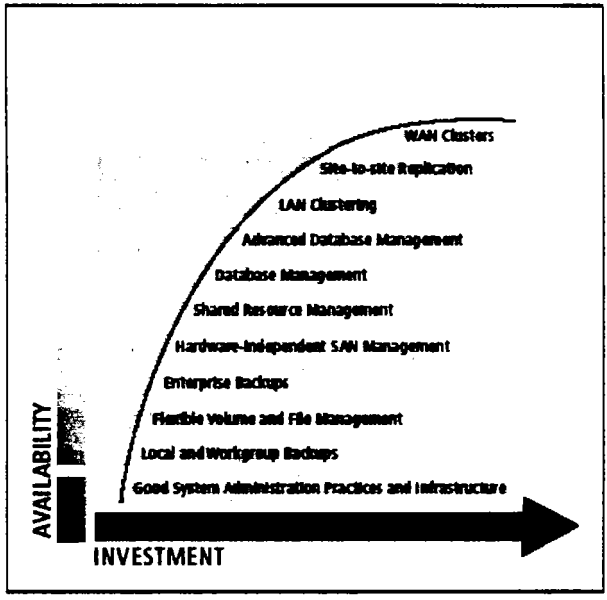


Figura 2.2
El índice de disponibilidad de Veritas

En el próximo capítulo se verán las técnicas que se utilizan en cada nivel de disponibilidad a detalle y que hacen posible el decremento de tiempo fuera causado por fallas.

En este capítulo se ha visto cómo es posible medir con ciertos parámetros la disponibilidad de los sistemas, definir el tiempo fuera desde el punto de vista de usuario y cuales son las causas, los principales tipos de fallas en ambientes reales así como también el riesgo que se corre cuando no se consideran los costos de pérdida por no invertir en esquemas de alta disponibilidad y por último cuatro niveles de disponibilidad de los sistemas los cuales van desde el más básico y menos costoso hasta el más completo y caro de implementar.

3. TÉCNICAS UTILIZADAS PARA EL MANEJO DE LA DISPONIBILIDAD

En este capítulo trataremos las técnicas que actualmente se emplean en el manejo de la disponibilidad de los sistemas de cómputo, comenzaremos desde la más básica y cerraremos con la más completa, las primeras dos que son los respaldos y RAID se enfatizan en los datos y serán básicas para las siguientes técnicas que mejoran y complementan con el fin de mantener la continuidad de las operaciones; estas serán la tolerancia a fallas, la replicación de datos y la recuperación de desastres siendo esta última la mejor y la más costosa.

3.1 RESPALDOS

Esta técnica consiste en almacenar los datos en otro medio que normalmente son cintas magnéticas o discos compactos y son hechos a través de hardware manejador de la media correspondiente y herramientas de software que hacen posible ésta tarea en la que se respaldará a los datos, son hechos de una manera periódica según sean las necesidades de una organización; pueden ser desde diarias hasta semanales, incrementales o completas.

Como se mencionó en el capítulo anterior, el primer nivel de disponibilidad consiste en tener respaldos del sistema los cuales serán requeridos en algún caso de recuperación de datos y/o aplicaciones. El tener un nivel de alta disponibilidad implementado no quiere decir que los respaldos estén de sobra y no sean contemplados, los datos no dejan de cambiar debido al procesamiento que a éstos se les da por parte de los usuarios y el día menos pensado requieren tener los datos de cualquier otro día para una verificación, auditoria o algún otro uso, el cual sea parte de su trabajo o también puede ocurrir algo peor como un desastre el cual tenga tal impacto que deje en su totalidad inservibles a los sistemas, tal vez tome tiempo el adquirir equipo para volver a trabajar pero la organización deseará hacerlo desde el punto más cercano en que ocurrió el desastre y para hacerlo así, se necesita de los datos respaldados, se puede ver entonces que los respaldos juegan siempre un complemento importante aún cuando se tenga un sistema de alta disponibilidad.

El componente más crítico a respaldar en un sistema son los datos, sobre todo aquellos que cambian constantemente, algunos puntos importantes en cuanto a respaldos se dan a continuación:

Los sistemas en espejo no reemplazan a los respaldos. El espejeo protege a los datos en contra de una falla de hardware del almacenamiento, pero no protege en el borrado o corrupción de un archivo, pues de ocurrir esto en cualquiera de los lados del espejo el archivo será afectado de los dos lados y se requerirá invariablemente de recuperarlo de un respaldo.

El uso más común de recuperación es debido al usuario. Normalmente un usuario accidentalmente borra o modifica archivos y/o directorios, hay que tener listos siempre los respaldos para recuperar desde un simple archivo hasta varios directorios.

Verificar regularmente los respaldos para su uso en recuperaciones. Los respaldos son el último recurso y el más preciado, pero si estos no son verificados revisando su contenido se habrá perdido tiempo en crear cintas inútiles, esto no quiere decir que se tenga que verificar toda una cinta, se puede probar aleatoriamente, también se puede crear una bitácora en un archivo de lo respaldado para tener la seguridad de que los datos están en la cinta empleada.

Mantener las cabezas del manejador de cinta limpias. Las cabezas sucias pueden causar que se grabe en cinta sólo basura y no los datos que se necesitan, también se necesita de limpieza cuando se presenta algún mensaje de error de lectura/escritura, esto es posible usando una cinta de limpieza, también es posible hacer un mantenimiento preventivo desarmando la unidad de cinta y remover el polvo acumulado.

Mantener las cintas en un lugar seguro y limpio. Las cintotecas deben ser accesadas solamente por los administradores del sistema y el lugar asignado a éstas debe ser limpio, lo más libre de polvo que sea posible, y alejado de fuentes de campos magnéticos que hagan que se deteriore lo respaldado o se deteriore la cinta quedando inservible para volverla a usar.

Las cintas se deterioran. No hay que asumir que las cintas que se ocuparon en un respaldo hace 5 o 6 años sirvan ahora, están expuestas a campos magnéticos que las afectan entre, otros factores ambientales a las que son expuestas debido a su uso o movimiento continuo, también hay cintas que se ciclan para volver a usar y a la larga son afectadas por el rebobinado y la cinta se dilata, se tienen que probar de manera periódica y hacer lo posible por pasar los datos a otra cinta más nueva.

Hacer dos copias de cintas críticas. Es bueno tener una copia de lo respaldado, teniendo el nivel de recuperación de desastres implementado es obligatorio en caso de la falla o pérdida de una cinta y mejor aún tener esa copia fuera del sitio normal donde se almacenen las cintas pues en caso de un desastre la recuperación sera garantizada.

Tener un plan de respaldos. Existen ahora utilerías para hacer los respaldos en forma incremental, es decir, respaldar sólo lo que ha cambiado ahorrando tiempo y recursos del sistema que se consumen al crear respaldos completos, ya sea semanal o mensualmente puede hacerse de acuerdo al nivel de cambios que se tenga, otro problema que hay es que normalmente hay sistemas en los cuales los servicios no deben parar y para hacer normalmente un respaldo se tenían que dar de baja los servicios para hacerlo de forma correcta pues se corría el riesgo de no respaldar los archivos en uso y hasta corromperlos, actualmente existen softwares que hacen posible los respaldos de manera confiable aún cuando el sistema está activo llamándose a estos respaldos en caliente o en línea.

3.2 REDUNDANCIA

Ésta se basa en introducir elementos adicionales en el sistema, con el objeto de detectar y recuperarse de esas fallas. Estos elementos añadidos son redundantes puesto que el sistema no necesita de ellos en su funcionamiento normal. Hay que tener mucho cuidado con estos elementos introducidos porque si no se realiza la operación en forma adecuada, pueden ser ellos los que originen los problemas.

Se pueden considerar dos tipos de redundancia, la primera se conoce como redundancia estática y se basa en el enmascaramiento de las fallas (copias que funcionan independientemente), y la segunda se conoce como redundancia dinámica y se basa en la detección y recuperación de las fallas. En este último caso existe una única copia, y si falla, se emplea otro procedimiento. Estas técnicas de redundancia son semejantes para hardware y software.

3.2.1 Redundancia en hardware

Para el hardware hay dos tipos de redundancia:

- Redundancia estática. Los componentes redundantes están siempre activos. Se utilizan para ocultar o enmascarar fallas. Una de estas técnicas es la redundancia modular triple (RMT) o redundancia modular n-ésima (RMN). Consiste en tres (n) subsistemas idénticos que operan en paralelo y un circuito votador. El resultado final se elige por mayoría. Con n=3 se enmascara la falla en un componente. Con n=2 sólo se pueden detectar fallas, no enmascararlas.

- Redundancia dinámica. Los componentes redundantes se activan cuando se detecta una falla. Algunos ejemplos de redundancia dinámica son: los "checksums" en los paquetes de comunicación y los bits de paridad en los accesos a memoria.

3.2.2 Redundancia en software

Para el software, también pueden identificarse dos aproximaciones:

Redundancia estática. Consiste en desarrollar N versiones (como mínimo) independientes de cada módulo. Las N versiones se activan concurrentemente y cuando terminan se comparan sus resultados, tal y como se aprecia en la figura 3.1. A continuación se toma como resultado del conjunto, uno sobre el que haya consenso, por ejemplo, por votación. Todo esto se hace mediante un proceso guía. Es fundamental obtener la mayor diversidad posible en el diseño del software. Esto se puede conseguir haciendo que los desarrolladores los hagan equipos diferentes. El objetivo es que las fallas de una versión no tengan que ver con las fallas de las demás versiones.

El esquema anterior presenta una serie de inconvenientes entre los que podemos citar los siguientes: en primer lugar, los errores de especificación no se enmascaran, en segundo lugar, la comparación de valores numéricos real es inexacta y por último que es una solución muy costosa.

A la programación con N versiones se le conoce como redundancia estática porque la relación de cada módulo con el resto, así como su relación con el manejador (proceso guía), es estática. También se considera estática porque todos los módulos actúan aunque no se produzcan errores.

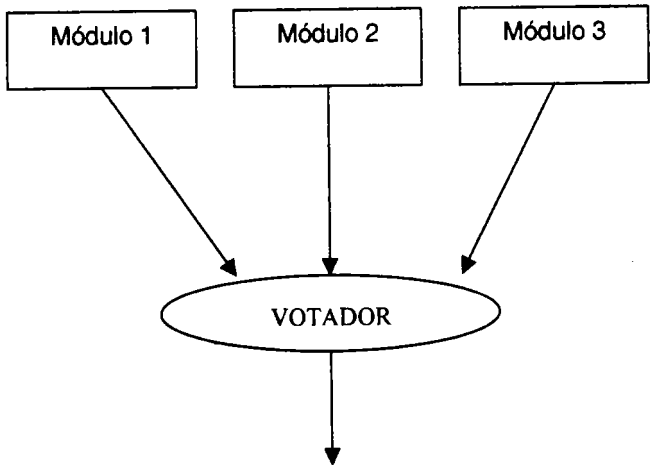


Figura 3. 1
Redundancia estática

Redundancia dinámica. En este caso y como diferencia fundamental con el caso anterior, los componentes redundantes sólo actúan cuando se producen errores, por eso esta alternativa es conocida como redundancia dinámica. Esta técnica tiene cuatro etapas:

1. Detección de fallas: Sólo cuando se detecta la falla entra en funcionamiento el módulo redundante.
2. Evaluación y confinamiento de los daños: Cuando se detecta una falla hay que decidir qué partes del sistema han quedado corruptas. Esta detección puede provocar retrasos en las tareas.
3. Recuperación de fallas: Las técnicas de recuperación de fallas pretenden llevar al sistema corrupto a un estado en el cual se puede continuar su operación normal, aunque quizás con una funcionalidad degradada.
4. Reparación de errores: Una falla es el síntoma de un error, aunque el daño puede ser reparado, el error puede seguir existiendo y por lo tanto puede reincidir.

3.2.3 RAID

La velocidad en la mejora de las prestaciones en la memoria secundaria ha sido considerablemente menor que la velocidad de mejora en procesadores y en memoria principal. Esta desigualdad ha hecho quizá, del sistema de memoria de disco, el principal foco de optimización en las prestaciones de las computadoras. Como en otras áreas de rendimiento de las computadoras, los diseñadores de memorias de disco reconocen que si uno de los componentes sólo se puede llevar a un determinado límite, se puede conseguir una ganancia en prestaciones adicionales, usando varios de esos componentes en paralelo. En el caso de la memoria de disco, esto conduce al desarrollo de conjuntos de discos que operen independientemente y en paralelo. Con varios discos, las peticiones separadas de E/S se pueden gestionar en paralelo, siempre que los datos requeridos residan en discos separados. Además, se puede ejecutar en paralelo una única petición de E/S si el bloque de datos al que se va a acceder está distribuido a lo largo de varios discos.

Con el uso de varios discos, hay una amplia variedad de formas en las que se pueden organizar los datos, y en las que se puede añadir redundancia para mejorar la seguridad. Esto podría dificultar el desarrollo de esquemas de bases de datos que se puedan usar en numerosas plataformas y sistemas operativos. Afortunadamente, la industria está de acuerdo con los esquemas estandarizados para el diseño de base de datos para discos múltiples, conocidos como RAID ("Redundant Array of Independent Disks", conjunto redundante de discos independientes). El esquema RAID consta de seis niveles[17] independientes, desde cero hasta cinco. Estos niveles no implican una relación jerárquica, pero designan diseños de arquitectura diferentes que poseen tres características comunes:

a) RAID es un conjunto de unidades físicas de discos vistas por el sistema operativo como una única unidad lógica.
b) Los datos se distribuyen a través de las unidades físicas de un conjunto.
c) La capacidad de los discos redundantes se usa para almacenar información de paridad, que garantice la recuperación de los datos en caso de falla de un disco.

Los detalles de las características segunda y tercera, cambian según los distintos niveles de RAID. RAID 0 no soporta la tercera característica. El término RAID fue originalmente ideado en un artículo de un grupo de investigación de la Universidad de California Berkeley[18]. El artículo perfilaba varias configuraciones y aplicaciones

[17] Algunos investigadores y compañías han definido niveles adicionales, pero los seis niveles descritos en esta sección son los convenidos universalmente.

[18] En ese artículo, el acrónimo RAID significaba conjunto redundante de discos baratos ("Redundant Array of Inexpensive Disks"). El término barato se usó para contrastar los discos pequeños de los conjuntos RAID, relativamente poco caros, frente a la alternativa de discos únicos, grandes y caros (SLED, "single large expensive disks"). Hoy, el SLED es esencialmente algo del pasado, con una tecnología similar a la usada tanto en configuraciones RAID como no/RAID. De acuerdo con esto, la industria ha adoptado el término independiente, para enfatizar que el conjunto RAID proporciona prestaciones significativas.

RAID e introducía las definiciones de los niveles de RAID, que todavía se usan. RAID se propuso para disminuir el aumento de la diferencia entre la velocidad del procesador y las unidades de disco electromecánicas, relativamente lentas. La estrategia es sustituir una unidad de discos de gran capacidad por varias unidades menores, y distribuir los datos de forma que se habiliten accesos simultáneos a los datos a través de distintas unidades, así, se mejoran las prestaciones de E/S y se posibilita un aumento más fácil de la capacidad.

La única contribución de la propuesta RAID es, efectivamente, hacer hincapié en la necesidad de redundancia. Aunque permitiendo que varias cabezas y actuadores operen simultáneamente se consigue mayor velocidad de E/S y de transferencia, el uso de varios dispositivos incrementa la probabilidad de falla. Para compensar esta disminución de seguridad, RAID utiliza la información de paridad almacenada, que habilita la recuperación de datos perdidos debido a la falla de un disco.

A continuación examinaremos cada nivel de RAID. En la tabla 3.1 de la pagina 54 se hace un sumario de los seis niveles. De ellos, los niveles 2 y 4 no se ofrecen comercialmente y no es probable que consigan aceptación industrial. Sin embargo, la descripción de estos niveles ayuda a clasificar las elecciones de diseño de algunos otros niveles.

La figura 3.2 de la pagina 64 ilustra los seis esquemas de RAID que mantienen los requerimientos de capacidad de datos de cuatro discos, sin redundancia. La figura remarca las zonas de datos del usuario y las de datos redundantes y señala los requerimientos relativos de almacenamiento de los distintos niveles. Nos referiremos a esta figura a lo largo de la presente discusión.

NIVEL 0 DE RAID

El nivel 0 de RAID no es un verdadero miembro de la familia de RAID, porque no incluye redundancia para mejorar las prestaciones. Sin embargo, hay algunas aplicaciones, como las de las supercomputadoras en las que las prestaciones y la capacidad son la preocupación primaria, y con un costo bajo es más importante que mejorar la seguridad.

Para el RAID 0, con todos los niveles de RAID, va más lejos que una sencilla distribución de datos a través del conjunto de discos: los datos se organizan en forma de tiras de datos a través de los discos disponibles. Esto se entiende mejor considerando la figura 3.2. Todos los datos del usuario y del sistema se ven como almacenados en un disco lógico. El disco se divide en tiras o segmentos ("stripes"); estas tiras pueden ser bloques físicos, sectores o alguna otra unidad. Las tiras se proyectan cíclicamente, en miembros consecutivos del conjunto.

Un conjunto de tiras lógicamente consecutivas, que proyectan exactamente, una tira en cada miembro del conjunto, se denomina franja. En un conjunto de n discos, las primeras n tiras lógicas (una franja, "stripe") se almacenan físicamente

en las primeras tiras de cada uno de los n discos, las segundas n tiras lógicas, se distribuyen en la segunda tira de cada disco, y así sucesivamente. La ventaja de esta disposición es que si una única petición de E/S implica a varias tiras lógicas contiguas, entonces las n tiras de esta petición se pueden gestionar en paralelo, reduciendo considerablemente el tiempo de transferencia de E/S.

En la tabla 3.1 se indica como el software de gestión de un conjunto proyecta entre el espacio del disco físico y el del disco lógico. Este software se puede ejecutar tanto en el subsistema de disco como en la computadora anfitrión.

Categoría	Nivel	Descripción	Grado de E/S (Solicitado)	Grado de transferencia de datos (Lectura/Escritura)	Aplicación típica
Estructura de tiras	0	No redundante	Tiras largas; excelente	Pequeñas tiras: excelente	Aplicaciones que requieren altas prestaciones con datos no críticos
Estructura de espejo	1	Espejo	Bueno/Regular	Regular/Regular	Controladores de sistemas; archivos críticos
Acceso paralelo	2	Redundante con código Hamming	Pobre	Excelente	
	3	Bit de paridad	Pobre	Excelente	Aplicaciones con muchas E/S
Acceso independiente	4	Bloque de paridad intercalado	Excelente/Regular	Excelente/Pobre	
	5	Paridad distribuida en bloques intercalados	Excelente/Regular	Excelente/Pobre	Grado de petición alto, lectura intensiva, consulta de datos

Tabla 3.1
Niveles de RAID

RAID 0 PARA ALTA CAPACIDAD DE TRANSFERENCIA DE DATOS

Las prestaciones de cualquiera de los niveles de RAID dependen críticamente de los patrones de petición del sistema anfitrión y de la distribución de los datos. Estas emisiones pueden ser más claramente direccionadas en RAID 0, donde el impacto de la redundancia no interfiere con el análisis. Primero, consideremos el uso de RAID 0 para logra una velocidad de transferencia de datos alta. Se deben conocer dos requisitos para que las aplicaciones tengan una velocidad de transferencia alta. Primero, debe existir una capacidad de transferencia alta en todo el camino entre la memoria de la computadora anfitrión y las unidades de disco individuales. Esto incluye controladores de buses internos, buses de E/S del anfitrión, adaptadores de E/S, y buses de memoria del anfitrión.

El segundo requisito es que la aplicación debe hacer peticiones de E/S que controlen el conjunto de discos eficientemente. Este requerimiento se satisface si la petición típica es de una gran cantidad de datos lógicamente contiguos, comparados con el tamaño de una franja. En este caso, una única petición de E/S implica la transferencia paralela de datos desde varios discos, aumentando la

velocidad efectivamente de transferencia, en comparación con la de un único disco.

RAID 0 PARA ALTAS VELOCIDADES DE PETICIÓN DE E/S

En los entornos orientados a transacciones, el usuario se suele preocupar más del tiempo de respuesta que de la velocidad de transferencia. Para una petición individual de E/S de una pequeña cantidad de datos, el tiempo de E/S está dominado por el movimiento de las cabezas del disco (tiempo de búsqueda) y el movimiento del disco (latencia rotacional).
En el entorno de transacción, puede haber cientos de peticiones de E/S por segundo.
Un conjunto de discos puede proporcionar velocidades altas de ejecución de E/S, balanceando la carga de E/S a través de los distintos discos. El balanceo de la carga efectiva, se consigue solamente si hay varias peticiones de E/S pendientes. Esto implica que hay varias aplicaciones independientes o una única aplicación orientada transacción que es capaz de generar varias peticiones de E/S asíncronas. Las prestaciones también se verán influidas por el tamaño de la franja. Si la franja es relativamente grande, de forma que una única petición de E/S sólo implique un único acceso a disco, entonces las peticiones de E/S que están esperando pueden ser tratadas en paralelo, reduciendo el tiempo de espera para cada petición.

NIVEL 1 DE RAID

RAID 1 se diferencia de los niveles 2 al 5 en cómo se consigue la redundancia. En estos otros esquemas de RAID, se usan algunas formas de cálculo de paridad para introducir redundancia; en RAID 1, la redundancia se logra con el sencillo recurso de duplicar todos los datos (espejeo). Según muestra la figura 3.2, se hace una distribución de datos, como en RAID 0. Pero en este caso, cada franja lógica se proyecta en dos discos físicos separados, de forma que cada disco del conjunto tiene un disco espejo que contiene los mismos datos.

En la organización de RAID 1 hay una serie de aspectos positivos:

1. Una petición de lectura puede ser servida por cualquiera de los discos que contienen los datos perdidos, cualquiera de ellos implica un tiempo de búsqueda mínimo más la latencia rotacional.
2. Una petición de escritura requiere que las dos tiras correspondientes se actualicen, y esto se puede hacer en paralelo. Entonces, el resultado de la escritura viene determinado por la menor de las dos escrituras (es decir, la que conlleva el mayor tiempo de búsqueda más la latencia rotacional). Sin embargo, en RAID 1 no hay "penalización en la escritura". Los niveles RAID del 2 al 5 implican el uso de paridad. Por tanto, cuando se actualiza una única tira, el software de gestión del conjunto debe calcular y actualizar la tira en cuestión.

3. La recuperación tras una falla es sencilla. Cuando una unidad falla, se puede acceder a los datos desde la segunda unidad.

La principal desventaja es el costo; requiere el doble de espacio en disco lógico que puede soportar. Debido a esto, una configuración RAID 1 posiblemente está limitada a unidades que almacenan el software del sistema y los datos, y otros archivos altamente críticos. En estos casos, RAID proporciona una copia de seguridad en tiempo real de todos los datos, en forma que en caso de falla de un disco, todos los datos críticos están inmediatamente disponibles.

En un entorno orientado a transacción, RAID 1 puede conseguir altas velocidades de petición de E/S si la mayor parte de las peticiones son lecturas. En esta situación, las prestaciones RAID 1 son próximas al doble de las de RAID 0. Sin embargo, si una parte importante de las peticiones de E/S son peticiones de escritura, entonces la ganancia en prestaciones sobre RAID 0 puede no ser significativa.
RAID 1 puede también proporcionar una mejora en las prestaciones de RAID 0 en aplicaciones de transferencia intensiva de datos con un alto porcentaje de lecturas. Se produce una mejora si la aplicación puede dividir cada petición de lectura de forma que ambos miembros del disco participen.

NIVEL 2 DE RAID

Los niveles 2 y 3 de RAID usan una técnica de acceso paralelo. En un conjunto de acceso paralelo, todos los discos miembros participan en la ejecución de cada petición de E/S. Típicamente, el giro de cada unidad individual está sincronizada de forma que cada cabeza de disco está en la misma posición en cada disco en un instante dado.

Como en los otros esquemas RAID, se usa la descomposición de datos en tiras. En el caso de RAID 2 y 3, las tiras son muy pequeñas, a menudo, tan pequeñas como un único byte o palabra. Con RAID 2, el código de corrección de errores se calcula a partir de los bits de cada disco, y los bits del código se almacenan en las correspondientes posiciones de bit en varios discos de paridad. Normalmente, se usa el código Hamming, que permite corregir errores en un bit y detectar errores en dos bits.

Aunque RAID 2 requiere menos disco que RAID 1, es todavía bastante caro. El número de discos redundantes es proporcional al logaritmo del número de discos de datos. En una sola lectura, se accede a todos los discos simultáneamente. El controlador del conjunto proporciona los datos perdidos y el código de corrección de errores asociado. Si hay un error en un solo bit, el controlador lo puede reconocer y corregir instantáneamente, con lo que el tiempo de acceso a lectura no es tan lento. En una escritura sencilla, la operación de escritura debe acceder a todos los discos de datos y de paridad.

RAID 2 debería ser solamente una elección efectiva en un entorno en el que haya muchos errores de disco. Si hay una alta seguridad en los discos individuales y en las unidades de disco, RAID 2 es excesivo y no se implementa.

NIVEL 3 DE RAID

RAID 3 se organiza de una manera similar a RAID. La diferencia es que RAID 3 requiere solo un disco redundante, sin importar lo grande que sea el conjunto de discos. RAID 3 utiliza un acceso paralelo, con datos distribuidos en pequeñas tiras. En vez de un código de corrección de errores, se calcula un sencillo bit de paridad para el conjunto de bits individuales en la misma posición en todos los discos de datos.

REDUNDANCIA

En el caso de una falla en una unidad, se accede a la unidad de paridad y se reconstruyen los datos desde el resto del dispositivo. Una vez que se sustituye la unidad que ha fallado, los datos que faltan se restauran en la nueva unidad y se reanuda la operación.

La reconstrucción de los datos es bastante sencilla. Consideremos un conjunto de cinco discos de los que de x0 a x3 contienen datos y x4 es el disco de paridad. La paridad para el i-ésimo bit se calcula de la siguiente forma:

$$x4(i) = x3(i) \oplus x2(i) \oplus x1(i) \oplus x0(i)$$

Supongamos que la unidad x1 ha fallado. Si sumamos $x4(i)$ $x1(i)$ a ambos miembros de la ecuación, tenemos que:

$$x1(i) = x4(i) \oplus x3(i) \oplus x2(i) \oplus x0(i)$$

Por tanto, se puede regenerar el contenido de cualquier tira de datos en cualquiera de los discos de datos de un conjunto a partir del contenido de las correspondientes tiras del resto de los discos del conjunto. Este principio es valido para los niveles 3,4 y 5 de RAID.

En caso de que un disco falle, todos los datos están todavía disponibles en lo que se denomina modo redundancia. En este modo, para lectura, los datos que faltan se recuperan "al vuelo" con la operación OR-exclusiva. Cuando se escriben datos en un conjunto RAID 3 reducido, se debe mantener la consistencia de la paridad para generaciones posteriores. Volviendo al funcionamiento global, se requiere que el disco que ha fallado se reemplace y se regenere todo su contenido en el nuevo disco.

BENEFICIOS

Puesto que los datos se dividen en tiras muy pequeñas, RAID 3 puede conseguir velocidades de transferencia de datos muy altas. Cualquier petición de E/S implicara una transferencia de datos paralela desde todos los discos de datos. Para grandes transferencias, la mejora de beneficios es especialmente notable.

Por otra parte, solo se puede ejecutar a la vez una petición de E/S. Por tanto, en un entorno orientado a transacciones, el rendimiento sufre.

NIVEL 4 DE RAID

Los niveles 4 y 5 de RAID usan una técnica de acceso independiente. En un conjunto de acceso independiente, cada disco opera independientemente, de forma que peticiones de E/S separadas se atienden en paralelo. Debido a esto, son mas adecuados los conjuntos de acceso independiente para aplicaciones que requieren velocidades de petición de E/S altas, y son menos adecuados para aplicaciones que requieren velocidades altas de transferencias de datos.

Como en otros esquemas de RAID, se usan tiras de datos. En el caso de RAID 4 y 5, las tiras son relativamente grandes. Con RAID 4, se calcula una tira de paridad, bit a bit a partir de las correspondientes tiras de cada disco de datos, y los bits de paridad se almacenan en las correspondientes tiras del disco de paridad.

RAID 4 lleva consigo una penalización en la escritura cuando se realiza una petición de escritura de E/S pequeña. Cada vez que se realiza una escritura, el software de gestión del conjunto debe actualizar no solo los datos del usuario, sino también los bits de paridad correspondientes. Consideremos un conjunto de cinco unidades en las que de x0 a x3 contienen datos y x4 es el disco de paridad. Supongamos que se realiza una escritura que implica solo una tira del disco x1. Inicialmente, para cada bit i, tenemos la siguiente relación:

$$x4(i) = x3(i) \oplus x2(i) \oplus x1(i) \oplus x0(i)$$

Después de la actualización, indicamos con prima los bits que han sido alterados,

$$\begin{aligned} x4'(i) &= x3(i) \oplus x2(i) \oplus x1'(i) \oplus x0(i) \\ &= x3(i) \oplus x2(i) \oplus x1'(i) \oplus x0(i) \oplus x1(i) \oplus x1(i) \\ &= x4(i) \oplus x1(i) \oplus x1'(i) \end{aligned}$$

Para calcular la nueva paridad, el software de gestión del conjunto debe leer la antigua tira del usuario y la antigua tira de paridad. Entonces, se pueden actualizar estas dos tiras con nuevos datos y calcular la nueva paridad. Por tanto, cada escritura de una tira implica dos lecturas y dos escrituras.

En el caso de una escritura de E/S de mayor tamaño que implique tiras en todas las unidades de disco, la paridad se puede obtener fácilmente con un cálculo usando solamente los nuevos bits de datos. Por tanto, la unidad de paridad puede ser actualizada en paralelo con la unidad de datos, y no habrá lecturas o escrituras extras.

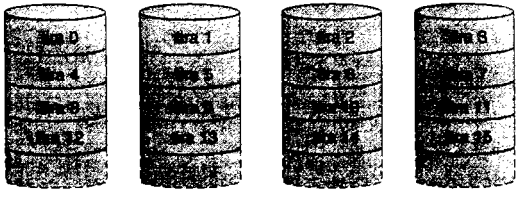
En cualquier caso, cada operación de escritura implica al disco de paridad, que por consiguiente se convertirá en un cuello de botella.

NIVEL 5 DE RAID

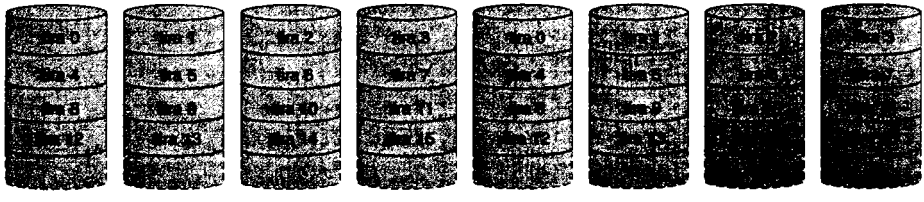
RAID 5 está organizado de manera similar a RAID 4. LA diferencia es que RAID 5 distribuye las tiras de paridad a lo largo de todos los discos. Una distribución típica es un esquema cíclico, como se muestra en la figura 3.2. Para un conjunto de n discos, la tira de paridad está en diferentes discos para las primeras n tiras, y este patrón se repite.

La distribución de las tiras de paridad a lo largo de todas las unidades, evita el potencial cuello de botella de E/S encontrado en RAID 4.

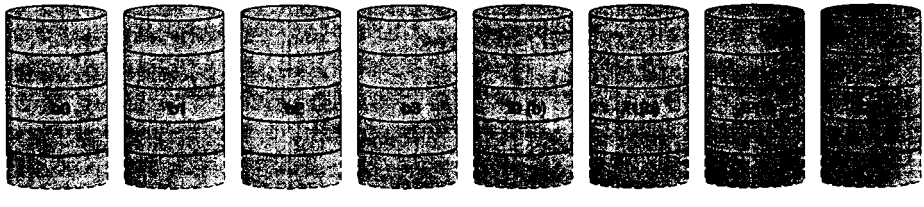
(a) RAID 0 (No redundante)



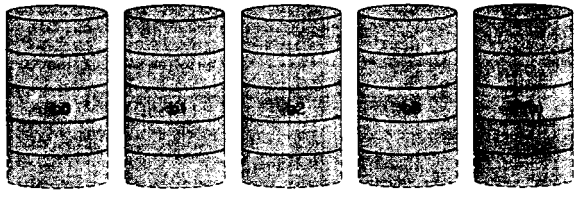
(b) RAID 1 (Con espejo)



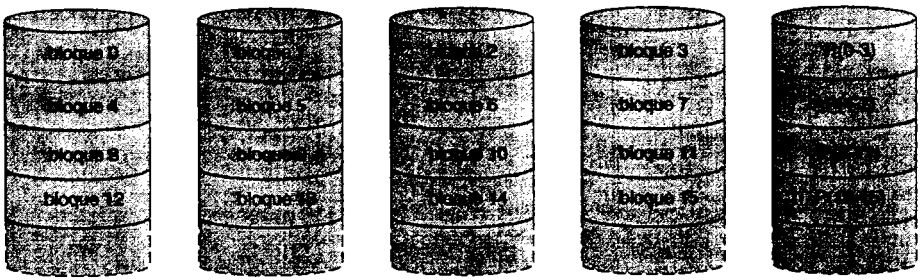
(c) RAID 2 (Redudante con código Hamming)



(d) RAID 3 (Bit de paridad intercalado)



(e) RAID 4 (Paridad en bloques)



(f) RAID 5 (Paridad distribuída a nivel de bloques)

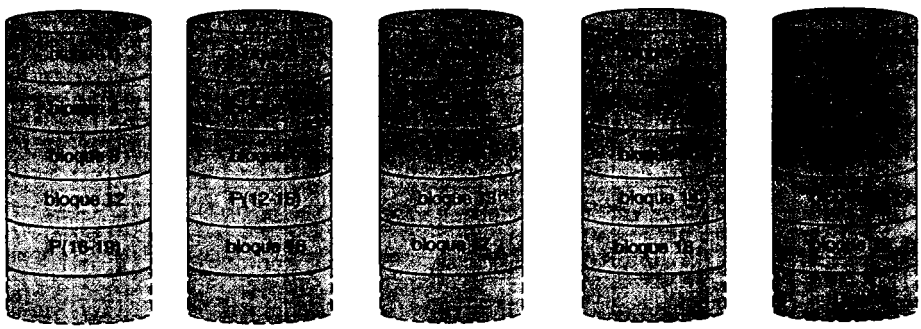


Figura 3.2
Esquemas de RAID

3.3 TOLERANCIA A FALLAS

Todo sistema físico sin excepción está sujeto a cambios en sus parámetros, los cuales modifican el comportamiento para el cual fueron diseñados. Estos cambios pueden ser debido a efectos de la temperatura, desgastes ocasionados por la fricción, el envejecimiento de los componentes, etc. Los cambios en los parámetros del sistema a zonas fuera de los límites de tolerancia especificados por el fabricante o de los límites establecidos de acuerdo a criterios de ingeniería, serán considerados como fallas. Estas modificaciones afectan en su mayor parte el buen funcionamiento del sistema provocando desde una reducción del desempeño hasta la posibilidad de accidentes más graves.

La rápida detección de la presencia de fallas en los sistemas puede ayudar a tomar acciones correctivas y de este modo reducir el daño potencial que esta falla puede ocasionar al sistema.

Al igual que en los sistemas físicos, también se presentan fallas en el software. Si una aplicación que calcula la solución a un problema científico falla, es razonable abortar el programa, ya que sólo se ha perdido tiempo de cómputo. En contraste, en el caso de un sistema de tiempo real esta puede no ser una solución aceptable. Una aplicación de control de procesos como es el control de un alto horno no se puede permitir el apagar el horno tan pronto como ocurre una falla porque las pérdidas económicas de esta operación son cuantiosas. Lo que es más importante, los sistemas pueden poner en peligro vidas humanas si abandonan el control de la aplicación.

Hoy en día, más y más funciones de control que previamente venían siendo desarrolladas por operadores humanos, así como métodos de control analógico suficientemente probados se administran mediante computadoras. En 1955 sólo el 10% del armamento de los Estados Unidos requería software. En 1986 la cifra se elevaba al 80%.

Antes de continuar, es preciso definir con rigor la terminología empleada en la disciplina de tolerancia a fallas:

- Se denomina fiabilidad de un sistema a la medida en que se conforme a la especificación autorizada de su comportamiento. Idealmente, esta especificación de comportamiento debería ser consistente y exhaustiva. Los tiempos de respuesta deben ser una parte importante de la especificación.
- Una avería es una desviación del comportamiento de un sistema respecto de su especificación. Vamos a considerar que un sistema es altamente fiable cuando se tiene una tasa de avería baja.
- De su definición misma se sigue que los conceptos de fiabilidad y avería se refieren al comportamiento del sistema, esto es, a su apariencia externa. Las averías son el resultado de estados inesperados en el interior del

sistema que eventualmente se manifiestan en el comportamiento exterior del sistema. Un estado interno no especificado se denomina error.

- La causa mecánica algorítmica de un error se denomina falla. Podemos distinguir tres tipos de fallas:
  - Fallas transitorias. Esta comienza en un instante dado del tiempo, permanece en el sistema durante algún tiempo y después desaparece. Suelen ocurrir en componentes de hardware que tienen una reacción adversa a una interferencia externa. Tras la interferencia, desaparece la falla. Es típico en sistemas de comunicaciones.
  - Falla permanente. Comienza en un instante dado del tiempo y permanece hasta que son reparados. Por ejemplo un cable roto o un error de diseño.
  - Fallas intermitentes. Son fallas transitorias que se producen de vez en cuando. Por ejemplo un componente de hardware que es sensible a la temperatura.

Hay que tener en cuenta que un sistema esta constituido por un conjunto de subsistemas o componentes. Por lo tanto, una avería en un componente A puede conducir a una falla en otro B. Esta falla en B dará lugar a un error que puede manifestarse como una avería en C, y así sucesivamente según se muestra en la figura 3.3

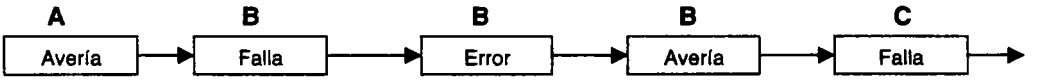


Figura 3.3
Cadena de averías, fallas y errores

3.3.1. Prevención y tolerancia de fallas

Hay dos formas de aumentar la fiabilidad de un sistema:

- Prevención de fallas. Se trata de evitar las fallas antes de que el sistema entre en funcionamiento
- Tolerancia de fallas. Se trata de conseguir que el sistema continúe funcionando aunque se produzcan fallas

En ambos casos el objetivo es desarrollar sistemas con modos de fallas bien definidos.

Falla de origen
Falta la página
67

La solución es usar técnicas de tolerancia a fallas. Esta puede proporcionarse a un sistema en diversos cursos en función de su naturaleza:

- Tolerancia completa (fail operational). El sistema sigue funcionando, al menos durante un tiempo, sin perder funcionalidad ni prestaciones
- Degradación aceptable (fail soft). El sistema sigue funcionando con una perdida parcial de funcionalidad o prestaciones durante la reparación de la falla
- Parada segura (fail save). El sistema se detiene en un estado que asegura su integridad y la del entorno hasta que se repare la falla

3.3.2 Detección de errores

Ningún esquema de tolerancia a fallas puede ser utilizado hasta que se detecta un error. La efectividad de un sistema de tolerancia a fallas depende de la efectividad de las técnicas de detección de errores. Dos técnicas de detección de errores pueden ser identificadas:

- Detección por el entorno de ejecución. Estos son los errores detectados por el entorno en que se ejecuta la aplicación.

  -Detección por el hardware: Fallas con "instrucción ilegal", "desbordamiento aritmético" o "violación de protección".
  -Detección por el núcleo de ejecución (RTS) o por el sistema operativo. Fallas como "error en límite de vector" o "valor fuera de rango". Estos tipos de error serán considerados en el contexto del lenguaje en capítulos posteriores.

- Detección por el software de aplicación. Son los errores detectados por la aplicación misma. Las técnicas de detección que pueden ser utilizadas por la aplicación caen en alguna de las siguientes categorías:

  -Duplicación de funciones (redundancia con 2 versiones).
  -Comprobaciones de tiempo. Podemos identificar dos tipos de comprobaciones. La primera es un proceso temporizador que se denomina **watchdog timer**. Si el temporizador no es reiniciado por el componente en un plazo dado, el componente se asume que está en un estado de error. La segunda se refiere a la detección plazos no cumplidos. La detección de esta falla la lleva a cabo el núcleo de ejecución.
  -Inversión de funciones. Son factibles en componentes donde hay un relación isomórfica (uno a uno) entrada y salida. Periódicamente puede comprobarse que el componente opera correctamente tomando una salida, evaluando la entrada funcionalmente correspondiente y comprobando esta con la entrada real.

-Códigos redundantes. Por ejemplo en el control de errores de comunicación.
-Validación de valores o estado. Comprueban si el estado o valor de un dato es razonable basado en la intención con la que se usa. El criterio puede ser proporcionado por el programador haciendo uso de facilidades del lenguaje como el subtipo generado con restricciones de un tipo en forma de rango. En general, la expresión que se valida se denomina **aserción**.

3.3.3 Evaluación y confinamiento de daños

Cuando se produce una falla, antes de que sea detectada y se ejecute el mensaje de error correspondiente, información errónea puede propagarse a otras partes del sistema y al núcleo de ejecución. Es importante, por lo tanto, confinar los daños causados por un error a una parte limitada del sistema. Por otra parte, la magnitud de los daños está estrechamente relacionada con las precauciones que hubiesen tomado los diseñadores del sistema en cuanto al confinamiento de esta. Se trata, pues, de construir compartimientos estancados, de tal forma que las fallas no puedan transmitirse de uno a otro. Hay varias técnicas que se puedan usar para estructurar el sistema de esa forma:

- Descomposición modular (confinamiento estático). Cuando el sistema se estructura en módulos con interfaces bien definidos, es más difícil que un error se transmita de un componente a otro. Hay que tener en cuenta que los módulos proporcionan una estructura estática del sistema, estructura que se pierde en tiempo de ejecución.

- Acciones atómicas (confinamiento dinámico). Igualmente importante para el confinamiento de daños es la estructura dinámica del sistema. Las acciones atómicas son un medio de estructura del sistema de forma dinámica. La actividad de un componente es atómica si no existe interacción con el sistema durante la duración de la acción. Para el resto del sistema la acción atómica de un componente tiene lugar instantáneamente y de forma indivisible. Las acciones atómicas se denominan también transacciones o acciones atómicas y sirven para llevar al sistema de un estado consistente a otro estado consistente y limitar el flujo de información entre componentes.

- Mecanismos de protección (acceso a recursos). Cuando dos componentes comparten un recurso, una falla en un componente puede dañar el recurso, que entonces actúa como transmisor de la falla. El acceso debe ser, por lo tanto, cuidadosamente protegido y controlado.

3.3.4 Recuperación de errores

Una vez que se ha detectado una situación de error y valorado y confinado su daño, puede iniciarse los procedimientos de recuperación. Ésta es la fase más importante de cualquier técnica de tolerancia a fallas. Consiste en sustituir el sistema en un estado correcto, desde el que pueda continuar la ejecución, aunque quizá con un nivel de servicio inferior. Hay dos formas de llevarla a cabo:

- Recuperación directa. Consiste en avanzar desde el estado erróneo hacia un nuevo estado correcto (también en el sistema controlado). Hay que realizar lo que se denomina valoración de daños, es decir, realizar predicciones precisas de la ubicación y causa del error. Ejemplos de recuperación directa incluye punteros redundantes en estructuras de datos y códigos autocorrectores.

- Recuperación inversa. Consiste en retroceder desde el estado erróneo a un estado anterior correcto (punto de recuperación). A partir de ésta, se ejecuta un bloque alternativo de la aplicación (con otro algoritmo). Sirve para recuperar el sistema ante fallas imprevistas, pero no puede deshacer los efectos que la falla puede haber tenido sobre el entorno del sistema controlado; es difícil deshacer los efectos del lanzamiento de un misil, por ejemplo. El punto al que el estado es restaurado se llama punto de recuperación.

Para establecer un punto de recuperación es necesario guardar información del estado apropiado en tiempo de ejecución.

Cuando en el sistema intervienen procesos concurrentes que se comunican, la recuperación inversa es más difícil. Consideremos los dos procesos de la figura 3.4 A lo largo del tiempo, cada uno de ellos va estableciendo sus propios puntos de recuperación R_{ij}. Los procesos se comunican haciendo uso de las primitivas IPC_j. Si P_1 detecta errores en el instante T_e, regresa al estado dado por su punto de recuperación R_{13} y el sistema vuelve a un estado consistente. Ahora bien, supongamos que es P_2 el que detecta el error en el instante T_e. Debe regresar al punto R_{22} con el inconveniente de que ya no es válida la comunicación IPC_4 y ésta debe ser anulada. Para anular IPC_4, P_1 debe volver a su punto de recuperación R_{12}, pero con la inconvención de que IPC_3 debe ser anulada y, por lo tanto, el punto de recuperación previsto inicialmente para P_2, R_{22}, no es válido y, en consecuencia, P_2 debe regresar a un punto de recuperación anterior, R_{21}. A su vez, el regreso de P_2 a R_{21} impone deshacer la comunicación IPC_2 y el retorno de P_1 al estado R_{11}. Este retorno implica que IPC_1 no es válida y el retorno de P_2 a su estado inicial. Este fenómeno en el que la recuperación inversa de un proceso implica deshacer todos los procesos con los que se comunican es conocido como efecto dominó. La solución al efecto dominó es establecer puntos de recuperación globales para el conjunto de procesos que interactúan. La secuencia de estos puntos de recuperación es conocida con el nombre de línea de recuperación.

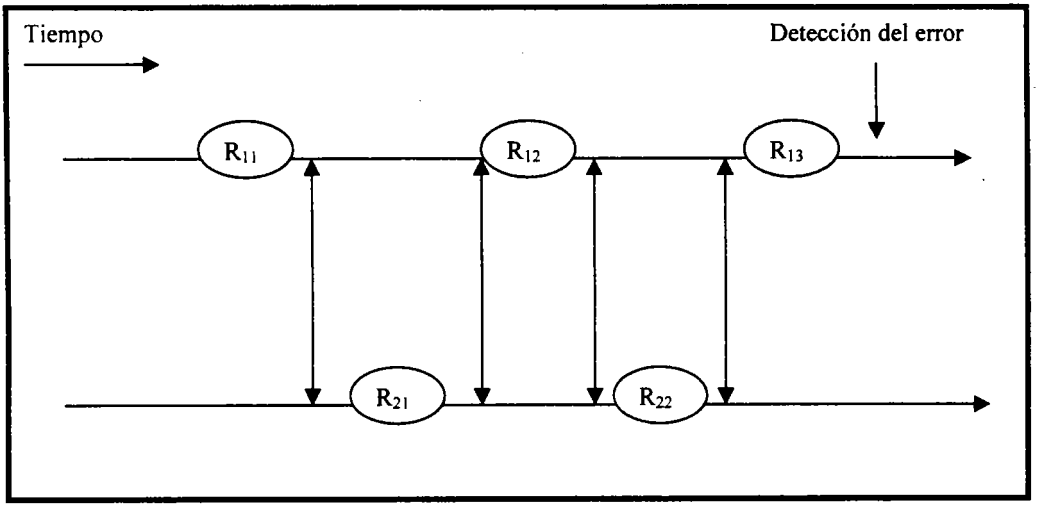


Figura 3.4
El efecto dominó

Reparación de fallas

Un error es la manifestación de una falla y aunque la fase de recuperación de la falla puede haber devuelto al sistema a un estado correcto, el error puede ocurrir. Por lo tanto, la fase final de la tolerancia a fallas es erradicar la falla del sistema de forma que pueda continuar prestando el servicio ordinario.

La reparación se divide en dos etapas: la localización de la falla y la reparación propiamente dicha. Las técnicas de detección de errores ayudan a seguir la pista del fallo de un componente.

La reparación automática de fallas es difícil de implementar y tiende a ser específica del sistema. Consecuentemente, algunos sistemas no realizan reparación de fallas asumiendo que estas son transitorias; otros asumen que la técnica de recuperación de errores es suficiente para eliminar la falla.

3.4 REPLICACIÓN

Las organizaciones que requieren un tiempo muy corto de recuperación en caso de desastre tienden a utilizar un tipo de replicación síncrona. En éste método, los datos se duplican del sitio primario al sitio secundario en tiempo real. Dependiendo a que nivel se realizará la replicación, ya sea a nivel de aplicación o del subsistema de almacenamiento, la replicación síncrona implica diferentes procesos.

A nivel de aplicación, involucra un proceso llamado ejecución en dos fases, ya que cuando alguna transacción debe ser replicada al sitio secundario, es necesario disponer de la respuesta afirmativa del sistema secundario y poder proceder con la siguiente. En caso de no tener la respuesta del sitio secundario no se procede con la siguiente transacción, asegurando de esta forma que la información es idéntica en ambos sitios. El ejemplo más común de este tipo de replicación es la que se produce a nivel de base de datos como SQL Server. En este escenario los agentes SQL que se encuentran activos en el servidor de base de datos primario y secundario, controlan el proceso de replicación, mediante el respaldo del log de transacciones de la base de datos por parte del agente en el sitio primario, y la aplicación del mismo en el sitio secundario por el agente destino.

A nivel de subsistema de almacenamiento, cada vez que se realiza la escritura en el subsistema principal, se realiza la misma en el subsistema secundario, de forma inmediata. En caso de no lograr realizar la escritura en el sitio de contingencia, no se permite almacenar datos en el sistema principal, de tal forma que se mantienen copias idénticas de la información. De todas formas es posible configurar la replicación para que continúe la escritura en el sitio principal y cuando sea posible se actualice el secundario; pero esto sería más cercano a una replicación asíncrona que a una síncrona.

El objetivo principal en la replicación síncrona de datos, es el de lograr que la pérdida de datos en caso de desastre sea casi nula y poder disponer de una recuperación rápida cuando ocurre la caída de uno de los sitios. Es claro que este proceso de ejecución en dos fases puede producir degradación del desempeño cuando las distancias entre los sitios son grandes y/o cuando el medio de comunicación utilizado no proporciona un ancho de banda necesario para el tráfico en cuestión. Debido a que este tipo de replicación necesita de un ancho de banda importante, los costos asociados a este escenario son bastante elevados.

Otra opción que se ha tomado más popular debido a sus menores costos involucrados, es la replicación asíncrona. Este tipo de tecnología se basa en la captura de transacciones completadas en el sitio principal, para ser aplicadas luego en los sitios secundarios. Esta duplicación ocurre automáticamente cuando existe algún cambio o en intervalos de tiempo predefinido por el administrador.

Es posible establecer estos intervalos de actualización cuando la utilización del canal de comunicaciones sea baja. La replicación asíncrona, a diferencia del caso

síncrono, no requiere de gran cantidad de ancho de banda para poder operar. Esto permite aplicar esta solución sobre distancias mayores sin gran degradación de desempeño y por tanto se pueden utilizar canales de comunicación más económicas. En contraposición es importante señalar que los tiempos de recuperación y la pérdida de datos en caso de desastre son mayores que en un escenario síncrono. En este tipo de escenarios la aplicación acumula una serie de transacciones realizadas en el sitio primario y luego las replica al sitio secundario, sin necesidad de finalizar las dos fases para que la transacción del sitio primario sea válida. El intervalo en que se realiza esta actualización dependerá de las necesidades propias de la organización.

3.5 RECUPERACIÓN DE DESASTRES

3.5.1 Tipos de desastres

En general, desastre es cualquier evento que, cuando ocurre tiene la capacidad de interrumpir la operación normal de una organización. Este incidente puede ser causado por:

Inundaciones Huracanes Temblores Tornados Incendios	Explosiones Contaminación Derramamiento de material peligroso Pérdida del servicio telefónico Pérdida de poder	Sabotaje Vandalismo Provocación de incendios

Si llegará a pasar, que debido a cualquier razón –fuego, explosión, sabotaje- la computadora falla, y no se tiene la capacidad de levantarla en un plazo máximo de 5 días. ¿Qué procedimientos de respaldo existen en el sistema en línea que ordene los procesos?; ¿cuáles son los requerimientos del cliente?; ¿cómo determinamos la disponibilidad de inversión?, etc.

3.5.2 Ciclo de vida de los desastres

El ciclo de vida de un desastre consta de cuatro periodos de tiempo:

a) Operaciones normales
b) Respuesta de emergencia
c) Procesamiento interno
d) Restauración

Operaciones normales

Las operaciones normales indican el periodo de tiempo antes de que ocurra un desastre. Esta fase del plan debe incluir la práctica de las operaciones que pretenden prevenir un desastre desde que inicia, y de aquellas que ayudan a mitigar el impacto del mismo, prever lo que podría ocurrir.

Respuestas de emergencia

Las respuestas derivadas de una situación de emergencia ocurren durante las pocas horas que siguen inmediatamente a un desastre. Esta fase de un plan identifica las actividades que pueden necesitar mayor atención durante ese periodo, con la finalidad de asegurar una respuesta a la organización y

proporcionar una lista de verificación de las emisiones importantes que pueden pasar inadvertidas durante la confusión que acompaña a los desastres.

Procesamiento interno

El procesamiento interno es un procesamiento alternativo que representa el tiempo de duración de la contingencia en relación con el soporte de las funciones esenciales de la empresa hasta que la capacidad de procesamiento normal sea restaurada. Estos procedimientos alternativos, deberían ser desarrollados por un departamento funcional dividido en tres fases:

- Inicio: Esta sección identifica la necesidad de una preparación específica para asegurar las transacciones desde "el negocio como usual" hasta una modalidad de procesamiento interno
- Soporte de las funciones esenciales del negocio: Esta sección describe los departamentos funcionales que están de acuerdo en el soporte de las funciones vitales del negocio durante el periodo de recuperación de desastres
- Recuperación de datos: En esta sección, el plan cubre las responsabilidades funcionales para retener los datos transaccionales que ocurren durante el periodo de procesamiento interno, así que los archivos y bases de datos pueden ser actualizados cuando la capacidad de procesamiento normal sea restaurado

Restauración

La restauración indica el periodo de tiempo destinado a aquellas actividades que se necesita realizar para recuperar una condición o capacidad de procesamiento en su operación normal. La restauración involucra necesariamente los pasos de la planeación, organización y control de tales actividades.

Cuando el clima económico es favorable, los planes de contingencia están al final de la lista de cosas que se necesitan hacer; cuando los beneficios son bajos, los planes de contingencia forman parte de las actividades prioritarias de la empresa ya que se trata de asegurar el negocio, además mientras mayor sea el costo en los proyectos del plan de contingencias, mayor será su plazo.

Respaldo

Cuando un servicio falla, la tarea principal del servidor debe ser la de recuperación; mientras que la responsabilidad principal del usuario es la de dar continuidad a las operaciones.

3.5.3 Fases de un plan de contingencia

Fase I. Análisis y Diseño

Estudia la problemática, las necesidades de recursos, las alternativas de respaldo, y se analiza el costo/beneficio de las mismas. Ésta es la fase más importante, pudiendo llegar al final de la misma incluso a la conclusión de que no es viable o es muy costoso su seguimiento. En la forma de desarrollar esta fase, se diferencian las dos familias metodológicas: análisis de riesgo e impacto en el negocio.

Las de análisis de riesgo se basan en el estudio de los posibles riesgos desde el punto de vista de probabilidad de que los mismos sucedan. Aunque los riesgos de incidente son escasos y poco fiables, aún así es más fácil encontrar este tipo de metodologías que las segundas.

Las de impacto en el negocio, se basan en el estudio del impacto (pérdida económica o de imagen) que ocasiona la falta de algún recurso de los que soporta la actividad del negocio. Estas metodologías son más escasas, pero tienen grandes ventajas como es el mejor entendimiento del proceso o el menor empleo de tiempo por ir más directamente al problema.

Las tareas de esta fase en las distancias metodologías planteadas son las siguientes:

Análisis de riesgo

1. Identificación de amenazas
2. Análisis de la probabilidad de materialización de la amenaza
3. Selección de amenazas
4. Identificación de entornos amenazados
5. Identificación de servicios afectados
6. Estimación de impacto económico por paralización de cada servicio
7. Selección de los servicios a cubrir
8. Selección final del ámbito del plan
9. Identificación de alternativas para los entornos
10. Selección de alternativas
11. Diseño de estrategias de respaldo
12. Selección de la estrategia de respaldo

Impacto en el negocio

1. Identificación de servicios finales
2. Análisis del impacto. En estas metodologías se evalúan los daños económicos y de imagen y otros aspectos no económicos.
3. Selección de servicios críticos
4. Determinación de recursos de soporte

5. Identificación de alternativas para entornos
6. Selección de alternativas
7. Diseño de estrategias globales de respaldo
8. Selección de la estrategia global de respaldo

Hay un factor importante a determinar en esta fase que es el time frame o tiempo que la organización puede asumir con paralización de la actividad operativa antes de incurrir en pérdidas significativas. Este factor marcará las estrategias de recuperación y se extraerá del análisis del impacto.

Fase II. Desarrollo de un plan

Esta fase y la tercera son similares en todas las metodologías. En ella se desarolla la estrategia seleccionada, implantándose hasta el final todas las acciones previstas. Se definen las distintas organizaciones de emergencia y se desarrollan los procedimientos de actuación generando así la actuación del plan.

Es en esta fase cuando se analiza la vuelta a la normalidad, dado que pasara de la situación normal a la alternativa, debe concluirse con la reconstrucción de la situación inicial de la contingencia.

Fase III. Pruebas y mantenimiento

En esta fase se definen las pruebas, sus características y sus ciclos, y se realiza la primera prueba como comprobación de todo el trabajo realizado, así como concientizar al personal implicado.

Así mismo se definen la estrategia de mantenimiento, la organización destinada a ello, y las normas y procedimientos necesarios para llevarlo acabo.

Herramientas

El esquema de una herramienta debe tener al menos los siguientes puntos:

- Base de datos relacional
- Módulo de entrada de datos
- Módulo de consultas
- Procesos de textos
- Generador de informes
- Ayuda en línea
- Hoja de cálculo
- Gestor de proyectos
- Generador de gráficos

3.5.4 Características de un plan de contingencia

Un plan de contingencia debe de:

- Tener la aprobación de los integrantes
- Ser flexible
- Contener un proceso de mantenimiento
- Tener un costo efectivo
- Enfatizar en la continuidad del negocio
- Asignar responsabilidades específicas
- Incluir un programa de prueba

Aprobación

El plan debe ser aceptable para auditores internos; fuera de auditores, el director, clientes y proveedores.

Flexibilidad

El plan deberá ser especificado en guías, en lugar de relacionar los detalles a situaciones individuales del desastre.

Mantenimiento

Eludir detalles innecesarios de manera que el plan puede ser fácilmente actualizado

Costo-efectividad

La planeación del proyecto deberá enfatizar en la necesidad de minimizar los costos del desarrollo del plan, respaldo redundante del procesamiento de la subscripción de honorarios, mantenimiento y costo de pruebas.

Continuidad de la empresa

El plan debe de asegurar la continuidad, durante un periodo de recuperación de desastres.

Respuesta organizada

El plan debe proporcionar una lista de verificación de salidas que necesitan atención inmediata que sigue al desastre. Así mismo incluirá listas de números de teléfono y las direcciones de individuos para contactarlos.

Responsabilidad

A individuos específicos deberá asignárseles la responsabilidad de cada salida que requiera atención durante la respuesta de emergencia y el tiempo del periodo del procesamiento interno.

Prueba

La prueba con los usuarios para revisar los procedimientos de verificación de respaldo debe de realizar algo específico en los intervalos de tiempo. De tal forma que el plan cuente con un estado de frecuencias de prueba y documente la metodología de prueba.

3.5.5 Identificaciones previas

3.5.5.1 Aplicaciones y activos críticos

Se denomina aplicaciones y activos críticos a aquellos cuya falta de disponibilidad causaría graves dificultades en la continuidad de las actividades del negocio.

Cada propietario de aplicación tiene que analizar que supondría para la empresa (ingresos que se dejarían de percibir, materiales que se estropearían, horas de trabajo que se perderían, etc.) la imposibilidad de procesar su aplicación. Deben tenerse en cuenta los siguientes factores:

- Que puede ser crítica solamente una parte de la aplicación o sólo algún activo de información necesario para otra aplicación, en la misma función o en otra diferente.
- El tiempo máximo que podría subsistir la empresa sin procesar la aplicación.

Con estas bases, cada propietario tiene que seleccionar y proponer al director funcional las aplicaciones y activos candidatos a ser nominados críticos, con la valoración de los costes asociados a la recuperación y las pérdidas potenciales, si no se recuperase.

La propuesta debe incluir una estimación de la periodicidad con que deberían efectuarse las copias de respaldo y la valoración de costos por el posible mantenimiento, en el departamento, de los datos de entrada en la aplicación durante el período existente entre dos copias de respaldo.

Cada director funcional, asistido por los propietarios, tiene que consolidar propuestas de los propietarios y determinar las aplicaciones y activos candidatos, a ser propuestos como críticos, teniendo en cuenta:

- Los activos de información necesarios para alguna aplicación de otras funciones
- El tiempo máximo que podría subsistir el negocio de la empresa sin el conjunto de aplicaciones activos críticos de la función, y proponerlos en el comité de dirección, del que forma parte, incluyendo en la propuesta la valoración consolidada de costos de recuperación y pérdidas potenciales.

El comité de dirección tiene que analizar el impacto que podría ocasionar, en el negocio de la empresa, la carencia prolongada de recursos, aplicaciones y activos y la no disponibilidad de la información que suministran, teniendo en cuenta todos los aspectos necesarios para la recuperación. Adicionalmente, deben valorarse las situaciones en las que podría ser declarada una emergencia y, en cada caso, quien está autorizado a declarar la situación de desastre, total o parcial.

La decisión final, y toda la documentación usada para ello, es de naturaleza confidencial y tiene que ser guardado como documento auditable y comunicada a las funciones y departamentos involucrados.

3.5.5.2 Centro Alterno

A propuesta de la función del sistema, el comité de dirección tiene que aprobar la alternativa más eficaz con el menor costo posible.

Si la empresa tiene capacidad para realizar la recuperación en un centro alterno propio, debe tener prioridad esta opción, pero documentándolo en un acuerdo formal, que debe poder ser auditado.

La otra opción, es contratar un centro alterno con alguna compañía especializada. Como regla general, el contrato contempla los conceptos de cargo siguientes:

- Por reserva de recursos (capacidad, almacenamiento, etc.)
- Por prueba realizada y/o planificada
- Por desastre real, en tiempo de ocupación
- Por soporte técnico y asesoría

Este último es el que puede considerarse como diferenciador entre un servicio de valor agregado y uno de simple disposición de los medios físicos para la recuperación.

En cualquiera de las dos opciones, el centro alterno, propio o externo, tiene que cumplir como mínimo los requisitos de seguridad física existentes para el centro origen.

Al margen de las consideraciones anteriores, el contrato, en cualquier caso, debe garantizar la disponibilidad del centro alterno y la realización de las pruebas periódicas de recuperación en las fechas predeterminadas. Antes de la firma del

contrato debe revisarse el centro alterno, verificar qué es adecuado y qué cubre las necesidades de recuperación previstas para la empresa. El contrato y el informe de la revisión previa deben guardarse como documentos auditables.

Todo el proceso de identificación descrita, tiene que ser repetido siempre que existan modificaciones en los servicios suministrados por el sistema, que así lo aconsejen, o al menos anualmente. El momento recomendado para esta revalidación es durante el análisis de los resultados de una prueba de recuperación.

3.5.6 Copias de Respaldo

Determinar su periodicidad, debe crearse un procedimiento que contenga los detalles de obtención de estas copias, de acuerdo con el fin para el que son creadas, recuperación parcial o total, y el destino, el propio centro alterno.

La obtención de un juego de copias de respaldo, puede realizarse:

- Sobre medios de almacenamiento desmontable, lo que implica su traslado al centro de almacenamiento alterno a la mayor brevedad posible y planificar el mantenimiento de, al menos tres juegos de copias de respaldo para asegurar, que al menos un juego permanece siempre en el centro de almacenamiento alterno de forma rotatoria. En el traslado tiene que depositarse el nuevo juego de copias y recoger para su reproceso el más antiguo de los existentes.

- Mediante la transferencia electrónica de los activos de información al centro de almacenamiento alterno, donde se cuenta con los dispositivos de grabación y lectura adecuados. Esta opción resuelve los problemas derivados del transporte, aunque hay que tener en cuenta la cantidad de activos de información y la distancia como nuevos parámetros que pueden ser causa de problemas e incremento en el costo.

Es recomendable planificar la obtención de las copias de respaldo al terminar todos los procesos diarios, con lo que todos los activos de información estarán actualizados y al mismo nivel.

3.5.6.1 Almacenamiento en el centro alterno

En el caso en que las copias de respaldo no se guarden en un centro de almacenamiento alterno, tiene que habilitar una zona aislada, controlada por un custodio, como almacenamiento en el centro alterno, para trasladar y guardar los juegos de copias de respaldo, para la recuperación en el centro alterno. Las características de protección y control de acceso físico tiene que ser las mismas que en la zona aislada original.

Si no existe un custodio de la empresa en el centro alterno, los juegos de copias tienen que permanecer cerrados, en los contenedores que son trasladados, y no accesibles por ninguna persona ajena a la empresa.

3.5.6.2 Almacenamiento en el Propio Centro

Tiene que habilitarse una Zona Aislada, controlada por un custodio, como almacenamiento alterno en el propio centro, para trasladar y guardar los juegos de copias de respaldo para la recuperación en el propio centro. Las características de protección y control de acceso físico tienen que ser las mismas que en la Zona Aislada original.

3.5.6.3 Respaldo Funcional Complementario

Durante el período de tiempo existente entre dos copias de respaldo consecutivas, las funciones de la empresa siguen introduciendo datos en los Activos de Información o modificando programas de las aplicaciones, si unos u otros han sido declarados críticos para la recuperación del negocio, todas las modificaciones realizadas tienen que ser retenidas por la función hasta que otra copia de respaldo las incluya.

Si se declara una situación de desastre, una vez restaurada la copia de respaldo al sistema alternativo, todos los Activos incluidos en el Sistema con posterioridad a la obtención de la copia de respaldo, tienen que ser recuperados para tener en el Sistema la situación en el momento del desastre.

Este plan complementario tiene que ser realizado por cada uno de los propietarios de aplicaciones o activos críticos y consolidado a nivel función, con la participación del coordinador funcional de seguridad.

Tiene que prevenirse que un desastre afecte a la función y pueda deteriorar o destruir totalmente esta información complementaria.

3.5.7 Pruebas de Continuidad

La función de sistemas de Información tiene que tener analizadas las alternativas, en lo referente a la capacidad de proceso y almacenamiento, para determinar en qué supuestos se realizaría la recuperación en el propio centro, ya que esta opción debe ser prioritaria, siempre que sea factible.

Entre las actividades planificadas para la recuperación en caso de desastre, tienen que incluirse las pruebas de continuidad de operaciones en el propio centro, en el centro alterno, la revisión de los procedimientos y su correspondiente actualización.

Tiene que planificarse, al menos, una prueba de continuidad al año, de tal forma que pueda garantizarse la recuperación en caso de desastre.

Una declaración ficticia de desastre, parcial o total, tiene que desencadenar la realización de una prueba de recuperación planificada, en el propio centro o en el centro alterno.

La función de Sistemas de Información tiene que nombrar formalmente un responsable de, al menos, las actividades siguientes:

- Coordinar con la dirección de las funciones la realización de las pruebas de Recuperación.
- Recabar la conformidad de los propietarios con los resultados de las pruebas de Recuperación
- Emitir los informes de los resultados de las pruebas de recuperación, y actualizar los Procedimientos de recuperación, siempre que haya cambios que así lo aconsejen, al menos una vez al año.

Durante las pruebas no puede usarse ningún Activo de Información que no esté contenido en las copias de respaldo. Cualquier utilización adicional, aunque sea encaminada a la finalización de las pruebas, tiene que ser incluida en el informe como incidente y la prueba declarada no satisfactoria.

Todos los incidentes detectados durante las pruebas tienen que quedar reflejados en el informe. Los propietarios o funciones afectadas tienen que tomar las medidas necesarias para corregirlos, mediante un plan de acción. Si algún incidente, habido en el transcurso de las pruebas, impide la recuperación, la prueba tiene que ser declarada no satisfactoria y repetida antes de 3 meses.

Una prueba se considera terminada de forma satisfactoria, cuando se ha podido recuperar el sistema y la información al mismo nivel que tenían en el momento de la declaración (ficticia) de desastre.

3.5.7.1 En el centro alterno

Durante las pruebas de recuperación se pueden diferenciar tres fases:

1. Fase de PREPARACIÓN, que no debe durar más de 24 horas desde la declaración de desastre, e incluye las actividades siguientes:
    - Traslado al centro alternativo
    - Recuperar las copias de respaldo más recientes
    - Volcar las copias de respaldo
    - Recuperar el sistema operativo

- Recuperar las aplicaciones y activos críticos
- Verificar el nivel del servicio recuperado. Al término de esta primera fase, el servicio debe reflejar la misma situación que cuando se obtuvieron las copias de respaldo.

2. Fase de EJECUCIÓN, cuya duración estará en función de la periodicidad de la obtención de las copias de respaldo (cuanto mayor sea el periodo entre dos copias, más tiempo tendrá que emplearse en la introducción de la información de las funciones), e incluye las actividades siguientes:

    - Comenzar las actividades de continuidad o introducir la información complementaria de las funciones.
    - Procesar las aplicaciones críticas hasta la recuperación total. Anotar los incidentes ocurridos.
    - Comprobar los resultados con las funciones (propietarios). Al término de esta fase, el servicio debe reflejar la situación que tenía cuando se declaró la situación de desastre.

3. Fase de INFORMACIÓN, en caso de que la prueba resulte satisfactoria, su duración no debe sobrepasar las 48 horas. Esta última fase contiene las actividades siguientes: Solicitar conformidad formal (por escrito) de las pruebas, a los propietarios. Realizar y distribuir el informe de las pruebas a propietarios y directores funcionales. Actualizar los procedimientos de recuperación. Si los resultados de la prueba son satisfactorios, la recuperación termina aquí. En caso contrario, se tiene que continuar con las actividades siguientes:

    - Crear los planes de acción para corregir los incidentes, fijando la fecha de repetición de la prueba
    - Realizar el seguimiento de los planes de acción e informar a la Dirección
    - Repetir la prueba antes de tres meses

Hasta la repetición de la prueba, la instalación debe mantenerse en situación de emergencia.

3.5.7.2 En el propio centro

Excepto el traslado al centro alterno, deben realizarse las mismas actividades que en el apartado anterior. La recuperación en el propio centro, implica la asignación de Recursos Informáticos al servicio esencial siniestrado, y por tanto la suspensión o la reducción de capacidad de otro servicio no esencial para el negocio de la empresa.

Generalmente, esto supone que el servicio recuperado funcione también degradado, al no tener la misma capacidad de proceso y almacenamiento.

3.5.7.3 Recuperación del centro siniestrado

Tiene que realizarse un análisis y una valoración de las posibles pérdidas de bienes muebles e inmuebles, Recursos Informáticos y Activos de Información, en caso de un desastre real.

La sustitución temporal está contemplada en apartados anteriores con la recuperación de los sistemas y servicios esenciales.

En este apartado se tiene que considerar la reparación del centro siniestrado, o su sustitución por otro de nueva construcción, con los costos asociados, el plazo de tiempo requerido en una situación de emergencia, que no debería sobrepasar los seis meses para la total recuperación, y el retorno al centro origen. A efectos de valoración, tienen que considerarse ambas alternativas de desastre, el total y el parcial.

Este plan es el único que teniendo que estar realizado y actualizado, no debe ser probado por el elevado costo que representaría.

3.5.8 Situación de Desastre

Cuando un problema, de cualquier índole o naturaleza, paraliza, total o parcialmente, los servicios informáticos de la empresa, el primer paso tiene que ser el análisis de la situación y sus posibles soluciones alternativas. Las soluciones tienen que ser evaluadas e informadas a la Dirección, quien procederá a declarar, o no, la situación de desastre:

- Parcial, a recuperar en el propio centro, con indicación de los sistemas esenciales a recuperar.
- Total, a recuperar en el centro alterno. En ambos casos, la situación de desastre debe estar declarada antes de 24 horas desde que se produjo el evento, y desencadenar los mecanismos planificados para la continuidad de las operaciones.

3.5.8.1 Continuidad de operaciones

Una vez declarada la situación de desastre, total o parcial, tienen que desencadenarse las actividades de recuperación de los sistemas esenciales siniestrados.

Las actividades a realizar son las mismas que están descritas en el apartado correspondiente a las pruebas planificadas.

En casos de desastre total y recuperación en el centro alterno, tiene que considerarse, adicionalmente, el retorno al centro origen, reparado o sustituido, antes de 6 meses.

Es necesario resaltar que la diferencia sustancial con una prueba, es que, por tratarse de un desastre real, un resultado no satisfactorio en la recuperación supondría pérdidas de recursos y activos con consecuencias económicas que en algunos casos pueden ser irrecuperables.

Hasta el momento se han visto diferentes técnicas que ayudan a tener una mejor disponibilidad de los sistemas y por consiguiente de los datos, pero no sirve de mucho si se implementan y ocurre un desastre, a excepción de la replicación y copias de respaldos mantenidos fuera del centro de cómputo cualquiera de las demás técnicas que se implementen no funcionarán en caso de pérdida total del sitio y no se tiene contemplado un plan de recuperación de desastres.

Este nivel de disponibilidad es el más alto y difícil de lograr, se deben contemplar planes los cuales se deberán llevar acabo en caso de un desastre y de estos dependerá hasta la futura existencia de una organización.

Los desastres van desde aquellos creados por el hombre, ejemplo de esto son los actos de terrorismo o la guerra, como los naturales tales como inundaciones o temblores. Lo anterior puede resultar en un cambio significativo en el quehacer del día a día de una organización; pérdida de todo el sistema de cómputo, de la propiedad y hasta de vidas.

Los planes y procedimientos de una recuperación de desastres siempre dependen de la compañía, la ubicación, la gente, entre otros factores que determinan los requerimientos específicos en caso de desastre.

Para una recuperación de desastres se necesita considerar lo siguiente:

Servidores de respaldo a distancia considerable. En un escenario de recuperación de desastres los servidores están situados a kilómetros lo cual ayuda a que el impacto no sea de ambos lados considerando los desastres naturales; no sirve de mucho tener el sitio alterno dentro del área de una falla geológica o dentro de la misma zona de tornados.

Servidores con recursos separados. En un desastre se debe estar preparado para la perdida total del sitio principal, para lograrlo es necesario mantener totalmente a los sistemas separados e independientes, no será permitido la compartición de recursos y en estos casos se aumenta la complejidad al ser necesario una replicación de datos en una WAN u otro método que transporte los datos críticos del sitio principal al sitio de recuperación.

Dar el servicio de manera normal desde el sitio alterno. Después de un desastre puede ser que no sea posible regresar al sitio original de trabajo y será

necesario adecuar lo antes posible el sitio de recuperación tal y como el original verificando la replicación o recuperando de respaldos y dependiendo del monto de datos será el tiempo en que se pueda volver a trabajar con normalidad.

También tener listos planes ya probados anteriormente con simulacros de traslado de recursos materiales y humanos con el fin de consumir el menor tiempo posible para volver a operar.

El tener un sitio de recuperación de desastres implica un gran desembolso económico debido a que se deberá tener el equipo a similitud del original y con una manutención constante y a pruebas de simulacros a la espera de un desastre y activarse, pero mientras esto no ocurra representará una pérdida en la economía de la organización pues se tendrá inactivado el equipo sin ser en lo más mínimo productivo.

En este capítulo se vieron diferentes métodos de hacer que los datos y los sistemas cuenten con un cierto grado de disponibilidad desde la más básica y menos costosa hasta la más completa y cara de implementar, sin duda cada técnica será implementada de acuerdo a las necesidades y recursos de una organización. Cuanto más disponibilidad de datos es requerido más se acrecentan los costos de inversión en los sistemas y éste es un factor determinante en el nivel de disponibilidad con el que contará la organización y en los riesgos que esté dispuesta a tomar así como del futuro de ésta.

4. ELEMENTOS DE DECISION PARA QUE UN SISTEMA SIEMPRE ESTE DISPONIBLE

Una vez discutidas las métricas y costos así como de las diferentes técnicas para lograr la disponibilidad en anteriores capítulos, es evidente que una disponibilidad lo más cercana a la total tiene un costo considerablemente alto pero que evaluado con las pérdidas que pueda representar la inactividad de un sistema éste costo muy posiblemente resulte una inversión.

El punto de vista de inactividad de un sistema es más palpable para quien administra sistemas que para un usuario final, el primero vera la inactividad ya sea por un fallo, por un mantenimiento o por una actualización requerida, en estos dos últimos casos la inactividad no será factor de pérdida ya que normalmente se llevan a cabo en horas o días inhábiles y serán imperceptibles para los usuarios finales.

En el caso de una empresa que ofrezca servicios 7 días x 24 horas x 365 días con un sistema de cluster implementado, una actividad que requiera dar de baja al sistema será algo transparente pues el contar con redundancia de componentes hará posible el trabajo aún en horas hábiles, primero se procederá en el equipo que esté inactivo, es decir el que está a la espera de un fallo del activo, una vez terminado se procederá con una migración de servicios del equipo primario al secundario para darlo de baja proceder con la misma actividad en el siguiente equipo.

Dicho lo anterior es posible visionar un sistema altamente disponible pero que aun cuenta con algunos huecos los cuales pueden ser minimizados si se toman en cuenta los temas a tratar en el presente capítulo que son un complemento a la alta disponibilidad de los sistemas: agentes, la seguridad, las actualizaciones del sistema y la documentación que son elementos clave para la disponibilidad y que hacen posible lograrla en un cien porciento a la vista de los usuarios finales quienes como instancia productiva son aquellos que determinan de algún modo y de acuerdo a su actividad la efectividad en el servicio proporcionado por los sistemas.

4.1 AGENTES

Los agentes son programas (combinaciones de archivos binarios, librerias y scripts) que funcionan como intermediario entre los recursos de un sistema y el software de un cluster.

Las propiedades de un agente son:

- Sólo un demonio de agente corren un sistema para cada tipo de recurso configurado.
- Un agente corre en una sola operación para un recurso a la vez.
- Los agentes son multihilo así que las operaciones pueden ser ejecutadas en paralelo en múltiples recursos del mismo tipo simultáneamente.
- Un recurso en cluster no podrá ser manejado sin agente.

Las tareas comunes de un agente son:

- Periódicamente monitorear los recursos y enviar información de estado hacia el software de cluster.
- Pone en línea los recursos cuando son requeridos por el software de cluster.
- Da de baja los recursos cuando se es requerido
- Reinicia los recursos cuando éstos presentan alguna falla
- Envía mensajes hacia el software de cluster y a las bitácoras del agente cuando se detectan errores.

Los agentes toman un conjunto de parámetros de entrada desde el programa del cluster, estos parámetros son los valores especificados por los atributos de los recursos. El agente pasa los parámetros de entrada como argumentos para los programas que desempeñan alguna tarea requerida con respecto al manejo del recurso. Los programas que son llamados por el agente son referidos como puntos de entrada en el agente, un punto de entrada es llamado cuando un evento ocurre y el agente toma alguna acción.

Los puntos de entrada comunes en muchos de los agentes son:

- Online: levanta o crea el recurso.
- Offline: Para o borra el recurso.
- Monitor: Determina el estado del recurso.
- Clean: Hace tareas definidas despues de que un recurso fallo.

Específicamente los agentes de Veritas se dividen en incluidos y no incluidos (bundled y unbundled), esto significa que con la adquisición del software de cluster es posible adecuar mediante parámetros lo que se quiere que un agente lleve a cabo a fin de obtener una alta disponibilidad de un recurso determinado.

Los agentes incluidos son aquellos que se instalan por default con la instalacion del software de cluster, gestionan recursos que por default son controlados por el sistema operativo y estan divididos de la siguiente forma:

Networking
- IPAgents
- IPMultiNIC
- IPMultiNICB
- MultiNICA
- MultiNICB
- NIC

Basic Storage Agents
- Disk
- DiskGroup
- DiskReservation
- Mount
- NFS
- Share
- Volume

Application Control Agents
- Application
- Process

VCS Infrastructure and Support Agents
- NotifierMngr
- Phantom
- Proxy
- ServiceGroupHB
- VRTSWebApp
- ElifNone
- FileNone
- FileOnOff
- FileOnOnly

Los agentes no incluidos se compran de manera separada al software de cluster y se destinan a aplicaciones que por lo regular no son de la misma marca de la plataforma de sistema operativo y en consecuencia éste no mantiene un control directo sobre la aplicación, los agentes no incluidos existentes en el mercado para Veritas son:

- DB2 UDB
- Informix
- iPlanet

- NetApp
- NetBackup
- Oracle
- Sybase

4.2. ACTUALIZACIONES

La naturaleza del cambio mediatiza todo el trabajo del software. El cambio es inevitable en la construcción de sistemas; por ello debemos desarrollar mecanismos de evaluación, control e implementación de modificaciones. El mantenimiento del software es, por supuesto, mucho más que una "corrección de errores". Podemos describir el mantenimiento describiendo las cuatro actividades que se llevan a cabo tras distribuir un programa.

La primera actividad de mantenimiento es debida a que no es razonable asumir que las pruebas del software hayan descubierto todos los errores latentes del sistema. Durante el uso de cualquier programa, se encontrarán errores. El proceso que incluye el diagnostico y la corrección de uno o más errores se denomina mantenimiento correctivo.

La segunda actividad que contribuye a la definición de mantenimiento se produce por el rápido cambio inherente a cualquier aspecto de la informática. Se anuncian nuevas generaciones de hardware en ciclos de unos 24 meses, se mejoran o modifican los equipos periféricos y otros elementos del sistema. Por otro lado, la vida útil del software puede fácilmente superar los diez años. Por tanto, el mantenimiento adaptativo –una actividad que modifica el software para que interaccione adecuadamente con su entorno cambiante- es tan necesaria como usual.

La tercera actividad que se puede aplicar a la definición de mantenimiento se produce cuando un paquete de software tiene éxito. A medida que se usa el software, se reciben de los usuarios recomendaciones sobre nuevas posibilidades, sobre modificaciones de funciones ya existentes y sobre mejoras en general. Para satisfacer estas peticiones, se lleva a cabo el mantenimiento perfectivo. Esta actividad contabiliza la mayor cantidad de esfuerzo empleado en el mantenimiento del software.

Si existe una completa configuración del software, la tarea de mantenimiento comienza con una evaluación de la documentación del diseño. Se determina las importantes características estructurales, de rendimiento y de interfaz del software. Se estudia el impacto de las correcciones o modificaciones requeridas y se traza un plan de actuación. Se modifica el diseño y se revisa.

Un programa con un flujo de control con "módulos" de 2000 líneas, con tres comentarios significativos por cada 9000 sentencias fuente y sin ninguna otra documentación debe ser modificado para acomodar los cambios en los requisitos de los usuarios. Tenemos las siguientes opciones:

1. Adentramos por las modificaciones, "luchando" con el diseño implícito y con el código fuente para implementar los cambios oportunos.

2. Intentar comprender a grandes rasgos el funcionamiento interno del programa, en un esfuerzo de llevar a cabo las modificaciones de forma más efectiva.
3. Rediseñar, recodificar y probar las partes del software que requieren modificaciones, aplicando un enfoque de ingeniería del software a todos los segmentos revisados.
4. Rediseñar, recodificar y probar completamente el programa.

No existe una única opción valida. Las circunstancias pueden dictar la primera opción incluso cuando fueran más deseables las otras. En lugar de esperar a recibir una petición de mantenimiento, la organización de desarrollo o de mantenimiento selecciona un programa que (1) vaya a estar en uso durante una determinada serie de años; (2) esté siendo usado correctamente y (3) pueda necesitar en un futuro cercano modificaciones o mejoras importantes. En este caso, se aplica la opción 2,3 o 4.

A primera vista, la sugerencia de redesarrollar un programa cuando existe una versión operativa puede parecer bastante extravagante. Antes de establecer un juicio, debemos considerar los siguientes puntos:

1. El costo de mantener una línea de código fuente puede ser de entre 20 y 40 veces el costo de desarrollo inicial de esta línea.
2. Rediseñar la arquitectura del software (estructura de datos y/o de programas) con metodologías modernas de diseño puede facilitar enormemente un futuro mantenimiento.
3. Debido a la existencia de un prototipo de software, la productividad de desarrollo puede ser mucho mayor que la media.
4. El usuario ya tiene experiencia con el software. Por tanto, es más fácil descubrir nuevos requisitos y la dirección del cambio.
5. Tras terminar el mantenimiento preventivo existirá una completa configuración del software.

Por lo que podemos concluir que cuando una organización de desarrollo de software vende productos de software, el mantenimiento preventivo se ve como "nuevas versiones" del programa.

4.2.1 Firmware

El concepto de microprogramación se atribuye al profesor Maurice Wikes (1951) que es el que presentó los conceptos que forman la base de las técnicas de microprogramación actuales, aunque hasta los años 60 comenzó a implementarse en las computadoras. La microprogramación es la escritura de programas que llevan a cabo la función de la unidad de control mediante la descripción de sus fases como una secuencia de operaciones elementales. La microprogramación introduce un sustrato de programación bajo el lenguaje máquina de la computadora, que posibilita la definición de sus instrucciones y en consecuencia la modificación del funcionamiento de la computadora a nivel funcional básico.

Es una parte integral de la computadora de gran importancia al considerar la seguridad y rendimiento de los sistemas operativos y consiste en un conjunto de pequeños programas secuenciales, microprogramas, formados por microinstrucciones, siendo éstas las que realmente interpreta el hardware.

Los microprogramas están formados por microinstrucciones que son mucho más elementales, en naturaleza y alcance, que las instrucciones en lenguaje máquina y se ejecutan en una memoria especial de alta velocidad llamada almacenamiento o memoria de control.

Las microinstrucciones se pueden clasificar en horizontales y verticales. Las verticales proporcionan un control explicito de las funciones en puntos determinados dentro de la unidad central de proceso. Por ejemplo, un bit determinado en la microinstrucción exigirá la puesta a cero de un registro específico en un tiempo específico de reloj. Generalmente, las microinstrucciones verticales contienen campos codificados y describen operaciones a ser realizadas por ciertos elementos de la unidad de control, la unidad aritmética y lógica, y por las funciones y destinos de información que pasa entre estas unidades.

Su ejecución es muy parecida a la de las instrucciones en lenguaje máquina. Una microinstrucción vertical típica especifica el movimiento de un campo entre registros. El microcódigo horizontal es bastante diferente. Cada instrucción necesita muchos más bits para especificar la operación del movimiento paralelo de datos entre muchos o todos los registros de datos de la unidad de control. Estas microinstrucciones son más poderosas que las verticales, pero los programas resultantes son también más difíciles de codificar y depurar.
Una de las aplicaciones de la microprogramación es la microdiagnosis, dado que los microprogramas tienen acceso directo al hardware, es posible realizar una detección y corrección de errores más extensa; es decir, realizar estas funciones con más detalle. Por tanto, introduciendo la microdiagnosis en los sistemas se consigue que sean más fiables.
De lo antes expuesto podemos decir que el firmware consiste en programas ejecutables almacenados en un circuito integrado programable, en lugar de estar incluido en el software básico, cada componente de la computadora puede tener un circuito programable.

4.2.2 Actualización del microcódigo

Todos los fabricantes de hardware realizan mejoras continuas (al menos una vez al año) de sus entornos operativos para logra la máxima fiabilidad e incorporar las mejoras en funcionalidades y servicios que permitan aumentar la disponibilidad del sistema para las transacciones en línea. El objetivo de la estrategia de actualización del microcódigo es ayudar a implantar fácilmente estas nuevas características y mejoras y participar en el proceso de mejora continua.

4.2.3 Estrategia de implantación del microcódigo

a) Todos los fabricantes de hardware desarrollan de forma periódica nuevos niveles de microcódigo, que aportan resolución de problemas, mejoras en los servicios o cambios en las características ya existentes. Para cada nuevo nivel de microcódigo existe un procedimiento de evaluación de la implantación, que describe el efecto de las actualizaciones sobre el entorno operativo.
b) Cada nuevo nivel de microcódigo se implanta después de pasar por un exhaustivo proceso de pruebas y evaluación que garantiza la compatibilidad con todos los niveles de revisión de hardware, así como la compatibilidad de determinadas funcionalidades.
c) Los técnicos revisan los nuevos niveles de microcódigo y, basándose en el tipo y complejidad de cada revisión, determinan cuáles de ellas deben designarse como notas de revisión. Estas notas pueden referirse a la información de anuncio de la revisión a la presentación de plataformas y revisiones.

4.3 SEGURIDAD

La seguridad de un sistema de cómputo se da cuando hay confianza en éste, el comportamiento del software y hardware es el esperado y la información almacenada es accesible e inalterable. A finales de la década de los ochenta los sistemas fueron punto de ataque de programas destinados a dañarlos, haciendo borrado masivo de datos, alterando información y provocando caídas de sistemas en empresas de servicios de reservación, instituciones bancarias, entre otras.

Fue imperante trabajar en políticas de seguridad para acceso a los centros de cómputo, proteger los huecos lógicos de seguridad de los sistemas operativos y hasta la fecha no ha sido posible evitar sean vulnerables a un ataque.

La seguridad en cómputo es algo que lleva a cabo la seguridad no solo de las computadoras sino todo aquello asociado a estas; el edificio donde se encuentren, las terminales, impresoras, cableado, cintas de respaldo y lo más importante los datos que en su conjunto constituyen la información y que reside en los sistemas de cómputo, también esto es llamado seguridad de la información. El propósito de la seguridad en cómputo es el proteger la información de todo intruso que acceda para alterarla, también el de establecer políticas que hagan posible el proteger a los sistemas de peligros inmediatos como el hacer mal un respaldo, derramar algún liquido, la compartición de claves, etc.

Hay tres aspectos importantes de la seguridad en cómputo y que se refieren a los datos: confidencialidad, integridad y disponibilidad.

Confidencialidad.

Un sistema de cómputo seguro no deberá presentar información a quien no esta autorizado a verla. En ambientes de negocios la confidencialidad asegura la protección de información privada tal como la nomina de una empresa, datos corporativos esenciales, memorandos internos y documentos de estrategia competitiva, este aspecto de seguridad es considerado por algunos como el punto más importante de seguridad, pues es de vital importancia cuidar la confidencialidad de un solo modo: dar acceso a la información privada solo a aquellos que estén autorizados.

Integridad.

Un sistema de cómputo seguro debe mantener continuamente la integridad de la información que se almacena, la integridad se refiere a que el sistema no debe corromper la información ni dejar que se modifique o se haga algún cambio accidental o no autorizado. En el ambiente financiero esta característica de la seguridad es fundamental pues tal vez la confidencialidad sea menos importante que la integridad de una transferencia de fondos monetarios y asegurar la exactitud de la transacción.

Disponibilidad.

Un sistema seguro debe mantener la información disponible a sus usuarios, la disponibilidad se refiere a que el software y el hardware se mantendrán trabajando de manera eficiente de tal forma que el sistema pueda recuperarse de manera rápida y completa si un desastre ocurre.

Lo opuesto a la disponibilidad es la negación de un servicio y esto significa que los usuarios del sistema no puedan obtener los recursos del sistema que ellos necesitan.

4.3.1. Tipos de seguridad.

4.3.1.1 Seguridad lógica.

La seguridad lógica comienza desde que se asignan usuarios de los sistemas y son autenticados mediante sus claves correspondientes, muchas veces éstas claves se descuidan compartiéndolas, escribiéndolas en papel, cambiándolas por otras claves que sean más sencillas de recordar, haciendo vulnerable a la información de los usuarios y al sistema mismo si se trata de un usuario con los atributos de administrador y que afecta directamente a la disponibilidad del sistema.

Es importante tener en cuenta las siguientes recomendaciones:

- No compartir las claves de usuarios y mucho menos la clave del usuario administrador.
- Hacer de la clave algo difícil de descifrar haciéndola de ocho caracteres y combinándola con caracteres especiales y alfanuméricos.
- No escribir la clave en muchas partes.

Algo de vital importancia en la seguridad lógica son las bitácoras pues los sistemas de cómputo y las diferentes aplicaciones generan una parte considerable de registros históricos que hacen posible el análisis de lo que pasa ante un incidente de seguridad. Es posible el detectar un comportamiento inusual del sistema, recavar información para resolver problemas, ser usado como una evidencia legal ante un intruso detectado e identificado.

La información generada en una bitácora de sistema típicamente contiene:

- Fecha y hora
- Direcciones IP de origen y destino
- Direcciones IP que generan bitácoras de intercambio de paquetes
- Usuarios
- Errores

Actualmente el hacer la revisión de bitácoras es una tarea tediosa y que hay administradores de sistemas que no lo llevan a cabo, siendo que esta actividad puede evitar horas de inactividad.

Afortunadamente hay herramientas tales como Logcheck, SWATCH, Patrol de BMC entre otros que hacen el análisis basado en patrones y parámetros ajustables a las necesidades de cada administrador de sistemas.

Otro punto importante son las actualizaciones de los sistemas que siempre cuentan con algún hueco de seguridad ya sea en sistemas operativos o en alguna aplicación una vez liberada alguna versión de éstos normalmente no pasa mucho tiempo cuando se les descubre algún imperfecto y se generan parches de seguridad para cubrir ese hueco y mantener actualizado en ese sentido al sistema, sobre actualización se hablara en una próxima sección de este capitulo.

La seguridad lógica de un sistema también puede depender de otro sistema encargado de monitorear y restringir el acceso tal es el caso de los sistemas cortafuegos (firewalls) los cuales tienen la finalidad de ser una barrera lógica ante intentos de intrusión por medio de restricciones y autenticaciones mas dedicadas y evitar cualquier acción que pueda alterar la seguridad de los sistemas resguardados dentro de la barrera de un cortafuegos.

4.3.1.2 Seguridad física.

La seguridad física de un centro de cómputo considera la ubicación y la disposición de este tomando en cuenta las características del equipo, su valor e importancia, lo ideal seria ubicarlo lejos del transito terrestre y aéreo debido a que las ondas electromagnéticas de radares y microondas puede afectar el desempeño de lo sistemas.

Hay que considerar también las características del terreno en cuanto a la humedad, el hundimiento del piso, si es zona sísmica, que se cuente con las líneas telefónicas suficientes, se tenga una instalación eléctrica adecuada con corriente regulada y tener antenas de comunicación propias para el centro de cómputo.

Hay también algunos factores dentro del centro de cómputo que hay que considerar tales como el piso falso a unos 40 centímetros aproximadamente, sellado hermético, nivelado topográfico, contar con tierra física, cubrir el cableado de alto y bajo voltaje, de telecomunicaciones y de señales para monitores.

Las paredes se recomiendan con pintura plástica lavable techo falso de plafón, una altura entre 2.70 a 3.30 metros, con puertas de acceso y salidas de emergencia así como también generadores de energía eléctrica fuera de la sala con la alimentación de la iluminación diferente a la que se tenga como alimentación para los equipos y el 25% debe ser de emergencia conectado a UPS.

Para el ambiente se recomienda el 99% de eficiencia de filtros sobre partículas de 3 micrones, seleccionar los filtros adecuados para otro tipo de contaminantes o partículas, aire de renovación y ventilación, equipos antivibraciones y contar con ductos lisos y sin desprendimientos de partículas.

El acondicionamiento de espacio y distribución del lugar debe ser acorde con:

- Especificaciones técnicas del equipo
- Áreas de cintas, discos archivos
- Evitar áreas con formas extrañas
- Preferencia a las formas rectangulares
- Consideraciones a futuro

Planos civiles y arquitectónicos:

- Hidráulicos
- Planta
- Memoria de cálculo
- Sanitario
- Líneas telefónicas
- Energía eléctrica

Control de acceso físico.

Se debe identificar el personal y a los visitantes, evaluar algún tipo de acceso magnético y tener vidrio reforzado sobro todo a los visitantes que tengan actividades de mantenimiento del lugar así como los de limpieza.

Tener por regla general y sin excepción la identificación plena de las personas que accedan al centro de datos.

4.4 DOCUMENTACIÓN

Documentación es un término para describir todas las instrucciones, programas y narrativos, esto es, casi cualquier escrito acerca del sistema. La documentación sirve para un sin número de propósitos. En un principio, durante el diseño, éste es el producto desarrollado por el equipo de diseño. Después de la puesta en marcha, es la base para realizar cambios en el sistema. La calidad de la documentación determina qué tan flexible es el departamento de servicio al responder a los requerimientos del usuario. La buena documentación sirve para reducir los conflictos entre los usuarios y el departamento de soporte, ya que los usuarios entenderán más fácilmente un sistema bien documentado. Una buena documentación significa que se dispone de una referencia adecuada cuando surjan los problemas, y ayudan a aprender cómo resolver los problemas con el sistema.
La documentación es una actividad a la cual debe dedicársele tiempo, si se desea que el sistema tenga éxito.
El líder del equipo de diseño debe estar consciente de los tipos de documentación necesaria y esforzarse para convencer al departamento de servicio para que lo prepare. Los usuarios del equipo de diseño pueden ayudar preparando la documentación de entrenamiento y la de referencia del usuario para el sistema.

4.4.1 Finalidad de la documentación

La documentación del sistema puede ahorrarle a la compañía miles de dólares. En algunas empresas, un solo individuo tiene en su cabeza todos los flujos de información. Pero si el individuo abandona la empresa, a ésta no le quedará más remedio que volver a estudiar todo el sistema o documentar o diseñar otro nuevo. A continuación se dan algunas de las razones por las cuales es importante una buena documentación:

1. Rotación de personal clave. Si el diseñador del sistema administrativo no ha documentado el sistema o su trabajo ha sido inadecuado o fragmentario, su sucesor deberá volver a estudiar el sistema para resolver los problemas o hacer modificaciones.
2. El sistema de información requerirá modificaciones para mejorar a ajustar a las condiciones cambiantes. Aun cuando no hay rotación de personal, existen escasas posibilidades de que los analistas de sistemas puedan recordar todos los detalles del sistema durante largo tiempo.
3. La creciente complejidad de los sistemas computacional requerirá documentación, de modo que los diseñadores originales no tendrán que familiarizarse con el equipo a medida que progresan los diseñadores originales no tendrán que familiarizarse con el equipo a medida que progresa el diseño del sistema.
4. La documentación revelará las características de un diseño deficiente y la falta de normas para imponer medidas correctivas.

4.4.2 Especificación y estandarización

Especificación

La especificación significa una descripción clara y suficientemente detallada. Al iniciarse el proceso del diseño se prepararan especificaciones de funcionamiento que describen los objetivos del sistema. Durante el proceso del diseño detallado, conviene formular especificaciones de diseño que describan los sistemas. Después se preparan especificaciones de operación, las cuales describen las funciones y actividades del personal encargado del sistema.

Estandarización

El empleo de procedimientos y documentación estandarizado proporciona la base de una comunicación clara y rápida, un adiestramiento mucho menos costoso de los analistas, la reducción de costos de almacenamiento y una evaluación del rendimiento de los analistas y del sistema.

Documentación de diseño

Durante el diseño, el propósito de la documentación es ayudar al control del proyecto proporcionando un registro de qué es lo que ha sido desarrollado y de qué es lo que ha sido cambiado. Es importante asegurase de que se consideren todas las partes del sistema y de que se notifique a todos los responsable de los componentes afectados. La documentación de diseño es una excelente base de datos para hacer estimaciones futuras sobre cuánto tomará el desarrollo del sistema.

Documentación de capacitación

La documentación de capacitación prepara para la conversión, instalación y utilización del sistema. La mayor parte de la información necesaria para entrenar puede desarrollarse a partir de la documentación del sistema analizado. La documentación de entrenamiento a los usuarios es utilizada para acortar la brecha entre los antiguos procedimientos y aquellos requeridos para el nuevo sistema. Esta documentación debe ser desarrollada por miembros del equipo de diseño del área usuaria en combinación con otros usuarios dentro de la organización.

Documentación de la operación

La sección de operación tiene que operar un sistema después de haber sido convertido. El grupo de operaciones necesita información sobre los procedimientos de operación normal y cómo responder los errores. Esta información se prepara mejor si lo hacen los analistas, y mucho de ello puede ser derivado de la documentación de diseño.

Documentación de referencia para el usuario

El último tipo de documentación que de debe desarrollarse es para que sirva de referencia al usuario después que se ha iniciado la operación del sistema. Esta información debe consultar al principio, cuando se tengan dudas o problemas. Si esta información es de buena calidad, los usuarios pueden contestarse sus propias preguntas sin necesidad de estar comunicándose con el departamento de servicio y por tanto se reduce los conflictos potenciales.
Afortunadamente, mucho de este material puede tomarse en forma directa de otra documentación, por ejemplo, las partes de procedimientos de los documentos de entrenamiento.
Se necesita un índice detallado para permitir que esta documentación sea una lista de condiciones de errores y acciones que deben tomarse en cada caso.

En este capítulo fue posible darse cuenta de la necesidad de elementos complementarios tales como los agentes para las aplicaciones y los elementos redundantes así como también la seguridad, las actualizaciones y una adecuada documentación que hacen a un sistema de cómputo confiable e integral en un ambiente controlado, es decir, con mejoras que hacen cumplir la disponibilidad del sistema y por consecuencia de los datos de una forma más cercana a la deseada.

5 ALTA DISPONIBILIDAD EN SERVIDORES REDUNDANTES

En éste capítulo se expondrán los elementos y las características que componen a un cluster de alta disponibilidad, una vez vistos los anteriores capítulos será posible observar de forma mas clara el esquema que se propone para la disponibilidad de los datos uniendo las ideas, elementos y características para lograrlo. Lo que a continuación se expone será en forma general lo último a contemplar en la presente tesis para dar paso a el procedimiento de instalación de un sistema de alta disponibilidad en el capítulo 6.

5.1 FALLAS EN SERVIDORES Y FAILOVER.

Puede tomar horas y hasta días para diagnosticar una falla, especialmente si ésta es de tipo intermitente y de las que difícilmente es resuelta, una vez que se diagnostica la falla se procede según el caso; si el problema es en hardware se debe obtener un reemplazo de la parte que falló llamando al proveedor del servicio para que lo haga, si el problema es en el software entonces debe ser obtenido el parche correspondiente para la aplicación o para el sistema operativo, asumiendo que el problema se arregló, el servidor debe ser reiniciado y la operación del sistema debe ser iniciada idealmente desde el momento del fallo con algún respaldo.

Si el servidor es crítico, las horas o días que se tome en arreglar alguna falla son inaceptables. Se podría poner las aplicaciones en un ambiente costoso de diseño implementado de tolerancia a fallas, el cual no ofrecerá la adecuada protección. Una solución práctica es poner dos o más servidores enlazados y si alguno falla el otro tomará su lugar.

Para garantizar la consistencia de datos y la rápida recuperación, los servidores deben conectarse a los mismos discos compartiéndolos, al ocurrir una falla la migración de servicios en otro servidor será activado esto es lo que se llama failover y de aquí en adelante se tomará con este nombre al proceso mencionado, el planteamiento original de este esquema considera que todo el sistema estará en un mismo lugar físico y así se tratará en este capítulo, existe el caso de faillover a través de redes más grandes y que normalmente se contemplan en casos de recuperación de desastres en una red de área extensa WAN y con replicación de datos implementada, esta última situación está fuera del caso práctico de la presente tesis.

El failover en el ambiente redundante debe de cumplir con los siguientes criterios:

Transparencia. El failover no debe tener más impacto a los usuarios que accedan al servidor que el que toma un reinicio del sistema, tal vez no sea notable y el usuario solamente tendrá que acceder de nuevo al sistema una vez que el servicio acabó de migrar y en algunos casos como servicios de archivos y web no se necesitará volver a acceder debido que en este esquema se presenta al exterior como un solo sistema para acceder y los discos compartidos en donde están las aplicaciones no se inactivarán.

Rapidez. El failover no deberá tomar más de cinco minutos, idealmente menos de dos, la mejor manera de lograr esta meta es tener el servidor secundario, el de respaldo, encendido, listo para tomar el lugar del primario y corriendo los procesos necesarios, si una reinicialización es requerida para el failover este tomará más tiempo pues normalmente el proceso que más tarda es la verificación de los sistemas de archivos.

Intervención manual mínima. Idealmente la intervención humana no debe ser requerida, el proceso de migración en su totalidad debe ser automatizado con ayuda del sistema de failover (Failover Management System), algunas aplicaciones requieren de alguna intervención manual para llevarlo a cabo pero esto no es deseable, idealmente el failover no deberá de requerir de una reinicialización del sistema.

Acceso de datos garantizada. Después de un failover, el servidor que toma el lugar del primario debe ver la misma copia de los datos críticos tal y como el servidor original lo hacía, así como también operar de la misma forma. La replicación de datos hacia otro servidor cuando los discos no son compartidos adiciona tanto riesgos como complejidad y no es aconsejable para servidores redundantes que se encuentren cercanos debido a los costos de implementación.

Los sistemas en una configuración de failover deben comunicarse entre ellos continuamente para así saber su estado, ésta comunicación en red privada es llamada heartbeat (latido), ésta comunicación será la que active un failover en caso de no haber respuesta del otro servidor, así como también de migrar los servicios de red en caso de alguna falla pasando el control de cluster de un servidor a otro.

Cuando ocurre un failover hay tres elementos críticos que se mueven del servidor primario al secundario o de respaldo:

1. Identidad de red. En un ambiente Ethernet[19] esto significa la dirección IP[20] del servidor y en algunos casos la dirección de hardware llamada dirección MAC[21].

2. Acceso a los discos compartidos. La tecnología del sistema operativo y particularmente el sistema de archivos esencialmente prohíbe el que múltiples servidores accedan los mismos discos al mismo tiempo por alguna razón, en un ambiente de discos compartidos el acceso lógico se restringirá a un servidor a la vez, así cuando el failover ocurra, el acceso a los discos por parte del servidor que fallo será denegado y asegurará el acceso al servidor de respaldo.

---

[19] Es una tecnología de red creada por Xerox y que en la actualidad es un estándar para redes de área local LAN.

[20] Internet Protocol por sus siglas en inglés se refiere a una dirección de cuatro octetos para identficar un sistema en una red.

[21] Media Acces Control por sus siglas en inglés es una dirección física o de hardware, hexadecimal y unica de red con las que los sistemas de cómputo salen de fábrica.

3. Conjunto de procesos. Una vez que los discos han sido migrados al servidor de respaldo todos los procesos asociados con los datos deben ser actualizados acorde a este servidor y la consistencia de datos debe ser asegurada desde la perspectiva de la aplicación.

La colección de estos elementos es comúnmente llamado grupo de servicio, si los servidores tienen múltiples grupos de servicio estos deben ser totalmente independientes para que puedan estar en cualquier sistema dentro de la configuración.

5.2 EVALUACIÓN DE APLICACIONES CENTRALIZADAS.

Al hablar de centralización normalmente se contempla a los servidores como la fuente principal de los servicios, además de que éstos serán en cantidad menor que los sistemas o procesos cliente que los accedan y el trabajo que realicen dependerá en todo momento de la disponibilidad de las aplicaciones que en los servidores reside, este entorno es dominante en los sistemas actuales aun cuando el manejo de datos tienda a distribuirse, siempre habrá aplicaciones de manera centralizada para dar servicio.

Teniendo en cuenta el entorno mencionado, la disponibilidad obviamente será afectada por fallas junto con el tiempo de recuperación el cual se divide en la identificación de la falla, el proceso de corrección y el levantamiento del sistema. En un ambiente redundante es posible reducir el tiempo de espera para volver a operar, solamente habrá que esperar a que el sistema de failover implementado actúe automáticamente y active la migración de servicios y el control de estos de un servidor a otro.

Uno de los aspectos más interesantes de trabajar con sistemas en failover es la nueva forma que se tiene en mente acerca del par de servidores con sus servicios y recursos, normalmente se piensa en una computadora como una caja con una identidad de red y que corre en ésta una o más aplicaciones localmente o a través de la red.

La colección de servidores y los discos compartidos en una configuración de failover se llama cluster, se ha usado este término y se usará para referirse a éste conjunto, el componente base no es el servidor sino la aplicación asociada con identidades de red, almacenamiento y recursos de cómputo. La computadora es meramente la aplicación, la red o el mecanismo que da el servicio.

La computadora en si, es el componente de menos interés en un mecanismo de servicio de aplicaciones, cualquier computadora que corra el sistema operativo correcto y que tiene el hardware adecuado puede tomar el lugar de otro que falle y continuar operando. Un sistema de failover da esta facilidad de hacer el cambio y de tener el entorno mencionado.

En este ambiente una dirección IP no conecta a un nombre en particular de servidor sino más bien conecta a un servicio en particular usando un nombre, el nombre y la dirección IP residirá en una máquina que es parte de un par redundante y no importará que servidor de éste par esté dando el servicio, el cluster será para el exterior una sola dirección y un solo nombre.

Para ejemplificar lo anterior se puede tomar el caso de un cambio de dirección de una familia completa en una misma localidad; para los familiares y amistades comunicarse y encontrar a las personas basta con marcar su número telefónico que representa la identidad exterior para quienes quieran comunicarse con algún elemento de la familia, el día que deciden cambiarse de domicilio dentro de una misma localidad es posible que conserven el mismo número de teléfono y las

personas seguirán marcando al mismo número sin importar el cambio de domicilio de la familia en cuestión.

Considerando ahora el par o cluster en configuración failover que actúa como una sola caja dando un servicio determinado, los usuarios se conectan a una dirección de red, presentan los datos a la red y obtienen una respuesta sin importar quien dentro de la caja esté dando el servicio pues la identidad de red del cluster siempre será la misma.

5.3 MANEJO DEL FAILOVER Y SUS REQUERIMIENTOS.

Como se ha visto, un par de servidores en configuración failover requiere más que estén ubicados uno cerca del otro, a continuación se examinarán los componentes necesarios que hacen posible la configuración:

Servidores. Se necesitan al menos dos servidores un primario y otro secundario, la aplicación crítica migrará de una a otro servidor cuando una falla ocurra, los servidores deberán tener el mismo sistema operativo, los mismos parches instalados, soportar los mismos archivos binarios ejecutables, estar configurados de la misma forma y tener la misma arquitectura. En las figuras 5.1 y 5.2 s presenta el servidor Sun modelo V880 que usaremos visto de frente y el anverso respectivamente.

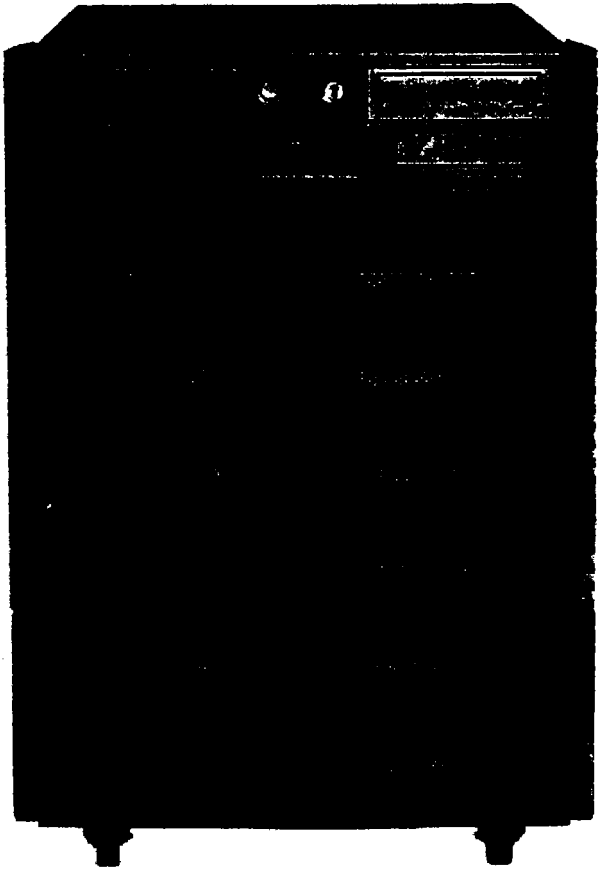


Figura 5.1
Servidor Sun V880

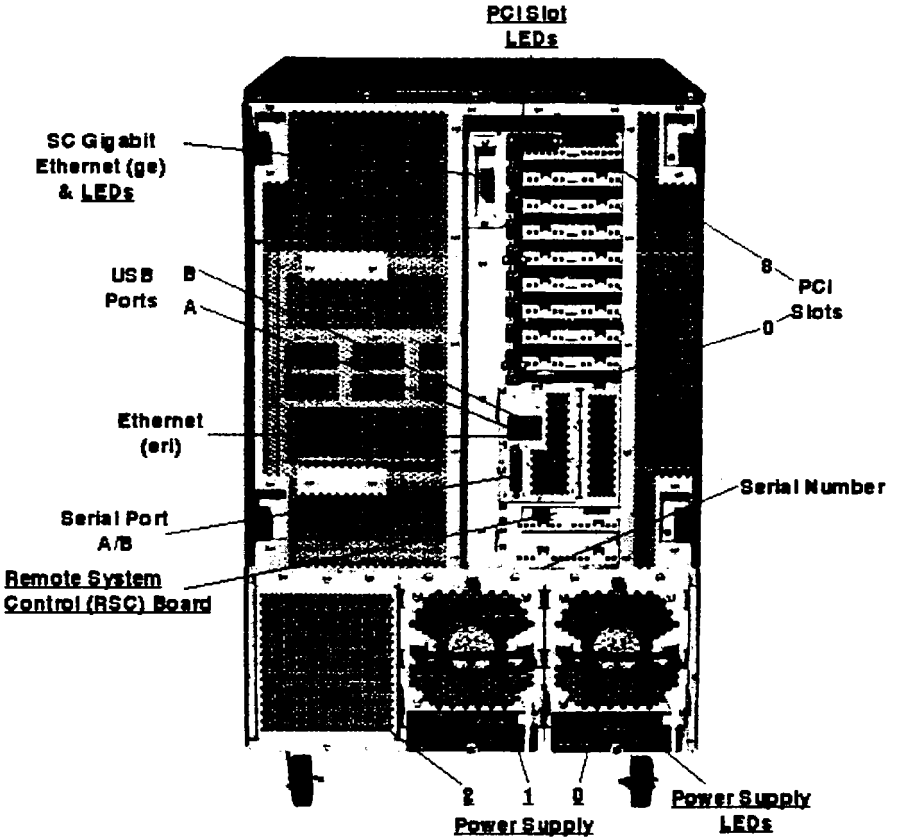


Figura 5.2
Anverso del Servidor Sun V880

Conexiones de red. Hay tres tipos de conexión de red necesarias, la más básica y recomendada es la conexión de red privada de heartbeat que hará posible el monitoreo entre los servidores y hacer la activación del failover en caso de ser necesario, para tal efecto usaremos una tarjeta de red qfe Quad Fast Ethernet que se ilustra en la figura 5.3, la segunda es la conexión de red pública o de servicio a la cual se conectarán los usuarios que quieran acceder a las aplicaciones, la cual se ilustra en la fgura 5.2 indicada como Ethernet (eri) y la tercera es la conexión de red, que el administrador del sistema tendrá para garantizar el acceso en caso de falla de las dos conexiones mencionadas para cada servidor.

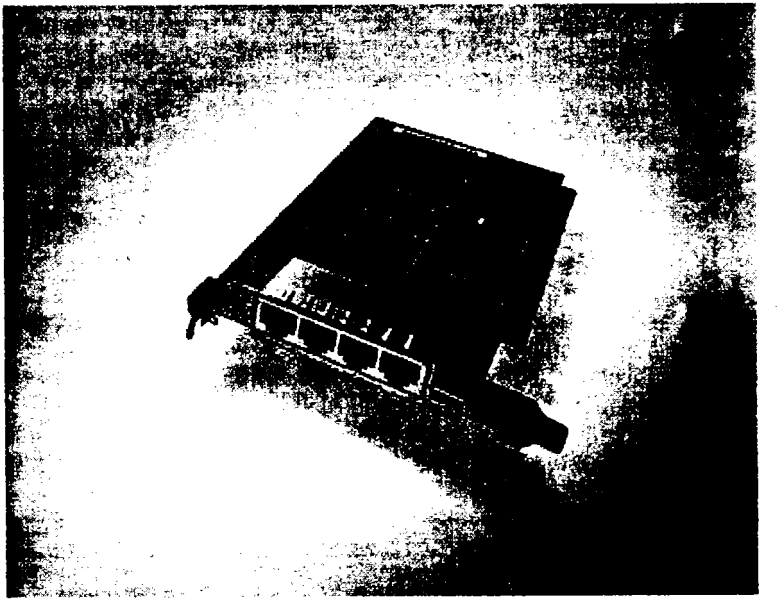


Figura 5.3
Tarjeta de red Quad Fast Ethernet

Discos. Hay dos tipos de discos requeridos en un ambiente de failover. Los incompartidos e internos también llamados discos privados que contienen el sistema operativo incluyendo el sistema de failover; se ilustran los discos internos del servidor en la figura 5.4 y los compartidos donde residen los datos de la aplicación crítica y donde se migrará de un servidor a otro cuando un failover ocurra, deben ser accesibles a los servidores, uno a la vez, este tipo de discos es conocido como discos públicos los cuales para el presente trabajo será el arreglo de discos modelo 5200 de la marca Sun como se ilustra en la figura 5.5

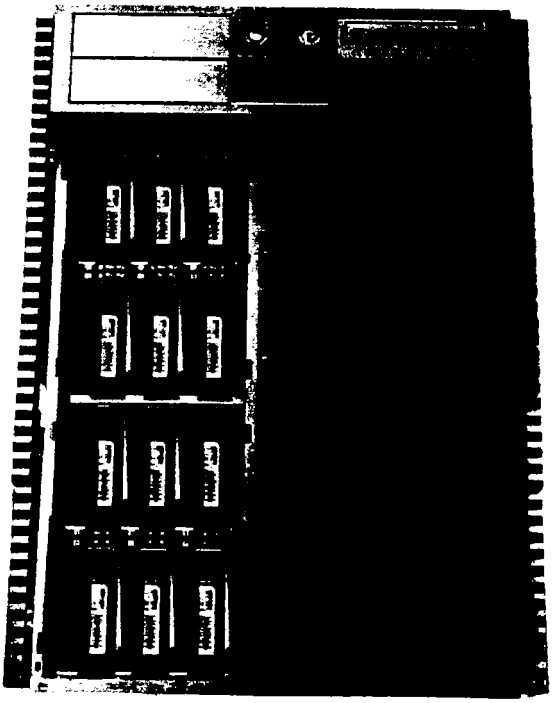


Figura 5.4
Discos internos del servidor

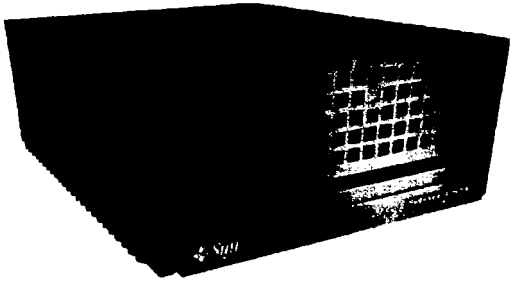


Figura 5.5
Arreglo de discos StorEdge A5220

Como se mencionó anteriormente hay configuraciones de failover los cuales no comparten ningún componente haciendo necesaria la replicación de datos, este tipo de configuración adiciona complejidad y dependencia en la red, es recomendable en failover para WAN.

Portabilidad de aplicación. Uno de los requerimientos en un ambiente failover es que las aplicaciones críticas puedan correr en ambos servidores, en uno a la vez si esto no es posible tampoco será la alta disponibilidad y para esto será necesario tener que comprar dos licencias, una para cada servidor, en estos casos es cuando se hace presente el alto costo de obtener un esquema redundante.

Sin puntos de falla. Esta idea es general y se refiere a cada elemento del par de servidores. Si hay un componente en el par configurado para failover que cause que el sistema se venga abajo entonces no se tendrá alta disponibilidad.

5.4 FALLAS EN SERVIDORES INCOMPATIBLES.

La idea fundamental como se ha venido planteando es que cuando se cuente con un par de servidores a configurar para hacer un cluster con failover sean de la misma plataforma y corran el mismo sistema operativo con la misma liberación de parches instalada, los servidores deben ofrecer el mismo desempeño a través de la memoria y velocidad de procesador, es decir, que los dos servidores idealmente deben ser completamente idénticos.

Si por alguna razón un servidor es menos rápido que el otro una vez que un failover ocurra será notorio para los usuarios debido a la pérdida de desempeño de las aplicaciones, esto obligará al administrador del sistema trabajar el tiempo mayormente posible con el servidor más rápido.

Algunos proveedores liberan al mercado variaciones de sus procesadores los cuales tienen pequeñas modificaciones del original, cada modelo de procesador tiene particularidades en cuanto a las instrucciones para diferentes optimizaciones que la aplicaciones requieren, algunas requieren de cambios particulares en cuanto a la arquitectura del kernel del sistema operativo.

Un ejemplo de SUN está en su familia de procesadores de la familia SPARC que tienen diferentes arquitecturas de kernel, requiriendo cada vez menos variaciones en el sistema operativo SunOS o Solaris, el servidor V880 usa el modelo de procesador UltraSPARC III el cual se ilustra en la figura 5.6 y usaremos la versión 9 de Solares el cual se presenta en la figura 5.7.

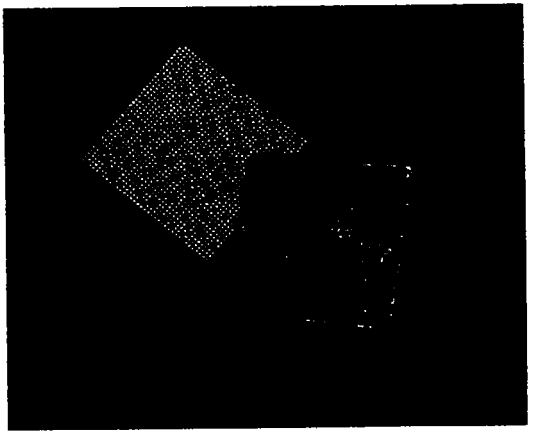


Figura 5.6
Procesador UltraSPARC III

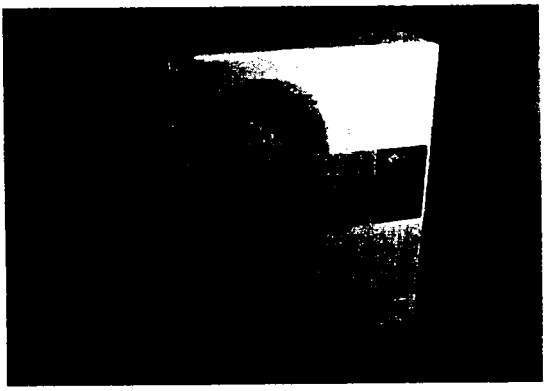


Figura 5.7
Sistema operativo Solaris 9

Cuando un procesador soporta modos especiales de acceso a memoria o una rápida copia de bloques de datos, las aplicaciones tales como los manejadores de bases de datos tomarán ventaja de esto, así que se debe asegurar de tener el mismo procesador en los servidores de un cluster.

Es asombroso oír sugerencias de pares en failover entre un servidor Unix y otro con Windows NT para lograr un ahorro considerable a la empresa debido a que son los servidores con que se cuenta, sin embargo, la gente que busca esto realmente no obtiene un ambiente de failover y mucho menos de alta disponibilidad.

Se tienen que considerar los siguientes puntos a fin de tener un failover entre servidores incompatibles:

Sistema de failover. Los dos sistemas deben de correr la misma versión de software de failover en nuestro caso Veritas Cluster Server que se ilustra en la figura 5.8, se pueden manejar variaciones de hardware para cada servidor tales como la arquitectura de kernel y cuando se intenta adicionar un servidor diferente se tendrán que correr riesgos que tal vez no se reflejen de inmediato, peor aun que pueda verse hasta que sea necesario un failover y el sistema de failover no trabaje correctamente.

Figura 5.8
Veritas Cluster Server

Interfaz de red. La red de heartbeat se vuelve mucho más complicada cuando las tarjetas de red son de diferente tipo y si las tecnologías de red son incompatibles se necesitará un puente de red u otro hardware para hacer posible su comunicación en la red privada que estos formen y por resultado un gasto no contemplado.

Discos. Los discos deben de ser compatibles físicamente para los dos servidores y tal vez se necesitará una tarjeta adicional en el mismo bus para hacer el arreglo de discos en uno u otro servidor.

Administración. Esta es una situación bastante difícil, un administrador de sistema requerirá los conocimientos de los dos diferentes sistemas operativos y ambientes de hardware y se requerirá de más tiempo para administrar dos sistemas diferentes y mantenerlos, además de que el trabajo hecho durante el día en un servidor necesitará ser transportado y traducido a fin de que el otro sistema pueda trabajar con los datos correctamente, éste trabajo representa hacer doblemente el trabajo a fin de mantener los datos correctamente en uno y otro servidor.

Soporte. Cuando hay problemas se tendrá que contactar a dos diferentes soportes los cuales verán el conflicto dentro de una configuración que no han probado y por resultado sólo harán el soporte del lado que les corresponde y en el peor de los casos no se harán cargo debido a que se tiene configurado algo que ellos no han probado ni dan soporte, también puede ocurrir un problema totalmente desconocido por los dos proveedores de soporte y estarán culpándose de la falla el uno al otro mientras el sistema está abajo. Actualmente hay un arreglo de disco con limitadas combinaciones para tener un failover con diferentes servidores, los cuales son los SAN (red con área de almacenamiento) y comparten los recursos que en ellos estén a diferentes sistemas.

5.5 SERVICIOS DE RED DEDICADOS.

Tres diferentes tipos de redes se pueden encontrar en una configuración failover: el heartbeat, el de producción o servicio y el de administración. Algunos sistemas operativos soportan interfaces virtuales de red en donde más de una dirección IP puede configurarse en la misma tarjeta de interfaz de red NIC.

Redes heartbeat. Estas redes son el medio por el cual los sistemas configurados en failover pueden comunicarse entre ellos, fundamentalmente los sistemas simplemente intercambian paquetes los cuales informan su estado o comandos de un servidor que direcciona al otro a tomar alguna acción.

El heartbeat debe trabajar a través de dos redes dedicadas, paralelas y no costosas en una configuración simple de dos nodos conectados por un par de cables de par trenzado cruzado, diez megabits por segundo es una velocidad suficiente para esta red podiendose usar el cableado Ethernet 10 Base-T, los paquetes suelen ser bastante pequeños y relativamente infrecuentes si se tiene el estándar de interfaces de cien megabits por segundo se puede usar un cable cruzado Fast Ethernet, realmente la velocidad no es algo de gran cuidado si se está dentro de las velocidades Ethernet.

Una red pequeña para heartbeat pueden trabajar también con cables regulares de red y concentradores de red pero esto es recomendable para clusters de más de un nodo, también hay otras opciones de construir una red de heartbeat como es usando linea serial usando protocolo PPP o SLIP a través de esta; aunque requieren de más configuración y cuidado que los enlaces Ethernet y por ende es menos apropiado para un ambiente crítico. Algunos sistemas operativos soportan más de una dirección IP en una sola tarjeta siendo esto útil para el funcionamiento del sistema de failover a través del heartbeat con direcciones virtuales y la real es usada para las conexiones que vienen del exterior hacia la interfaz de red de producción o servicio que a continuación se explicará.

Red de producción o servicio. Para proveer el servicio para el cual fue implementado un sistema, el par de servidores necesita ser conectado al menos a un servicio de red público y esta red debe ser la que aloje la conexión de todo aquel cliente que quiera obtener las aplicaciones críticas de los servidores, la red pública tendrá como un única cara visible a la dirección IP del par de servidores y la cual será la dirección externa para obtener un servicio del sistema en cluster.

Se tendrán también diferencias para la conexión a los servidores mientras que para un administrador será por su nombre real y físico del servidor, para un cliente podrá ser por éste o por el nombre del servicio que proporcione, es decir el nombre lógico.

Red de administración. En algunas configuraciones de failover el servidor de respaldo se inicializará sin una conexión de red aparte de las que se tiene con su par de heartbeat, esto tiene un mal efecto pues los servicios que se dan como el de nombres, correo electrónico o impresión necesitarán ser deshabilitados antes

de que la máquina inicialice y sólo levantar estos servicios cuando el servidor reciba un failover.

La mejor manera de solucionar lo anterior es configurar a los servidores con una tarjeta adicional para conectarse a la red pública, a esta conexión se le llama interfaz administrativa debido a que es solamente para propósito de garantizar la conexión al servidor en particular para que el administrador del sistema pueda acceder en caso de fallas.

5.6 DISCOS COMO APOYO PARA LA REDUNDANCIA

En una configuración failover hay dos tipos de discos: uno contiene información para inicialización, del sistema operativo y del sistema, estos discos están dedicados a sólo un servidor y por tanto son privados e independientes del segundo tipo de discos los cuales son compartidos los cuales son accesados por los dos servidores y contiene los datos críticos necesarios para las aplicaciones y por los cuales los servidores son de alta disponibilidad.

Discos privados. Estos discos son generalmente ubicados dentro de cada servidor, sin embargo, esto no es un requerimiento absoluto de hecho es mejor ubicarlos externamente de este modo si un servidor falla no será necesario extraer los discos del servidor siendo menos el tiempo que se lleve en una reparación.

Estos discos tienen el sistema operativo, la identidad del sistema, paginación y particularmente el sistema de failover para que pueda activarse en el momento de inicialización del sistema debe estar en los discos privados. Por definición estos discos no deben ser compartidos, sólo un servidor puede ver los datos en estos discos.

El contenido de estos discos debe ser espejeado, es decir, implementar RAID 1 adicionando mejor desempeño y protección al sistema operativo, así como también llevar a cabo una sincronización en el control de cambios para tener los mismos datos entre los dos servidores y se lleve a cabo el failover sin ningún contratiempo a causa de alguna inconsistencia como puede ser el no espejear el espacio de paginación ya que el no hacerlo resulta en un riesgo para las aplicaciones ya que de ocurrir algún fallo en la partición reservada para paginación no habrá forma de evitar un tiempo abajo y por ende un failover.

Discos compartidos. Este tipo de discos son los que contienen los datos críticos, los sistemas en cluster deben tener acceso físico a éstos, aunque es critico que solo sea un sistema a la vez y esto esté controlado. Si los dos sistemas tratan de escribir en los discos compartidos al mismo tiempo la corrupción de datos es inevitable.

Hay dos esquemas de discos compartidos: uno es comúnmente llamado dual hosting, en este modelo los servidores, típicamente dos, están físicamente conectados al mismo arreglo de discos al mismo tiempo, el acceso es gestionado por el software de cluster instalado en los sistemas, así cuando un failover ocurre el acceso a los discos compartidos para escritura esta garantizado para uno de los dos sistemas.

El otro método es llamado shared nothing, en este modelo los datos son replicados a través de la red, posiblemente a través de la red de heartbeat o por otra red privada paralela entre lo servidores, éste es un modelo mas complejo y costoso pues requiere de una red dedicada y funcional a distancias considerables para hacer posible la alta disponibilidad en recuperación de desastres y asegurar la replicación de datos con un mínimo de diferencia entre los servidores, en el

presente trabajo se utilizará el manejador de volúmenes en los discos duros compartidos Veritas Veritas Volume Manager software que se ilustra en la figura 5.9.

Figura 5.9
Veritas Volume Manager

En este capítulo vimos los requerimientos y elementos necesarios para armar un cluster de alta disponibilidad, lo que se usará tanto en hardware y software y ciertos lineamientos tales como el no combinar sistemas operativos no porque no fuese posible el implementarlos sino más bien por el hecho de que se tendrán reacciones impredecibles por parte de una u otra computadora; si bien es cierto que los sistemas operativos UNIX son parecidos tales como Linux, HP-UX, AIX y Solaris tienen particularidades que los hacen diferentes y por lo tanto de cierta forma incompatibles al tratar de conjuntarlos en la implementación de un esquema de alta disponibilidad, se detallará en especificaciones e instalación del cluster en el siguiente capítulo.

6. IMPLEMENTACIÓN DE UN SISTEMA DE ALTA DISPONIBILIDAD (UNIX-SUN-SOLARIS) USANDO VERITAS CLUSTER.

Este es el último capítulo en el cual se hablará sobre la instalación de un cluster de alta disponibilidad, iniciando con las características y especificaciones técnicas de los servidores así como de los requerimientos tanto de software como de hardware, la instalación del sistema operativo y la configuración de la redundancia de los componentes del sistema, para tal efecto se describirá paso a paso la instalación del software que espejeará al sistema operativo así como también el manejador de volúmenes y del cluster mismo para el control de la disponibilidad.

6.1 CARACTERISTICAS DEL HARDWARE SELECCIONADO

6.1.1 Servidor

El servidor Sun Fire V880 es ideal para la ejecución de un amplio rango de aplicaciones, tales como Internet y bases de datos, así como comercio electrónico. La arquitectura equilibrada del sistema contribuye a garantizar un excelente tiempo de respuesta. Este servidor combina alto rendimiento y escalabilidad excepcional con un conjunto de funciones de confiabilidad, disponibilidad y mantenimiento.
El servidor tiene la capacidad de soportar hasta 8 procesadores UltraSparc III, con un máximo de 32 GB de memoria. El subsistema de discos interno está comunicado por un canal de fibra óptica que soporta hasta 12 discos de 36.4 GB. El bus del sistema permite una velocidad de transferencia de 9.6 GB/segundos., los adaptadores de E/S integrados y las 9 ranuras PCI proporcionan un sistema altamente escalable y bien balanceado.
Este servidor ofrece otras funciones avanzadas, además de ranuras PCI, discos, fuentes de poder y ventiladores de enfriamiento "Hot-Swap". Entre las funciones adicionales figuran la recuperación automática del sistema (ASR, por sus siglas en inglés) y el control remoto del sistema (RSC, por sus siglas en inglés). La función ASR se configura alrededor de componentes que fallan, permitiendo, de esta forma, la recuperación del servicio que es lo más importante. La función RSC facilita la administración remota y/o centralizada. Todas estas características contribuyen a proporcionar un nivel superior de disponibilidad para las aplicaciones de misión crítica.

6.1.2 Especificaciones técnicas del servidor SunFire V880

OPCIONES DE PROCESADOR

Procesador	UltraSPARC III
Arquitectura	Superescalar SPARC 9
Cache (incorporada al procesador)	64KB para datos y 32KB para instrucciones
Secundaria (L2)	Externo de 8MB
MEMORIA	Máxima de 32 GB en el sistema cuando se configura con 8 procesadores

INTERFACES ESTÁNDARES/INTEGRADAS

Red	Gigabit Ethernet y Ethernet 10/100 BaseT
E/S	Controladores de disco FC-AL

ALMACENAMIENTO MASIVO Y MEDIOS

Adaptadores del sistema	SCSI interno para soportar el DVD interno
Cinta interna	Unidad opcional DDS-3 de 12 GB o DDS-4 de 20 GB
Disco interno	Hasta 12 discos FC-AL de 36.4 GB a 10000 RPM de 3.5" x 1.0" y soporte para unidades de 72.8 GB

OPCIONES DE CONSOLA

Monitor	Monitor opcional de 17" hasta 24"
Tarjeta de video	Sun PGX64 8/24 bits
Teclado y mouse	Teclado y mouse USB

FUENTES DE PODER

2 requeridas, 3 para redundancia, cada una con su propio cable de poder

Entrada CA máxima	1500 W por fuentes de poder
Salida CD máxima	1100 W por fuente de poder

AMBIENTE OPERATIVO

CA	100-240 VAC, 47-63 Hz, 1.48 KVA
Temperatura operativo	5° a 35° C, humedad relativa de 20% a 80%, sin condensación.

DIMENSIONES Y PESO

Altura	71.4 cm con ruedas
Ancho	48 cm.
Profundidad	83.6 cm.
Peso	88.1 kg. mínimo, aproximado 130.9 kg. máximo, aproximado

SOFTWARE

Ambiente operativo	Solaris 8 r/01 o superior
Lenguajes	C, C++, FORTRAN, Java

6.1.3 Arreglo de discos

El arreglo StorEdge A5200 es un equipo que ofrece alta capacidad de almacenamiento, así como alta disponibilidad para empresas pequeñas y medianas. Este arreglo de discos ofrece un excelente ancho de banda en transferencia de datos ya que tanto los discos como la comunicación es a través de fibra óptica. Su diseño incorpora componentes redundantes, dos canales de fibra, conexión a red, conexiones hot-plug (característica en los dispositivos que permite extraerlos del arreglo sin necesidad de eliminar la corriente eléctrica en cada uno de ellos), gracias a estas características se puede decir que el arreglo ofrece fiabilidad y disponibilidad. Es un dispositivo ideal para soluciones de misión crítica.

6.1.3.1 Especificaciones técnicas del arreglo de discos

TARJETA DE COMUNICACIÓN CON EL SERVIDOR

Tarjeta Sbus	Single-width fibre channel Canal óptico de comunicación serial
Tarjeta aceleradora de comunicación	Memoria cache de escritura rápida con 32 Mb de capacidad

SOFTWARE SOPORTADO

VERITAS Volume Manager	Sistema operativo solaris 2.7, 2.8 y 2.9 Interfaz gráfica X/Motif Software para manejo virtual del arreglo Diferentes niveles de RAID, 0 (striping), 1(espejo), 1+0 (espejo con striping) y 5 (striping con paridad distribuida) Software para valanceo de carga y buscador de ruta alterna

CARACTERÍSTICAS LOS DISCOS

Físicas	3.5 pulgadas de ancho incluyendo conector
De comunicación	Canal de fibra de loop arbitrado a 100 MB/s.
Capacidad soportada	18.2 GB, 36.4 GB, a 10000 rpm

DIMENSION Y PESO

Altura	22.7 cm.
Ancho	49.5 cm.
Profundidad	62.5 cm.
Peso (con 22 discos)	54 Kg.

AMBIENTE OPERATIVO

AC	200-240 VAC, 47-63 Hz., 2X24 A
POTENCIA DE SALIDA	3900W
OPERACIÓN	5° a 35°

6.2 CARACTERISTÍCAS DEL SOFTWARE

6.2.1 Revisión general del sistema operativo Solaris

Solaris es un sistema operativo a nivel empresarial que abarca el Sun Operating System (SunOS, sistema operativo de SUN) de multiprocesos y usuarios múltiples. También es un sistema operativo de red que se ejecuta en sistemas de computadoras personales basadas en Intel, además de sistemas construidos alrededor de arquitectura de CPU de SPARC. Estos sistemas pueden tener hasta 128 procesadores operando al mismo tiempo en el servidor Sun Fire 15K. Por tanto, cuando los administradores hablan de SUN, es probable que se refieran a sistemas de cómputo basados en SPARC o el entorno operativo de Solaris.

Solaris es el sistema operativo tipo UNIX dominante en el mercado, hoy día. Los sistemas de SUN son el hardware de elección para aplicaciones de alta disponibilidad, con sistemas de bases de datos, servidores Web y tareas que requieren mucha capacidad de cómputo como modelado y simulaciones. Estos sistemas se utilizan ampliamente en organizaciones comerciales y de investigación y desarrollo. También se integran bien en redes heterogéneas integradas por sistemas Linux y Microsoft Windows, sobre todo como servidores de archivos.

Algunas de las principales diferencias entre UNIX y las plataformas tipo Microsoft pueden rastrease si volvemos a los días de los sistemas de multiusuario, multiproceso. Por ejemplo, los kernels de Solaris pueden rastrear su origen al sistema V y a las variantes de UNIX de la Berkeley Software Distribution (BSD, distribución de software de Berkeley), mientras que Windows NT está basado en el kernel VMS desarrollado originalmente por los sistemas VAX de alto desempeño. Hay muchas similitudes; sin embargo, en Windows 2000, Micrososft introduce un nuevo servicio llamado Active Directory (AD) que es una reminiscencia del Network Information Service (NIS/NIS+) jerárquico de SUN, que se utiliza para administrar usuarios, sistemas y datos de dominio en redes grandes.

Los beneficios del uso de Solaris sobre otros sistemas operativos por lo general se vuelven evidentes en multiprocesamiento simétrico (SMP), en un entorno de multiusuario, o en ambos. Aunque Microsoft Windows soporta varios CPU, Solaris soporta hasta 128 CPU´s operando al mismo tiempo con una escala casi lineal de desempeño. Algunos otros sistemas operativos parecen dedicar la mayor parte de la capacidad de procesamiento de un segundo, tercero o cuarto CPU a la planificación más que las operaciones. Además, Solaris es particularmente adecuado para soportar cientos de usuarios interactivos en un solo sistema; es decir, todos los usuarios pueden registrar su inicio de sesión utilizando un escritorio que se está ejecutando en un servidor central.

Aunque Microsoft Windows presenta estupendos productos que permite a los usuarios ejecutar un escritorio de manera remota, estos productos sólo suelen permitir que un solo usuario ejecute una sesión en cualquier momento. Hay pocas restricciones firmes colocadas en los sistemas Solaris en lo que se refiere al soporte de usuarios que han iniciado sesión de manera concurrente. Esta es una razón importante por lo que los sistemas Solaris son favorecidos en un nivel empresarial. Con frecuencia, escuchamos a administradores que dicen que Linux hace todo esto y más. Es cierto que Linux tiene soporte a SMP, y también es cierto que Linux es un sistema multiusuario. Sin embargo, tiene que considerar la inversión que hace una empresa en hardware y software para realmente comprender los principales beneficios de Solaris como plataforma.

Solaris está 100 por ciento soportado y administrado por Sun Microsystems; Linux está desarrollado por Linus Trovalds, y el soporte comercial lo brindan diversos distribuidores, incluyendo Red Hat (http://www.redhat.com) y SuSe (http://www.suse.com). Aunque pueden pagar cuotas por soporte a ambos, igual que puede pagarlos Sun, Red Hat y SuSe no son los propietarios del código fuente del sistema operativo al que dan soporte, mientras que Sun sí lo es. En este sentido, Solaris tiene más en común con Microsoft Windows; es una plataforma de propietario que está 100 por ciento soportada por la organización que la administra. Otra ventaja de Solaris ha sido el desarrollo del lenguaje de programación de Java, que ha crecido rápidamente hasta capturar casi el 10 por ciento del mercado mundial en ingeniería de software.

6.2.2 Instalación del sistema operativo

6.2.2.1 Open Boot Prom

Una de las principales diferencias de hardware entre sistemas SPARC que ejecutan Solaris y sistemas PC que ejecutan Linux o Microsoft Windows es que los sistemas SPARC tienen una interfaz llamado OpenBoot PROM, que puede utilizarse para modificar parámetros de firmware antes del inicio. Está basado en el lenguaje de programación Forth y puede utilizarse para ejecutar programas que realizan las siguientes funciones:

- Inicio del sistema
- Realización de diagnósticos sobre dispositivos de hardware.
- Pruebas de conectividad con los diferentes periféricos.

Cuando se ejecuta un comando en el Open Boot PROM, puede pasar varias opciones a cada comando para modificar su comportamiento. Por ejemplo el comando boot toma varias opciones diferentes, incluyendo el dispositivo que debe iniciarse. Para iniciar desde el dispositivo predeterminado de inicio (por lo general, el disco duro principal), se escribiría:

```
ok> boot
```

Sin embargo, también es posible iniciar utilizando el CD-ROM al usar el comando

```
ok> boot cdrom
```

6.2.2.2 Tareas previas a la instalación del sistema operativo

Antes de instalar el sistema, se necesitará la siguiente información:

- Nombre de host (por ejemplo, www) Es el nombre que desea dar a su host para identificarlo de manera única en la red de área local.
- Dirección IP
- Nombre del dominio (por ejemplo, unam.mx) El nombre de dominio es la organización a la que pertenece su host. Todos los hosts de internet deben pertenecer a un dominio.
- Servidor de nombres (DNS) El servidor DNS relaciona la dirección IP con el nombre del dominio y viceversa.
- Máscara de red (por ejemplo, 255.255.255.0) La máscara se utiliza para localizar hosts que forman parte de la misma subred en la red de área local.

6.2.2.3 Instalación del sistema operativo

Hay cuatro configuraciones que se han desarrollado para Solaris, y se muestra junto con su tamaño aproximado de instalación en la tabal 6-1

DISTRIBUCIÓN	TAMAÑO APROXIMADO
Entire Distribution Plus Original Equipment Manufacturer (OEM) Support	2.4 Gb
Entire Distribution Without OEM Support	2.3 Gb
Developer System Support	1.9 Gb
End User System Support	1.6 Gb

Tabla 6.1
Tipos de instalación de Solaris y su tamaño

Al iniciar la instalación pide se seleccione el lenguaje, para nuestro caso usaremos el ingles ISO8859-15 como se muestra en la figura 6.1.

```
                Select a Language

          0   English
          1   French
          2   German
          3   Italian
          4   Japanese
          5   Korean
          6   Simplified Chinese
          7   Spanish
          8   Swedish
          9   Traditional Chinese


             Select a Locale

       58  U.S.A. (en_US.ISO8859-15)

         ... < remaining lines removed > ...
```

Figura 6.1
Selección de lenguaje

Al terminar, se iniciará la fase de carga del ambiente gráfico figura 6.2.

```
Starting Window System
```

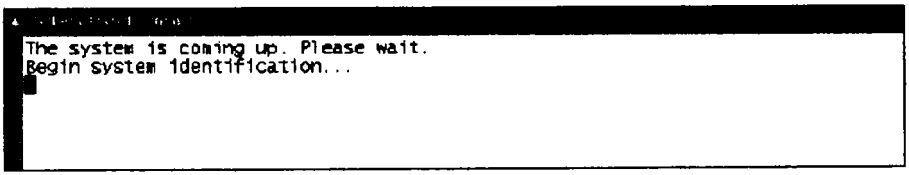


Figura 6.2
Inicio de ambiente gráfico

Iniciará la identificación del sistema, donde seleccionaremos los parámetros requeridos para nuestra aplicación figura 6.3 y 6.4.

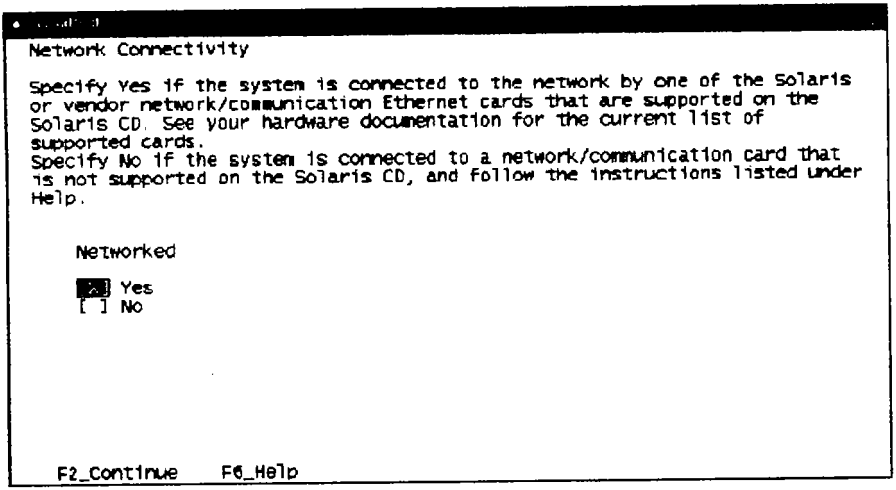


Figura 6.3
Selección para estar en red

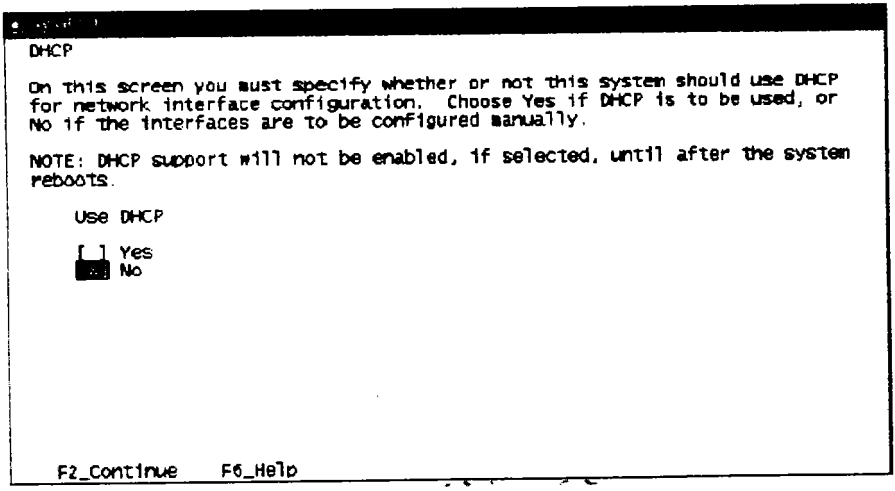


Figura 6.4
Selección de uso de DHCP

Nuestro sistema cuenta con dos tarjetas de red por lo que son reconocidas y seleccionaremos la hme0 como primaria como se muestra en la figura 6.5

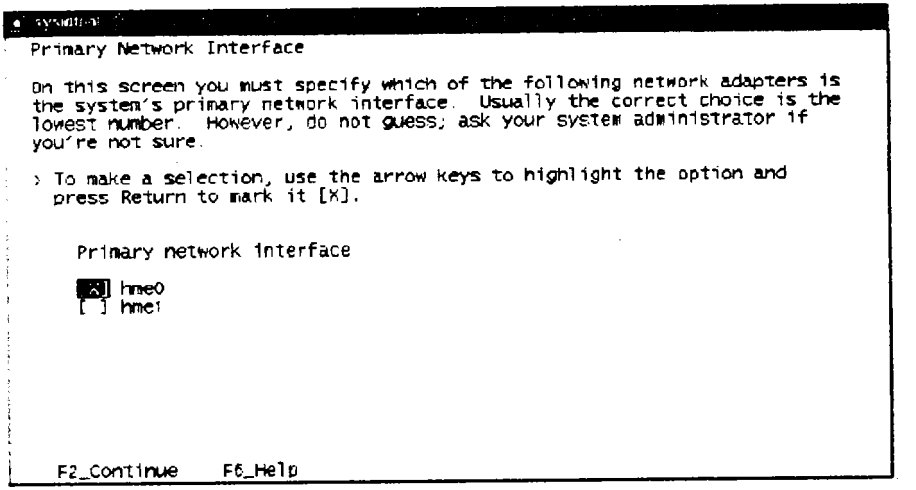


Figura 6.5
Selección de interfaz de red

Este nodo tendrá el nombre de sys41 como se indica en la figura 6.6

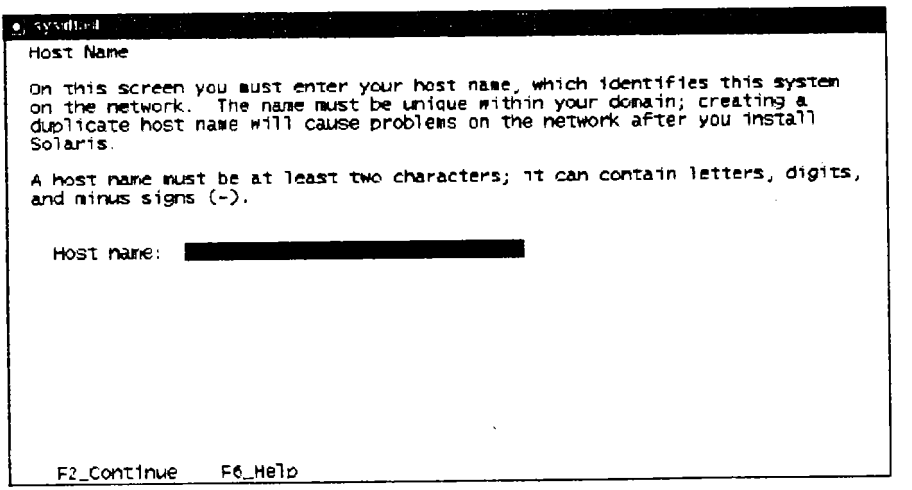


Figura 6.6
Nombre del nodo

Después asignaremos la dirección IP al nodo primario que será 192.168.30.41 como se muestra en la figura 6.7 y el la figura 6.8 se muestra la selección para ser o no parte de una subred, elegiremos que si.

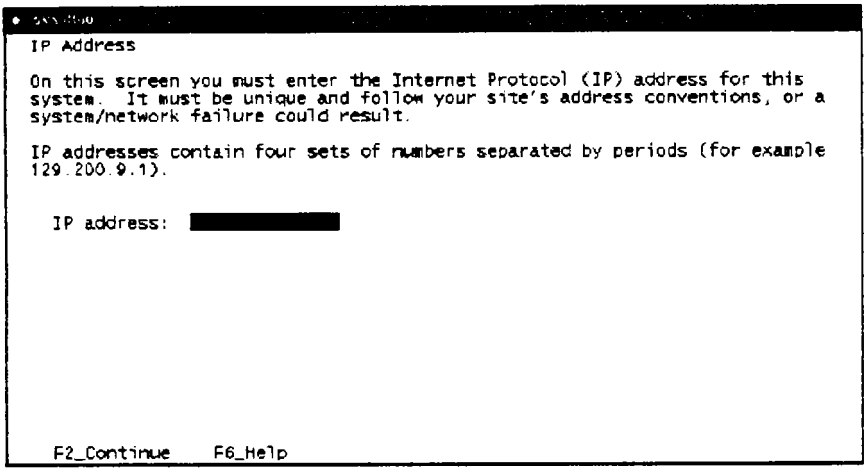


Figura 6.7
Asignación de dirección IP

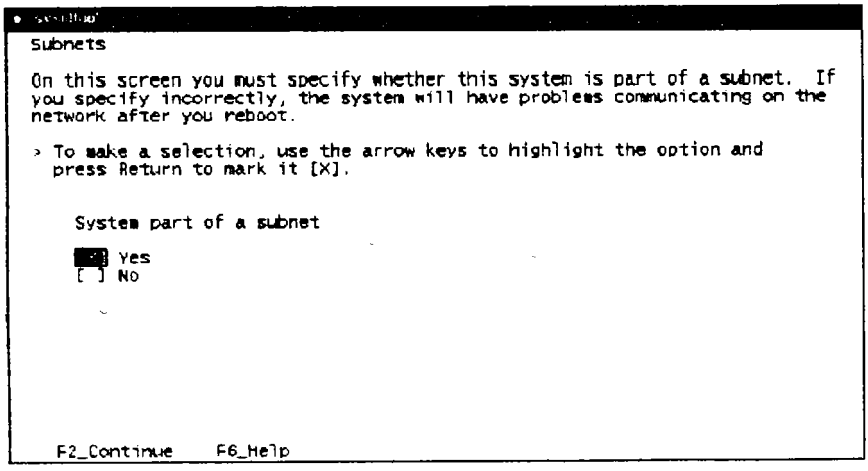


Figura 6.8
Selección para ser parte de una subred

Seleccionaremos en la siguiente ventana la mascara de red para nuestro caso corresponde 255.255.255.0 como lo indica la figura 6.9 y el la figura 6.10 se hace la selección de no para la habilitación de IPv6 pues usaremos IPv4.

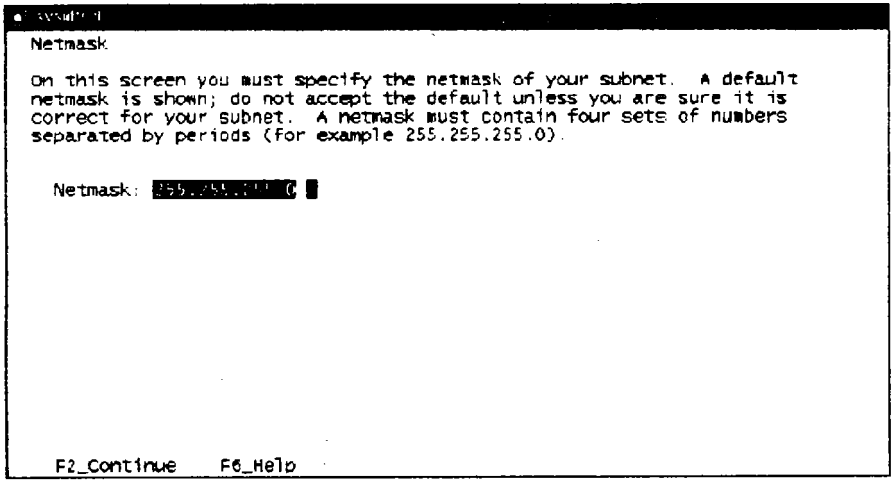


Figura 6.9
Selección de mascara de red

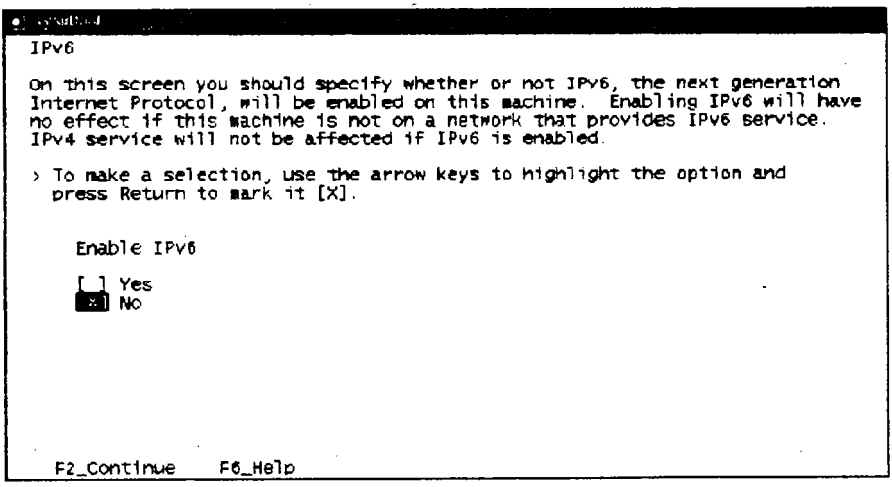


Figura 6.10
Selección de IPv6

A continuación se presenta la ventana de selección de ruteo por defecto se selecciona elegir una como se ilustra en la figura 6.11 y en la figura 6.12 se teclea la dirección de ruteo que será en nuestro caso la dirección 192.168.30.1.

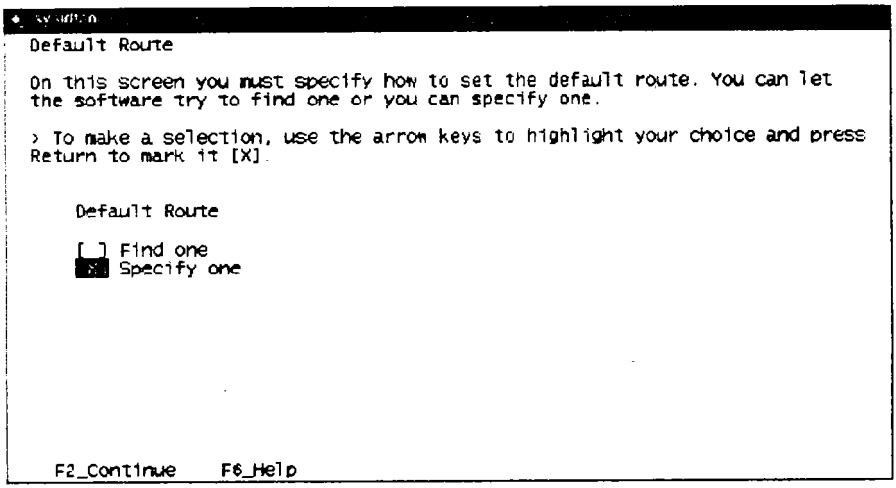


Figura 6.11
Selección de ruteo por defecto

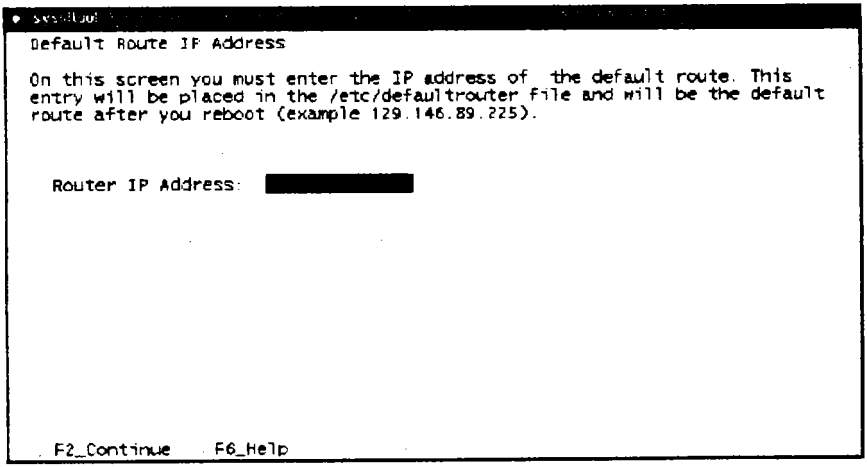


Figura 6.12
Dirección de ruteo por defecto

Se muestra a continuación una ventana que despliega la información seleccionada y capturada a fin de confirmar que los datos están correctos como lo ilustra la figura 6.13.

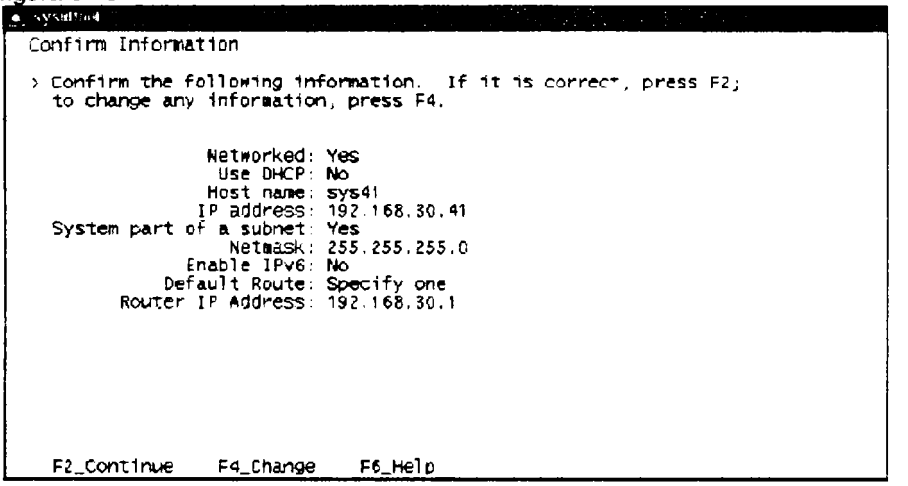


Figura 6.13
Confirmación de la información

Las políticas de seguridad no se configurarán ya que estaremos dentro de un Intranet con lo cual las normas de seguridad serán cubiertas con la predeterminada del sistema operativo como lo muestra la figura 6.14.

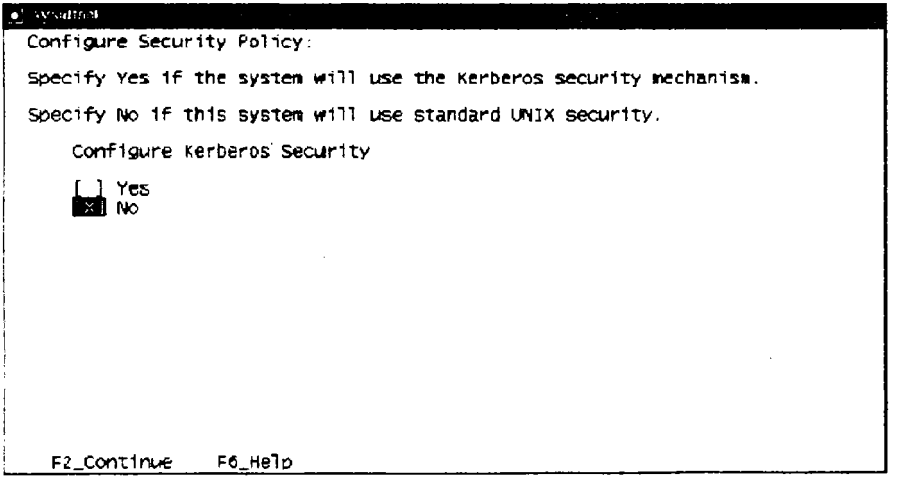


Figura 6.14
Selección para configuración de política de seguridad

A continuación se desplegará la ventana de selección de servicios de nombres, seleccionaremos ninguno como se muestra en la figura 6.15

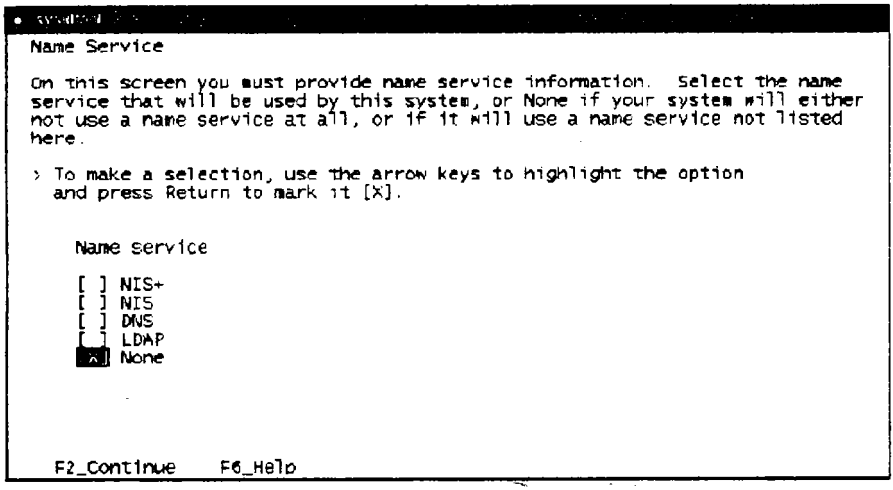


Figura 6.15
Selección de servicios de nombres

El siguiente paso consiste en seleccionar la zona horario para nuestro caso seleccionaremos Americas como lo muestra la figura 6.16

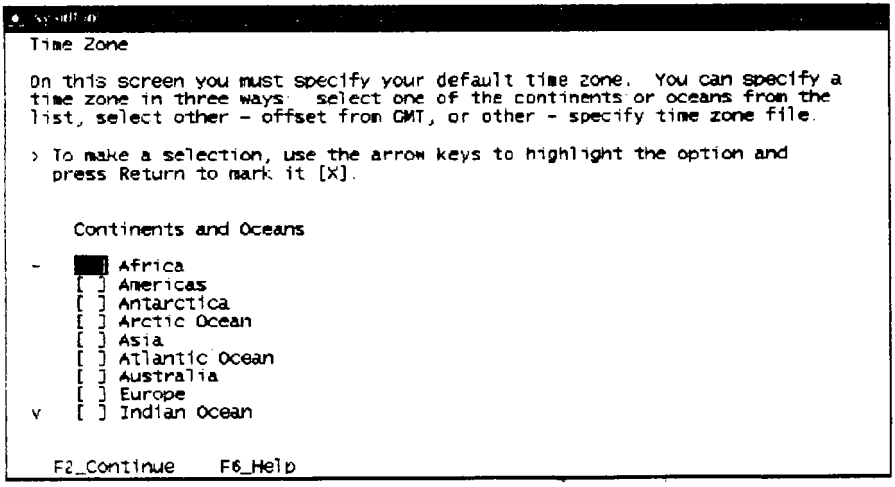


Figura 6.16
Selección de Zona horaria

A continuación se desplegará la selección de países, seleccionaremos México como se muestra en la figura 6.17.

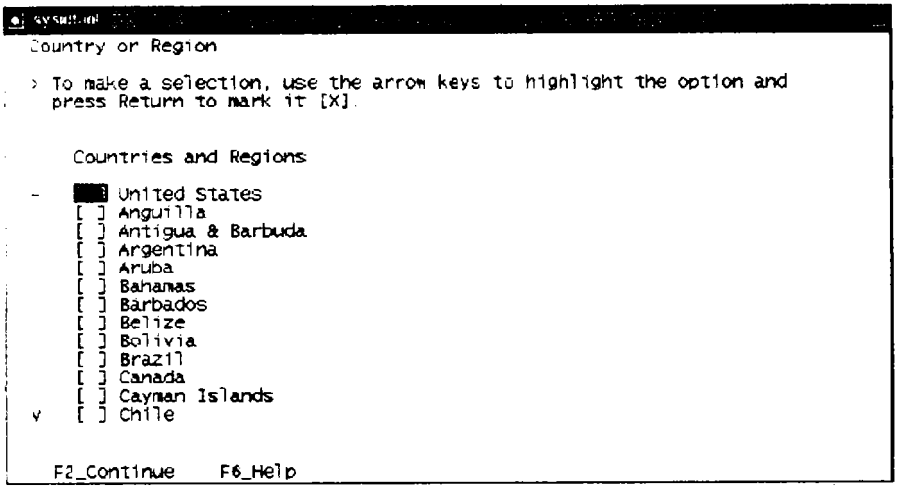


Figura 6.17
Selección de país.

El sistema procederá a identificar a los parámetros seleccionados como lo indica la figura 6.18

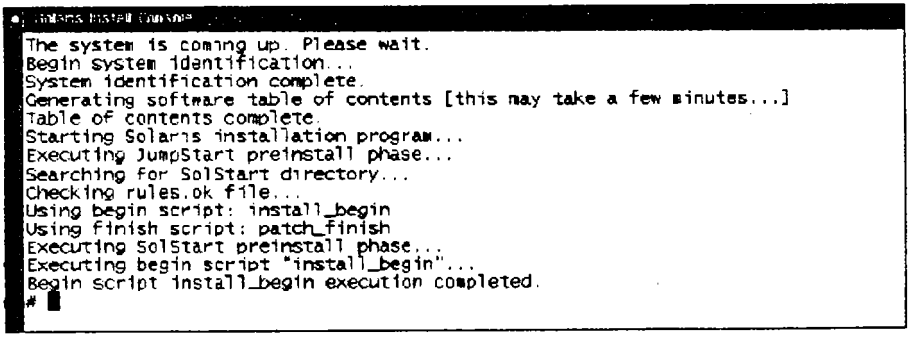


Figura 6.18
Comienzo de identificación del sistema

Se seleccionará a continuación la opción de instalación estándar como lo muestra la figura 6.19.

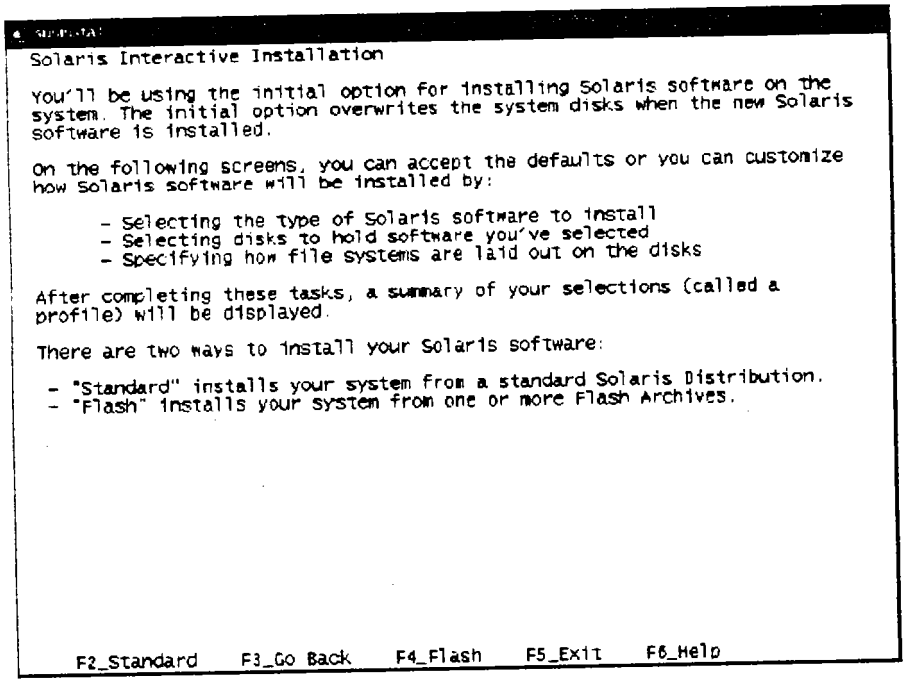


Figura 6.19
Selección de forma de instalación

En la siguiente ventana se muestra que el sistema operativo tiene soporte para el manejo de 64 bits en el hardware se ilustra en la figura 6.20

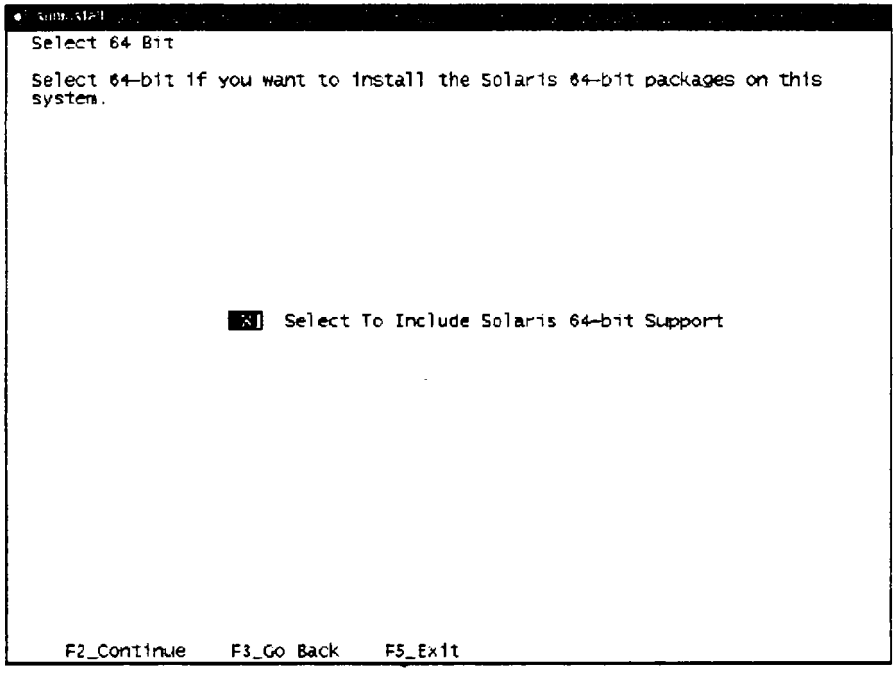


Figura 6.20
Selección de bits de operación

Después en la siguiente ventana se hace la selección de software dependiendo del tipo de instalación que se requiera nosotros selecionaremos la primera, es decir, Entire Distribution plus OEM support 64-bit como se muestra en la figura 6.21.

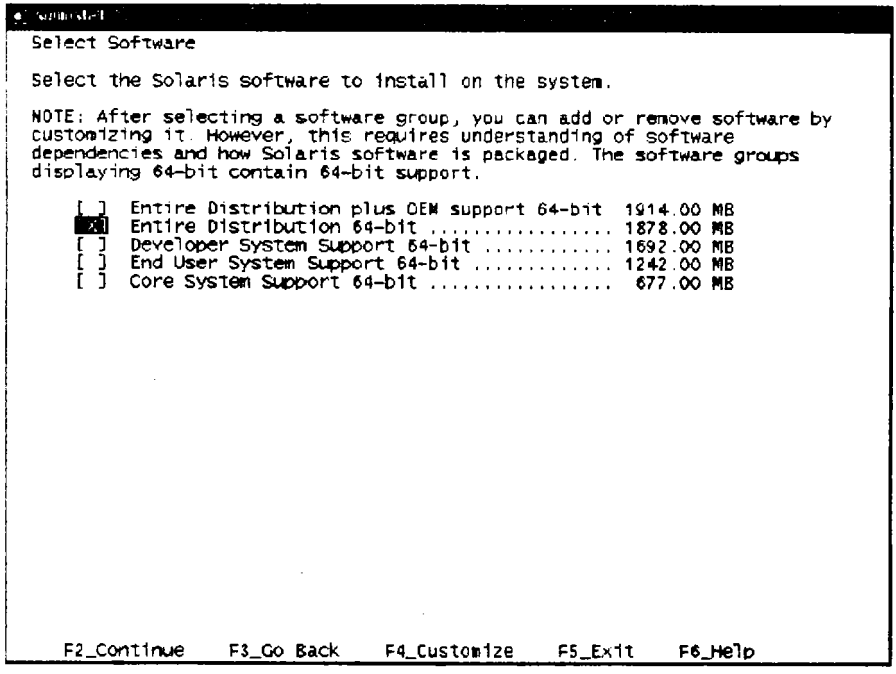


Figura 6.21
Selección de software a instalar

De fábrica los discos son creados con particiones, por lo que al ser detectados por el sistema operativo advierte que este tiene datos por lo que cuestiona si se quiere guardar esa información en la figura 6.22.

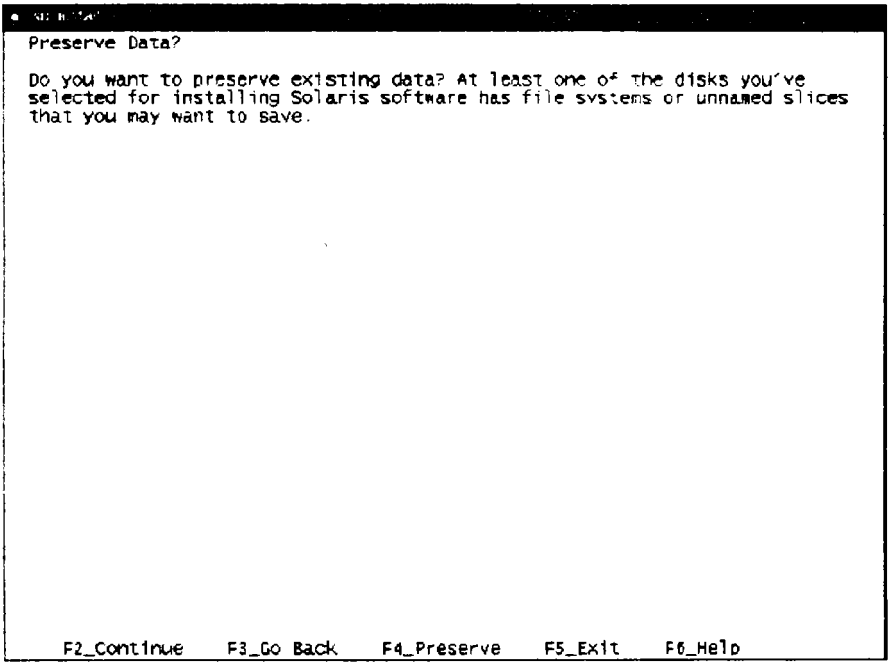


Figura 6.22
Preservación de datos

En la siguiente ventana se pregunta acerca de la partición del disco, hay dos opciones hacerlo automático o hacerlo manual se seleccionará este último (F4_Manual Layout) como puede verse en la figura 6.23.

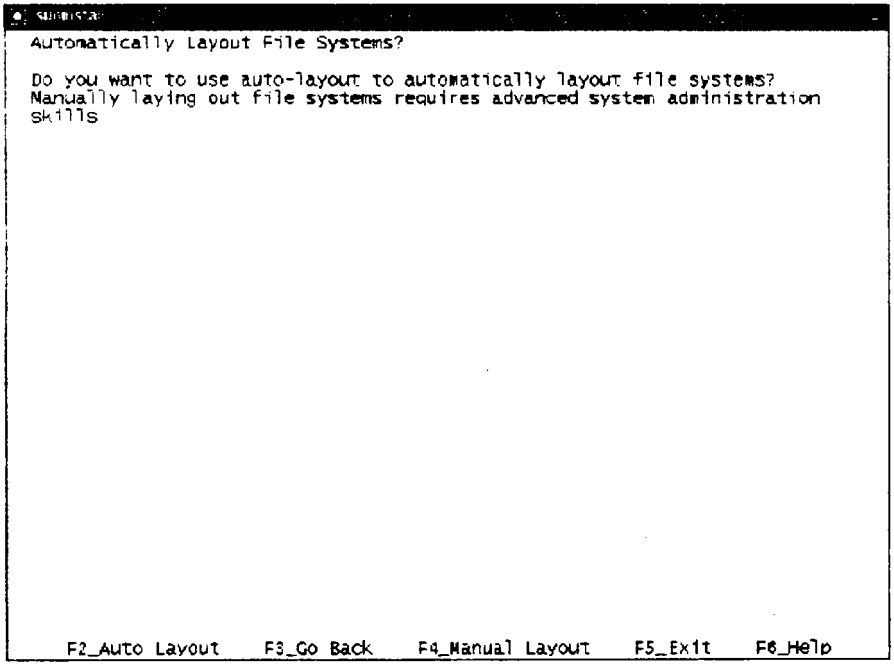


Figura 6.23
Selección para hacer partición automática o manual

En esta ventana se muestra el disco que fue seleccionado para instalar el sistema operativo en la partición número dos que es donde se encuentra toda la capacidad del disco, aquí seleccionamos la opción F4 como se indica en la figura 6.24

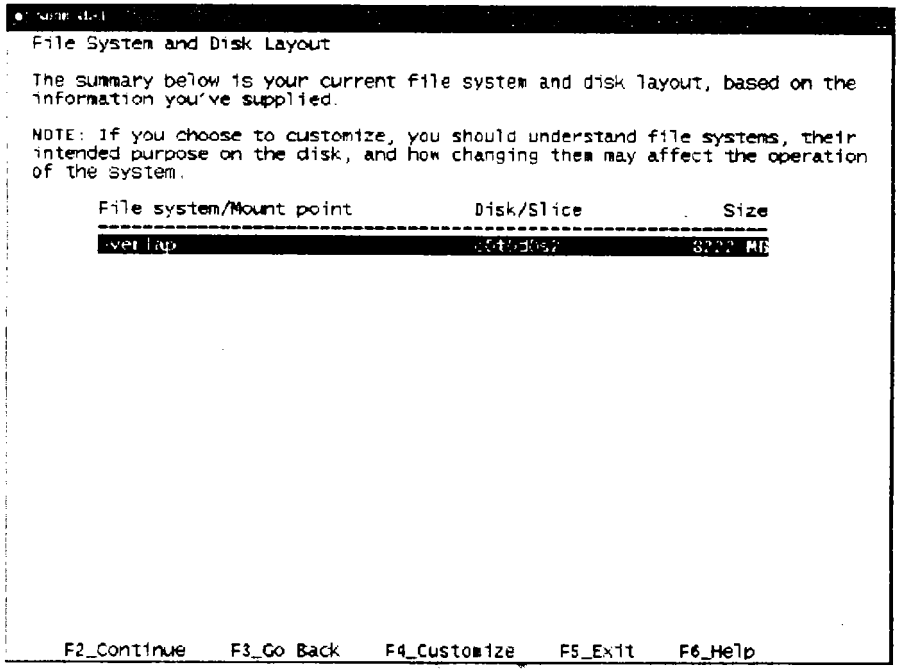


Figura 6.24
Disco seleccionado

El sistema operativo ofrece la opción de crear particiones recomendadas de forma automática, pero la descartaremos pues no aplica para nuestra situación como se ilustra en la figura 6.25. Al seleccionar la opción de partición manual nosotros creamos las particiones que requerimos así como el espacio necesario para cada una de ellas.

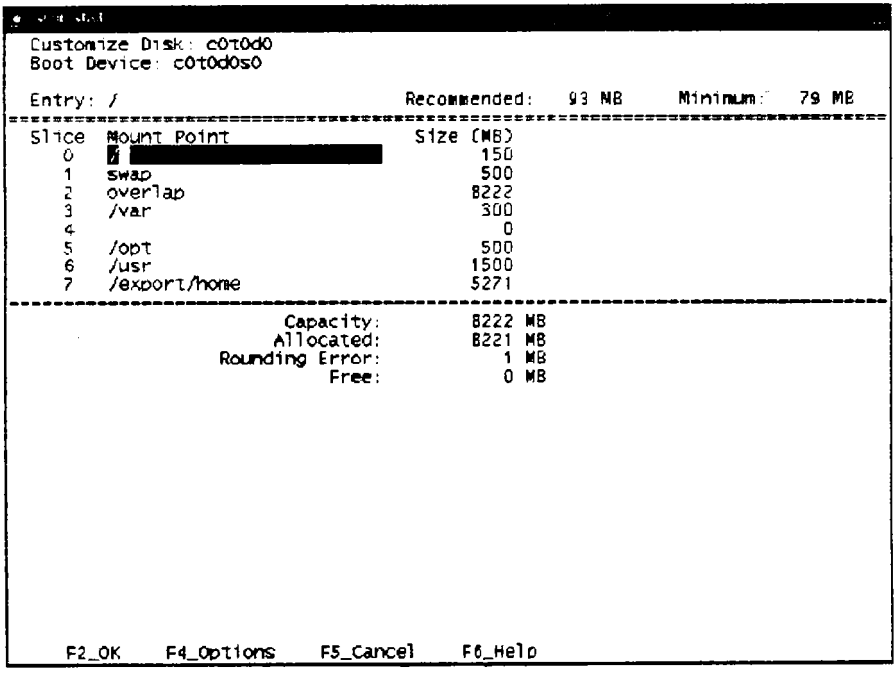


Figura 6.25
Particionamiento de disco

A continuación se presenta la confirmación de datos para el particionamiento como en la figura 6.26.

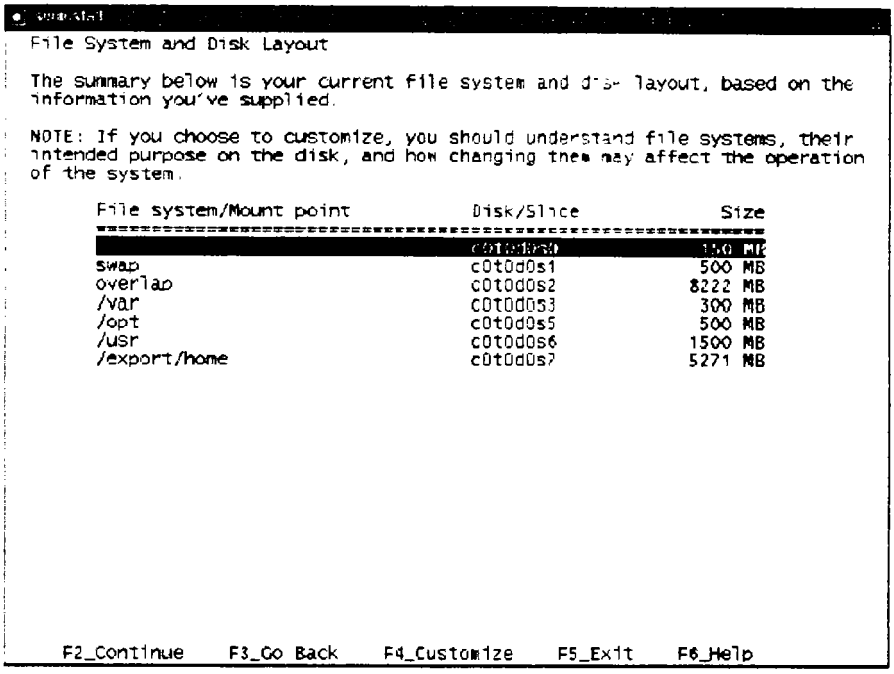


Figura 6.26
Disco particionado.

La siguiente ventana es para confirmar los parámetros de instalación como se muestra en la figura 6.27.

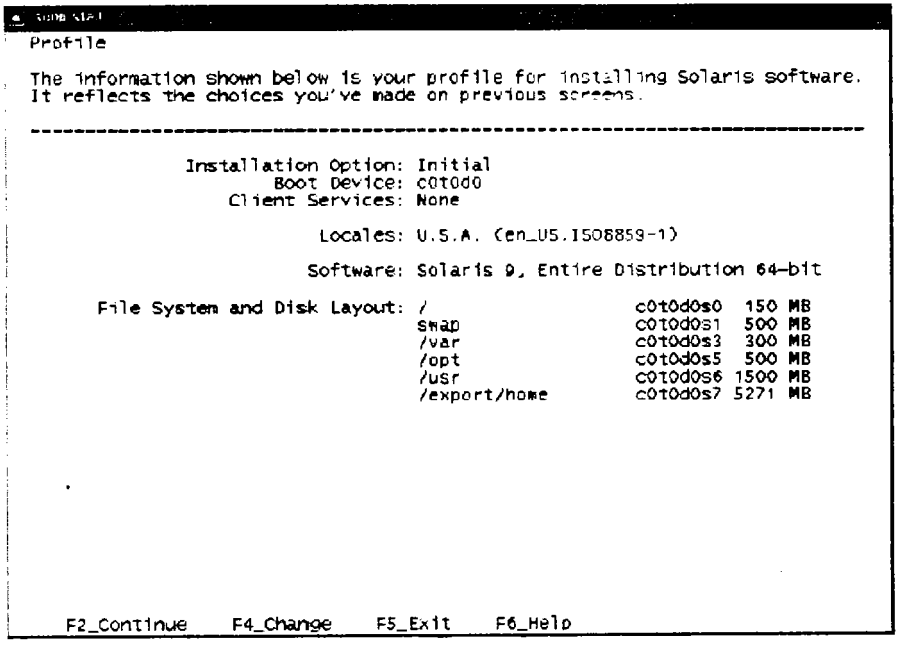


Figura 6.27
Confirmación de datos

En la figura 6.28 se ilustra la selección para elegir un autoreinicio o reinicio manual, elegiremos el manual.

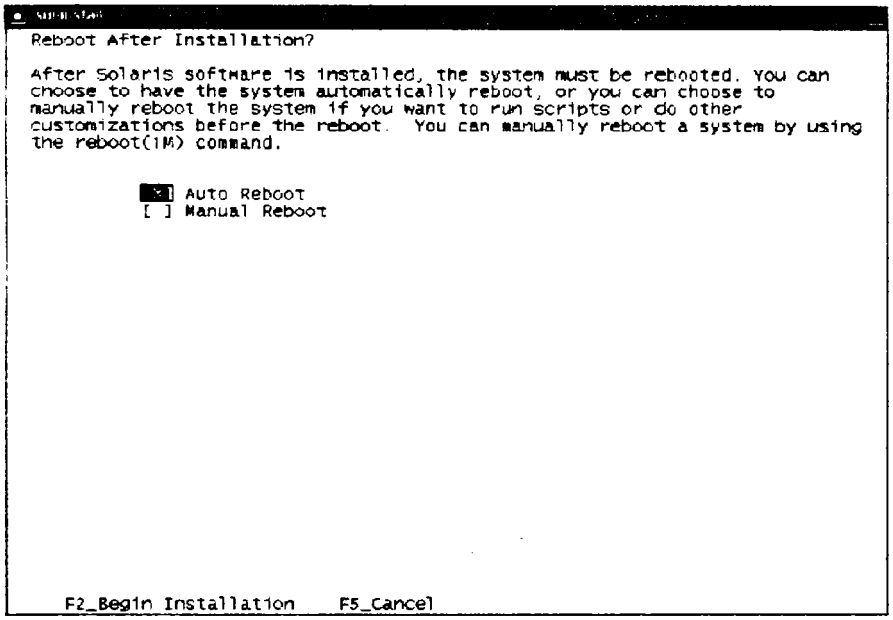


Figura 6.28
Selección de reinicio

Por último se despliega una ventana que mostrara el status del avance de la instalación como se ilustra en la figura 6.29.

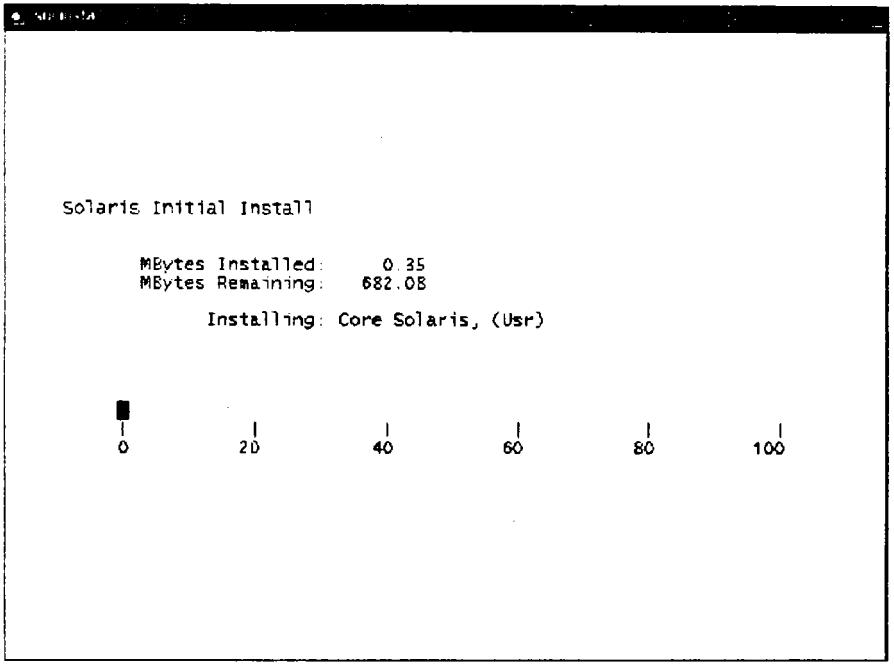


Figura 6.29
Progreso de la instalación

Es importante hacer notar que con los mismos pasos se procederá a instalar el nodo secundario y sólo habrá dos diferencias: el nombre del nodo sys42 y la dirección IP que será 192.168.30.42

6.2.3 Redundancia del disco de sistema operativo usando Solstice Disk Suite.

Solstice Disk Suite es un producto de software que está incluido en el sistema operativo, este software permite manejar un gran número de discos y realizar operaciones de RAID. Aunque ésta es una muy buena razón para usar Solstice Disk Suite es menester mencionar que incrementa la capacidad de datos almacenados y la disponibilidad de estos. Este software usa discos virtuales para poder manejar discos físicos y así asociar datos. En Disk Suite un disco virtual es llamando meta-dispositivo. Un meta-dispositivo funciona de forma idéntica que un disco físico tanto para la aplicación como para el sistema operativo, el usuario no se percatará de esta situación. Es de vital importancia mencionar que los meta-dispositivos son construidos de particiones. Las particiones pueden estar tanto en diferentes discos como en diferentes tarjetas controladoras sin general conflicto alguno para su administración.

6.2.3.1 Beneficios de usar Solstice Disk Suite

- Disponibilidad. Acceso a los datos tanto de lectura como escritura ya que permite construir espejos, RAID 5, y HOT SPARES, el último permite evacuar los datos de un disco a otro si alguno fallara.
- Confiabilidad. Los datos están protegidos contra corrupción.
- Desempeño. Mejora el rendimiento en la entrada/salida de datos ya que dependiendo del meta-dispositivo que se configure se permitirá un balanceo de carga.
- Capacidad. Los file systems pueden ser de mayor capacidad que los discos físicos.

6.2.3.2 Creación de un disco virtual de para el disco del sistema operativo usando RAID 1

Estando en la consola del equipo con los permisos de root se ejecutará la siguiente serie de comandos.

a) Crear las bases de datos así como las replicas

```
# metadb  -a -f  -c3 /dev/dsk/c0t0d0s7 /dev/dsk/c1t0d0s7
```

b) Crear el primer lado del espejo con la primera partición

```
#metainit -f d1 1 1 c0t0d0s0
```

c) Crear el primer lado del espejo con la segunda partición

```
#metainit -f d2 1 1 c0t0d0s1
```

d) Crear el primer lado del espejo con la tercera partición

```
# metainit -f d4 1 1 c0tt0d0s2
```

e) Crear el primer lado del espejo con la cuarta partición

```
# metainit -f d6 1 1 c0t0d0s3
```

f) Crear el primer lado del espejo con la quinta partición

```
# metainit -f d8 1 1 c0t0d0s4
```

g) Crear el primer lado del espejo con la sexta partición

```
# metainit -f d9 1 1 c0t0d0s5
```

h) Crear el segundo lado del espejo usando la primer partición del segundo disco

```
# metadb  -f d0 1 1 c1t0d0s0
```

i) Crear el segundo lado del espejo usando la segunda partición del segundo disco

```
#metainit -f d3 1 1 c1t0d0s1
```

j) Crear el segundo lado del espejo usando la tercer partición del segundo disco

```
# metainit -f d5 1 1 c1t0d0s2
```

k) Crear el segundo lado del espejo usando la cuarta partición del segundo disco

```
# metainit -f d7 1 1 c1t0d0s3
```

l) Crear el segundo lado del espejo usando la quinta partición del segundo disco

```
# metainit -f d9 1 1 c1t0d0s4
```

m) Crear el segundo lado del espejo usando la sexta partición del segundo disco

```
# metainit -f d21 1 1 c1t0d0s5
```

n) Usar el siguiente comando para que se modifiquen automáticamente los archivos /etc/vfstab y /etc/system, con la partición de donde se esta iniciando el sistema operativo

```
# metaroot d1
```

o) Ejecutar el comando lockfs

```
#  lockfs -fa
```

p) Crear los sub-espejos del meta-dispositivo con los siguientes comandos

```
#metainit d10 -m d1
#metainit d23 -m d2
#metainit d45 -m d4
#metainit d67 -m d6
#metainit d89 -m d8
#metainit d90 -m d20
```

q) Unir ambos lados de los espejos para crear el met-dispositivo

```
#metattach d10 d0
#metattach d23 d3
#metattach d45 d5
#metattach d67 d7
#metattach d89 d9
#metattach d90 d21
```

r) Editar el archivo /etc/vfstab
s) Dar un reinicio al equipo
t) Se aplicaran los mismos pasos para el segundo nodo

La siguiente figura muestra la forma en como se generará el disco virtual

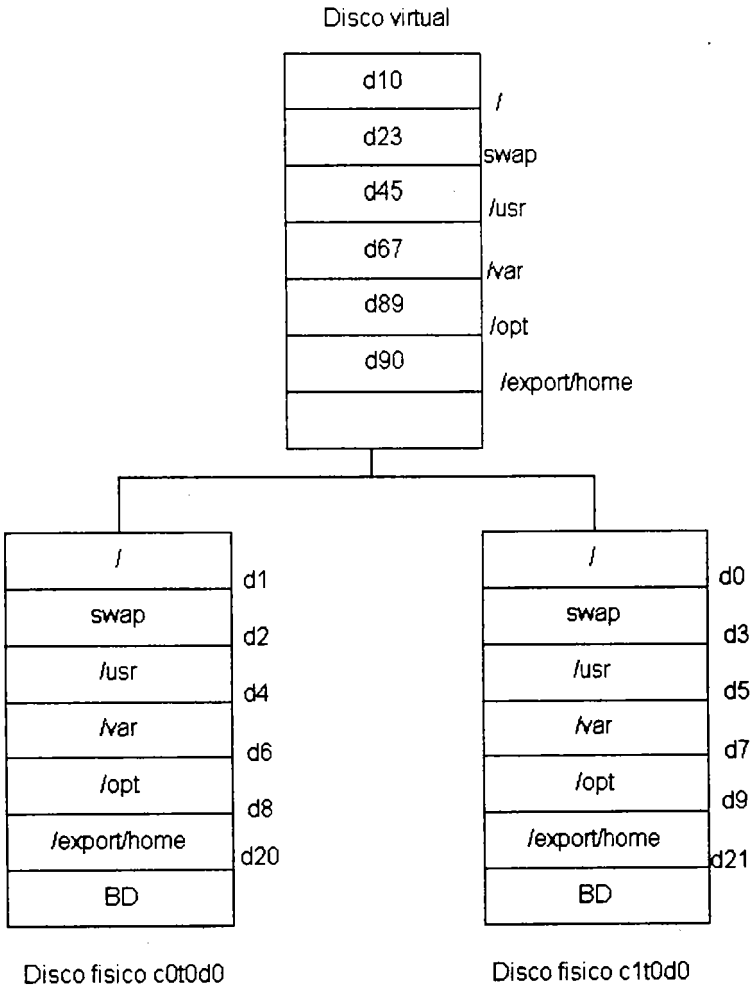


Figura 6.31
Composición del disco virtual.

6.2.4 Instalación de Veritas Volume Manager

Con todas las versiones de Veritas Volume Manager está incluida documentación donde se exponen diferentes escenarios para la instalación o actualización del sistema, ésta nos puede servir de guía para nuestras necesidades.
El CD-ROM de Veritas Volume Manager contiene los siguientes paquetes.

PAQUETE	DESCRIPCIÓN
VRTSvmdev	Encabezados y archivos de librerias

VRTSvmdoc	Documentación para el usuario
VRTSvmman	Paginas del Manual
VRTSvmsa	Administrador de arreglos (ambiente gráfico)
VRTSvxvm	Archivos binarios

Al insertar el CD-ROM dentro del equipo y ejecutar el comando pkgadd encontraremos lo siguiente:

```
#pkgadd -d pwd

The following packages are available:
     1 VRTSvmdev    VERITAS Volume Manager, Header and
                    Library Files
                    (sparc) 3.0.2,REV=08.30.2003.15.56
     2 VRTSvmdoc    VERITAS Volume Manager (user
                    documentation)
                    (sparc) 3.0.2,REV=08.26.2003.22.51
     3 VRTSvmman    VERITAS Volume Manager, Manual Pages
                    (sparc) 3.0.2,REV=08.30.2003.15.55
     4 VRTSvmsa     VERITAS Volume Manager Storage
                    Administrator(sparc)3.0.3,
                    REV=08.27.2003.13.55
     5 VRTSvxvm     VERITAS Volume Manager, Binaries
                    (sparc) 3.0.2,REV=08.30.2003.15.56

Select package(s) you wish to process (or 'all' to process
all packages). (default: all ) [?,??,q]: all
```

Nosotros responderemos "all" para que se instale todo el software.

6.2.4.2 Inicialización de Veritas Volume Manager

Es muy importante mencionar que al instalar el software no se ha inicializado, esto es, es necesario dar un reinicio al sistema y configurar el software de acuerdo a nuestras necesidades. Entenderemos como inicialización cuando el software maneje los discos o estén bajo el control de Volume Manager. Para inicializar los discos es necesario activar el programa vxinstall, lo cual se realizará en los siguientes párrafos.

6.2.4.3 Diferencias entre encapsulado e inicialización

Como ya se mencionó Volume Manager (VM) no inicializará al menos que un grupo de discos tenga el nombre de "rootdg", este grupo debe tener al menos un disco. Este es un requerimiento del software y para cumplir con este requerimiento Volume Manager ofrece dos opciones:

- Encapsular el disco de boot (el que tiene el sistema operativo)
- Inicializar un disco y agregarlo al grupo llamado "rootdg"

Proceso de encapsulado: Cuando un disco es encapsulado por VM, los file systems son preservados y una pequeña porción del disco es usado para poner un encabezado de VM. Si el disco es de boot, VM se percata de esta situación y toma acciones conducentes para preservar los datos.

Proceso de inicialización: Cuando un disco es inicializado por VM el disco es reparticionado con la configuración que maneja VM. VM genera únicamente dos particiones en el disco la tres y la cuatro. La primera es muy pequeña y es usada para guardar la configuración de VM y la segunda que es el resto del disco se usa para guardar los datos del usuario o aplicación. Al inicializar los discos toda la información que éste tenga se pierde.

6.2.4.4 Configuración de Veritas Volume Manager

Lo primero que hace el programa "vxinstall" es buscar todas las tarjetas controladoras que se encuentran en el sistema y al terminar de buscar las controladoras pregunta por el tipo de configuración, rápida o la que se hace de acuerdo a las características de cada sistema (custom). Es muy importante leer cuidadosamente las preguntas que el programa va haciendo pues con este tipo de configuración los datos que estén dentro de los discos duros se perderán. Este programa solo se debe ejecutar una sola ocasión pues existe la posibilidad de que se comporte de forma errónea en las ocasiones subsecuentes. A continuación se muestra la salida de este comando al ser ejecutada en el servidor

```
#vxinstall

Generating              list              of              attached
controllers.........................
— |σ—— |σ—— |σ——|σ——|σ——

Volume Manager Installation
Menu: Volume Manager/Install

The Volume Manager names disks on your system using the
controller and disk number of the disk, substituting them
into the following pattern:
          C<controller> t<target> d<disk>

If the Multipathing driver is installed on the system then
forthe disk devices with múltiple access paths, the
controller number represents a multipath pseudo controller
number. For example, if a disk has 2 paths from controllers
c0 and c1, then the Volume Manager
```

```
Display only one of them such as c0 to represent  both the
controllers.

Some examples would be:
c0t0d0 - first controller, first target, first disk
c1t0d0 - second controller, first target, first disk
c2t1d0 - second controller, second target, first disk

The Volume Manager has detected the following controllers on
your
System:
c0:
c1:
c2:
Hit RETURN to continue
```

Después de identificar todas las tarjetas controladoras de discos, vxinstall pregunta sobre el tipo de instalación.

```
Volume Mager Installation
Menu: Volume Manager/Install

You will now be asked if you wish to use Quick Installation
or Custom Installation.Custom
Installation allows you to select how the Volume Manager will
andel the installation
of each disk attached to your system.
Quick Installation examines each disk attached to your system
and attempts to create volumes to cover all disk partitions
that might be used for files systems or for others similares
purposes.
If you do not wish to use some disks with the Volume Manager,
or if you wish to reinitialize some disks, use the Custom
Installation option Otherwise, we suggest that you use the
Quick Installation option.

Hit RETURN to continue.

Volume Manager Installation Options
Menu: Volume Manager/Install

1       Quick Installation
2       Custom Installation

?       Display help about menu
??      Displayhelp about the menuing system
q       Exit from menus
```

```
Select an operation to perform: 2
```

Nosotros seleccionamos la opción 2 ya que la instalación rápida no es recomendada pues por defecto encapsula el disco de boot.
Durante el proceso de instalación VM detecta que el disco tiene datos y por lo tanto pregunta si queremos encapsularlo, a lo que responderemos que no.

```
Volume Manager Custon Installation
Menu: Volume Manager/Install/custom

The c0t0d0 disk is your Boot Disk. You can not add it as new
disk.
If you encapsultae it, you will make your root filesystem and
other system areas onnthe Boot Disk into volúmenes. This is
required if you wish to mirror your root filesystem or system
swap area.

Encapsulate Boot Disk [y,n,q,?] (default: n) n

Volume Manager Custon Installation
Menu: Volume Manager/Install/custom
Generating list of attached disk on c0........

<excluding root disk c0t0d0>
Hit RETURN to continue

The Volume Manager has detected the following disk on
controller c2
c2t0d0  c2t1d0 c2t5d0 c2t9d0 c2t10d0 c2t16d0 c2t19d0 c2t22d0
c2t26d0

The Volume Manager has detected the following disk on
controller c3
c3t0d0 c3t1d0 c3t5d0 c3t9d0 c3t10d0 c3t16d0 c3t19d0 c3t23d0
c3t26d0

Is this correct [y,n,q,?] (default : y) q

#
```

Al terminar de detectar las tarjetas controladoras procederemos a construir nuestros volúmenes virtuales de la siguiente forma:

```
#vxdiskadm
```

```
Volume Manager Support Operations
Menu: VolumeManager/Diks

   1     Add or initialize one or more disk
   2     Encapsulate one or more disks
   3     Remove a disk
   4     Remove a disk for replacement
   5     Replace a failed or removed disk
   6     Mirror volumes on a disk
   7     Move volumes from a disk
   8     Enable acces to (import) a disk group
   9     Remove access to (deport) a disk group
   10    Enable (online) a disk device
   11    Disable (offline) a disk device
   12    Mark a disk as a spare for a disk group
   13    Turn off the spare flag on a disk
   14    Unrelocate subdisk back to a disk
   15    Exclude a disk from hot-relocation use
   16    Make a disk available for hot-relocation use
   17    Prevent multipathing/Suppress device from VxVM's view
   18    Allow multipathing/Unsuppress devices from VxVM's
         view
   19    List currently suppressed/non-multipathed devices
   20    Change the disk naming scheme
   21    Get the newly connected/zone disks in VxVM view list
         disk information

   ?        Display help about menu
   ??       Display help about the menuing system
   q        Exit from menus

   Select an operation to perform: 1

   Select disk devices to add: [<pattern-list>,all,list,q,?]
   c2t0d0
   Which disk group [<group>,none,list,q,?] (defualt: rootdg)
   mercado
   There is no active disk group named mercado

   Create a nes group namaed mercado? [y,n,q,?] (default: y)
   y

   Use a default disk name for the disk? [y,n,q?] (default:
   y) n

   Add disk as a spare disk for mercado? [y,n,q,?] (default:
   n) n
```

```
Exclude disk from hot-relocation use? [y,n,q,?] (default:
n) y
A new disk group will be creayed named mercado and the
selected disks
will be added to the disk group with disk names that will
be specified interactively.
They will be marked as nohotuses and excluded from hot-
relocation use

    c2t0d0

     Continue with operation? [y,n,q,?] (default: y) y

    Initializing device c2t0d0

     Use a defualt private region length for this disk?
     [y,n,q,?] (default :y) y

  Enter disk name for c2t0d0 [<name>,q,?] (default:
mercado01) mercado01

    Creating a new disk group named mercado containing the
disk device
    c2t0d0 with the name mercado.

     Add or initialize other disks? [y,n,q,?] (default: n) n
```

Después de realizar esta operación en dos ocasiones procedemos a ejecutar el siguiente comando con la salida respectiva, donde se muestra la configuración de los dos grupos de discos

```
#vxdisk list
DEVICE       TYPE      DISK         GROUP        STATUS
c2t0d0s2     sliced    merc01       mercado      online
c2t1d0s2     sliced    merc13       mercado      online
c2t5d0s2     sliced    merc02       mercado      online
c2t9d0s2     sliced    merc14       mercado      online
c2t10d0s2    sliced    merc03       mercado      online
c2t16d0s2    sliced    merc04       mercado      online
c2t19d0s2    sliced    merc05       mercado      online
c2t22d0s2    sliced    merc06       mercado      online
c2t26d0s2    sliced    apl01        aplmercado   online
c3t0d0s2     sliced    merc07       mercado      online
c3t1d0s2     sliced    merc15       mercado      online
c3t5d0s2     sliced    merc08       mercado      online
c3t9d0s2     sliced    merc16       mercado      online
c3t10d0s2    sliced    merc09       mercado      onlin
```

```
c3t16d0s2    sliced    merc10       mercado      online
c3t19d0s2    sliced    merc11       mercado      online
c3t23d0s2    sliced    merc12       mercado      online
c3t26d0s2    sliced    apl02        aplmercado   online
```

Ahora se procede a crear el volumen virtual con el grupo de discos mercado

```
#vxassist -g mercado make datos1_ms 683Mb layout=mirror
c2t0d0 c3t0d0 c3t1d0 c3t23d0 c2t22d0 c3t19d0 c2t19d0 c3t16d0
c2t16d0 c3t10d0 c2t10d0 c3t5d0 c3t1d0
```

6.2.5 Características de Veritas Cluster Server

Cluster Server de VERITAS; es la mejor solución para industrias que cuentan con sistemas abiertos y requieren proteger sus servicios críticos de tiempos fuera ya sea que estén planeados o se presenten de imprevisto. Los servidores en cluster puede aumentar el funcionamiento del negocio ya que proveen mínimos tiempos de recuperación en caídas intempestivas

Es Altamente escalable pues soporta hasta 32 servidores en una SAN y ambientes cliente/servidor. VERITAS Cluster proporciona la flexibilidad para agregar o quitar los servidores en el Cluster según las necesidades, sin poner el servicio fuera de línea.

6.2.5.1 Instalación y configuración de Veritas Cluster Server

Desde el cd-rom nos cambiamos al directorio clustr_server

```
# cd cluster_server
```

Ya en el directorio se ejecuta el archivo installvcs.

```
# ./installvcs
```

3. Al ejecutar este archivo el software enviará lo siguiente:

```
VERITAS CLUSTER SERVER 4.0 INSTALLATION PROGRAM
      Copyright (c) 2003 VERITAS Software Corporation.
      All rights reserved.

      VERITAS, the VERITAS Logo and all other VERITAS
      product names and slogans are trademarks or
      registered trademarks of VERITAS Software
      Corporation. VERITAS and the VERITAS Logo Reg. U.S.
      Pat. & Tm. Off. Other product names and/or slogans
```

```
    mentioned herein may be trademarks or registered
    trademarks of their respective companies.

Enter the system names separated by spaces on which to
install VCS:
```

Después de los dos puntos pondremos el nombre de los nodos que configuraremos, para nuestro caso se llamarán sys41 y sys42.

Al terminar de dar los nombres y dar la tecla intro, el software se encargará de identificar las características de cada uno de los nodos y comenzará la instalación del software en el nodo.

```
Checking OS version on sys42 .....................SunOS  5.9
Checking VRTSvcs package .....................not installed
Verifying communication with sys41 ..........ping successful
Attempting rsh with sys41 ....................rsh successful
Attempting rcp with sys41 ....................rcp successful
Checking OS version on sys41 .....................SunOS 5.9
Checking VRTSvcs package .....................not installed
Creating log directory on sys41 ......................Done

Logs for installvcs are being created in
/var/tmp/installvcsdate_time.

Using /usr/bin/rsh and /usr/bin/rcp to communicate with
remote systems.

Initial system check completed successfully.


Installing VERITAS Infrastructure packages on sys42:

Checking VRTSvlic package ..................... not installed
Checking VRTScpi package ...................... not installed
Checking file system space ........ required space available
Installing VRTScpi 4.0.4 on sys42 ..................... Done
Installing VRTSvlic 3.02.005d on sys42 ................ Done

Installing VERITAS Infrastructure packages on sys41:

Checking VRTSvlic package .....................not installed
Checking VRTScpi package ...................... not installed
Checking file system space ........ required space available
Copying VRTScpi package to sys41..................... Done
Installing VRTScpi 4.0.4 on sys41..................... Done
Copying VRTSvlic.tar.gz to sys41 ..................... Done
Installing VRTSvlic 3.02.005d on sys41 ................ Done

VERITAS Infrastructure packages installed successfully.
```

A estos paquetes se les conoce como infraestructura ya que son los encargados de monitorear y realizar el cambio de servicios en los nodos

Inmediatamente al terminar de instalar la infraestructura se solicitarán las licencias

```
Each system requires a VCS product license before
installation. License keys for additional product features
should also be added at this time.

Some license keys are node locked and are unique per system.
Other license keys, such as demo keys and site license keys,
are registered on all systems and must be entered on the
first system.

VCS Licensing Verification:

Checking VCS license key on sys42 ..............not licensed
Enter a VCS license key for sys42: [?] XXXX-XXXX-XXXX-XXXX-
XXX

Registering XXXX-XXXX-XXXX-XXXX-XXX on sys42 ........ Done

Do you want to enter another license key for sys42? [y,n,q,?]
(n)

Registering XXXX-XXXX-XXXX-XXXX-XXX on sys41

Checking VCS license key on sys41 ..................Cluster
Server

Do you want to enter another license key for sys41? [y,n,q,?]
(n)

VCS licensing completed successfully.

Press [Return] to continue:
```

Al dar enter se iniciará la instalación de todos los paquetes que faltan para dar funcionalidad al sistema.

```
installvcs can install the following optional VCS packages:

VRTSobgui              VERITAS Enterprise Administrator
VRTSvxfen              VERITAS I/O Fencing
VRTSvcsmn              VERITAS Cluster Server Man Pages
VRTSvcsdc              VERITAS Cluster Server Documentation
VRTScssim              VERITAS Cluster Server Simulator
```

```
VRTScscm                    VERITAS Cluster Server Cluster
                            Manager

1) Install all of the optional packages

2) Install none of the optional packages

3) View package description and select optional packages

Select the optional packages to be installed on all
systems? [1-3,q,?] (1)
```

Al preguntar por los paquetes a instalar, responderemos con el número uno que corresponde a la instalación de todos los paquetes faltantes, e iniciará la instalación

```
installvcs will install the following VCS packages:
VRTSperl        VERITAS Perl 5.8.0 Redistribution
VRTSob          VERITAS Enterprise Administrator Service
VRTSmuob        VERITAS Enterprise Administrator Service
                Localized Package
VRTSobgui       VERITAS Enterprise Administrator
VRTSllt         VERITAS Low Latency Transport
VRTSgab         VERITAS Group Membership and Atomic Broadcast
VRTSvxfen       VERITAS I/O Fencing
VRTSvcs         VERITAS Cluster Server
VRTSvcsmg       VERITAS Cluster Server Message Catalogs
VRTSvcsag       VERITAS Cluster Server Bundled Agents
VRTSvcsmn       VERITAS Cluster Server Man Pages
VRTScspro       VERITAS Cluster Server VEA Provider
VRTSvcsdc       VERITAS Cluster Server Documentation
VRTSjre         VERITAS Java Runtime Environment
                Redistribution
VRTScutil       VERITAS Cluster Utilities
VRTScssim       VERITAS Cluster Server Simulator
VRTScscw        VERITAS Cluster Server Configuration Wizards
VRTSweb         VERITAS Java Web Server
VRTSvcsw        VERITAS Cluster Manager (Web Console)
VRTScscm        VERITAS Cluster Server Cluster Manager

            Press [Return] to continue:

Checking VCS installation requirements on sys42:

Checking VRTSperl package ..... not installed
Checking VRTSob package ..... not installed
Checking VRTSmuob package ..... not installed
Checking VRTSobgui package ..... not installed
Checking VRTSllt package ..... not installed
Checking VRTSgab package ..... not installed
Checking VRTSvcs package ..... not installed
```

```
Checking VRTSvcsmg package ..................... not installed
Checking VRTSvcsag package ..................... not installed
Checking VRTSvcsmn package ..................... not installed
Checking VRTScspro package ..................... not installed
Checking VRTSvcsdc package ..................... not installed
Checking VRTSjre package ....................... not installed
Checking VRTScutil package ..................... not installed
Checking VRTScssim package ..................... not installed
Checking VRTScscw package ...................... not installed
Checking VRTSweb package ....................... not installed
Checking VRTSvcsw package ...................... not installed
Checking VRTScscm package ...................... not installed
Checking VERITAS patch 115209 .................. not installed
Checking VERITAS patch 115212 .................. not installed
Checking VERITAS patch 115210 .................. not installed
Checking file system space ..... required space is available
Checking had process ........................... not running
Checking hashadow process ...................... not running
Checking CmdServer process ..................... not running
Checking notifier process ...................... not running
Checking vxsvc process ......................... not running
Checking vxfen driver .......................... not running
Checking gab driver ............................ not running
Checking llt driver ............................ not running

Installation requirement checks completed successfully.
```

Al terminar de instalar el software inmediatamente y de forma automática se lanza la configuración del cluster y pregunta si se quiere realizar de manera manual o semi-automatica.

```
It is optional to configure VCS now. If you choose to
configure VCS later, you can either do so manually or run
the installvcs -configure command. Are you ready to
configure VCS? [y,n,q] (y) y
```

Aceptamos realizarlo en este momento y continuará con una serie de preguntas para realizar la configuración

```
To configure VCS the following is required:

A unique Cluster name
A unique Cluster ID number between 0-255
Two or more NIC cards per system used for heartbeat links

One or more heartbeat links are configured as private
links
```

```
One heartbeat link may be configured as a low priority
link
All systems are being configured to create one cluster

Enter the unique cluster name: [?] cluster-unam

Enter the unique Cluster ID number between 0-255: [b,?] 7
```

El instalador iniciará la detección de las tarjetas de red del servidor y seleccionaremos la que se configurará como hearbeat

```
Discovering NICs on sys42 ...discovered hme0 qfe0 qfe1 qfe2
qfe3

Enter the NIC for the first private heartbeat NIC on sys42:
[b,?] qfe0

Would you like to configure a second private heartbeat link?
[y,n,q,b,?] (y) Enter the NIC for the second private
heartbeat NIC on sys42: [b,?] qfe1

Would you like to configure a third private heartbeat link?
[y,n,q,b,?](n)

Do you want to configure an additional low priority heartbeat
link? [y,n,q,b,?] (n)

Are you using the same NICs for private heartbeat links on
all systems? [y,n,q,b,?] (y)
```

Al terminar de configurar el instalador permitirá verificar la información que se ingresó

```
Cluster information verification:
     Cluster Name: cluster-unam
     Cluster ID Number: 7
     Private Heartbeat NICs for sys42: link1=qfe0 link2=qfe1
     Private Heartbeat NICs for sys41: link1=qfe0 link2=qfe1
Is this information correct? [y,n,q] (y)
```

Veritas Cluster permite que el software sea manejado no sólo por el administrador del servidor, si no por un usuario que se puede encargar solo de administrar el sistema de alta disponibilidad, y es en esta parte donde el software de instalación cuestiona sobre esta situación

```
The following information is required to add VCS users:
A user name
A password for the user
User privileges (Administrator, Operator, or Guest)

Do you want to set the password for the Admin user
(default password='password')? [y,n,q] (n) y Enter
New Password:******

Enter Again:******

Do you want to add another user to the cluster? [y,n,q] (y)

Enter the user name: [?] aragon
Enter New Password:*******

Enter Again:******* Enter the privilege for user aragon
(A=Administrator, O=Operator, G=Guest): [?] a

Would you like to add another user? [y,n,q] (n)

     User: admin          Privilege: Administrators

     User: aragon         Privilege: Administrators

     Passwords are not displayed

Is this information correct? [y,n,q] (y)
```

Cuando se termina de dar de alta los usuarios se iniciará la configuración sobre los servicios de red que el sistema de alta disponibilidad proveerá

```
The following information is required to configure
Cluster Manager:

A public NIC used by each system in the cluster

A Virtual IP address and netmask for Cluster Manager

Do you want to configure Cluster Manager (Web
Console) [y,n,q](Y)


Active NIC devices discovered on sys42: hme0

Enter the NIC for Cluster Manager (Web Console) to use on
sys42: [b,?](hme0)


Is hme0 to be the public NIC used by all systems [y,n,q,b,?]
(y)
```

Configuraremos la dirección IP virtual que existirá para todo el sistema, es decir la que todos los clientes conocerán, por lo tanto se realizará en los siguientes pasos.

```
Enter the Virtual IP address for Cluster Manager: [b,?]
      192.168.30.50
Enter the netmask for IP 192.168.30.50: [b,?] (255.255.255.0)

Cluster Manager (Web Console) verification:
            NIC: hme0
            IP: 192.168.30.50
            Netmask: 255.255.255.0
      Is this information correct? [y,n,q] (y)
```

A partir de este momento se configurarán los servicios que se conocen como globales, es decir, aquellos que residen en ambos servidores pero que solamente el servicio es dado por uno de ellos, el otro se activa cuando un altercado ocurre en el servidor que provee el servicio.

```
The following is required to configure the Global Cluster
Option:
A public NIC used by each system in the cluster
A Virtual IP address and netmask

The Virtual IP address and NIC may be the same as
those configured for Cluster Manager (Web Console)

Do you want to configure the Global Cluster Option?
[y,n,q] (y)
```

Enseguida se inicia la configuración de los parámetros para generar los identificadores globales del sistema.

Enter the Virtual IP address for Cluster Manager: [b,?] (192.168.30.50)

Al presionar la tecla de enter se procede a verificar dichos parámetros como se muestra a continuación.

```
Global Cluster Option configuration verification:
                  NIC: hme0
                  IP: 192.168.30.50
```

```
                Netmask: 255.255.255.0

Matching Cluster Manager (Web Console) Virtual IP
configuration

Is this information correct? [y,n,q] (y)
```

Una vez que se valida la configuración se procede a instalar el software correspondiente

```
VCS packages can be installed on systems consecutively or
simultaneously.  Installing packages on systems
consecutively takes more time but allows for better error
handling.


Would you like to install Cluster Server packages on
all systems simultaneously? [y,n,q,?] (y) n

Installing Cluster Server 4.0 on sys42:


Installing VRTSperl 4.0 on sys42 .......Done 1 of 70 steps
Installing VRTSob 3.2.503.0 on sys42....Done 2 of 70 steps
Installing VRTSmuob 3.2.514.0 on sys42..Done 3 of 70 steps
Installing VRTSobgui 3.2.514.0 on sys42.Done 4 of 70 steps
Installing VRTSllt 4.0 on sys42.........Done 5 of 70 steps
Installing VRTSgab 4.0 on sys42.........Done 6 of 70 steps
Installing VRTSvxfen 4.0 on sys42.......Done 7 of 70 steps
Installing VRTSvcs 4.0 on sys42.........Done 8 of 70 steps
Installing VRTSvcsmg 4.0 on sys42.......Done 9 of 70 steps
Installing VRTSvcsag 4.0 on sys42......Done 10 of 70 steps
Installing VRTSvcsmn 4.0 on sys42......Done 11 of 70 steps
Installing VRTScspro 4.0 on sys42......Done 12 of 70 steps
Installing VRTSvcsdc 4.0 on sys42......Done 13 of 70 steps
Installing VRTSjre 1.4 on sys42........Done 14 of 70 steps
Installing VRTScutil 4.0 on sys42......Done 15 of 70 steps
Installing VRTScssim 4.0 on sys42......Done 16 of 70 steps
Installing VRTScscw 4.0 on sys42.......Done 17 of 70 steps
Installing VRTSweb 4.0 on nort.........Done 18 of 70 steps
Installing VRTSvcsw 4.0 on sys42.......Done 19 of 70 steps
Installing VRTScscm 4.0 on sys42......Done 20 of 70 steps
Adding patch 115209-05 on sys42........Done 21 of 70 steps
Adding patch 115212-05 on sys42........Done 22 of 70 steps
Adding patch 115210-05 on sys42........Done 23 of 70 steps


Installing Cluster Server 4.0 on sys41:
```

```
Copying VRTSperl.tar.gz to sys41.......Done 24 of 70 steps
Installing VRTSperl 4.0.2 on sys41.....Done 25 of 70 steps
Copying VRTSob.tar.gz to sys41........ Done 26 of 70 steps
Installing VRTSob 3.2.514.0 on sys41...Done 27 of 70 steps
Copying VRTSvcsw.tar.gz to sys41........ Done 61 of 70 steps
Installing VRTSvcsw 4.0 on sys41 ....... Done 62 of 70 steps
Copying VRTScscm.tar.gz to sys41 ....... Done 63 of 70 steps
Installing VRTScscm 4.0 on sys41 ....... Done 64 of 70 steps
Copying patch 115209-05 to sys41 ....... Done 65 of 70 steps
Adding patch 115209-05 on sys41 ........ Done 66 of 70 steps
Copying patch 115212-05 to sys41 ....... Done 67 of 70 steps
Adding patch 115212-05 on sys41 ........ Done 68 of 70 steps
Copying patch 115210-05 to sys41 ....... Done 69 of 70 steps
Adding patch 115210-05 on sys41 ........ Done 70 of 70 steps

Cluster Server installation completed successfully.

Press [Return] to continue:


Creating Cluster Server configuration files ..........Done
Copying configuration files to sys42.................Done
Copying configuration files to sys41.................Done
Cluster Server configured successfully.
```

A partir de este momento se inicia el servicio del sistema de alta disponibilidad.

```
Do you want to start Cluster Server processes now? [y,n,q]
(y)

Starting Cluster Server:

Starting LLT on sys42 ..............................Started
Starting LLT on sys41 ..............................Started
Starting GAB on sys42 ..............................Started
Starting GAB on sys41 ..............................Started
Starting Cluster Server on sys42 ...................Started
Starting Cluster Server on sys41 ...................Started
Confirming Cluster Server startup ........2 systems RUNNING

Cluster Server was started successfully.

Press [Return] to continue:
```

Al poner en servicio el sistema, el software manda una mensaje donde se encuentra la bitácora de la instalación

```
Installation of Cluster Server 4.0 has completed
successfully.
The installation summary is saved at:
/opt/VRTS/install/logs/installvcsdate_time.summy
The installvcs log is saved at:
/opt/VRTS/install/logs/installvcsdate_time.log
 The installation response file is saved at:
/opt/VRTS/install/logs/installvcsdate_time.response
```

6.2.6. Administración del cluster

El comando hagrp muestra el estado de los servicios del grupo que se solicite

```
#hagrp -display cluster-unam

#Group     Attribute             System     Value
sys41       AutoFailOver          global     1
sys41       AutoRestart           global     1
sys41       AutoStart             global     1
sys41       AutoStartList         global     sys41
sys41       CurrentCount          global     1
sys41       Evacuating            global     255
sys41       ExtMonApp             global
sys41       ExtMonArgs            global
sys41       FailOverPolicy        global     Priority
sys41       FromQ                 global
sys41       Frozen                global     0
sys41       IntentOnline          global     1
sys41       LastSuccess           global     0
sys41       ManualOps             global     1
sys41       MigrateQ              global
sys41       NumRetries            global     0
sys41       OnlineRetryInterval   global     0
sys41       OnlineRetryLimit      global     0
sys41       Parallel              global     0
sys41       PreOffline            global     0
sys41       PreOnline             global     0
sys41       PrintTree             global     1
sys41       Priority              global     0
sys41       Restart               global     0
sys41       SourceFile            global     ./main.cf
sys41       SystemList            global     sys41 0
sys41       SystemZones           global
```

```
sys41       TFrozen              global     0
sys41       TargetCount          global     1
sys41       ToQ                  global
sys41       TriggerEvent         global     1
sys41       TypeDependencies     global
sys41       UserIntGlobal        global     0
sys41       UserStrGlobal        global
sys41       AutoDisabled         sys41   0
sys41       Enabled              sys41   1
sys41       PreOfflining         sys41   0
sys41       PreOnlining          sys41   0
sys41       ProbesPending        sys41   0
sys41       State                sys41   |ONLINE|
sys41       UserIntLocal         sys41   0
sys41       UserStrLocal         sys41
#
sys42        AutoFailOver         global     1
sys42        AutoRestart          global     1
sys42        AutoStart            global     1
sys42        AutoStartList        global     sys40
sys42        CurrentCount         global     1
sys42        Evacuating           global     255
sys42        ExtMonApp            global
sys42        ExtMonArgs           global
sys42        FailOverPolicy       global     Priority
sys42        FromQ                global
sys42        Frozen               global     0
sys42        IntentOnline         global     1
sys42        LastSuccess          global     0
sys42        ManualOps            global     1
sys42        MigrateQ             global
sys42        NumRetries           global     0
sys42        OnlineRetryInterval  global     0
sys42        OnlineRetryLimit     global     0
sys42        Parallel             global     0
sys42        PreOffline           global     0
sys42        PreOnline            global     1
sys42        PrintTree            global     1
sys42        Priority             global     0
sys42        Restart              global     0
sys42        SourceFile           global     ./main.cf
sys42        SystemList           global     sys40 0 sys41 1
sys42        SystemZones          global
sys42        TFrozen              global     0
sys42        TargetCount          global     1
sys42        ToQ                  global
sys42        TriggerEvent         global     1
```

```
sys42        TypeDependencies     global
sys42        UserIntGlobal        global     0
sys42        UserStrGlobal        global
sys42        AutoDisabled         sys40   0
sys42        AutoDisabled         sys41   0
sys42        Enabled              sys40   1
sys42        Enabled              sys41   1
sys42        PreOfflining         sys40   0
sys42        PreOfflining         sys41   0
sys42        PreOnlining          sys40   0
sys42        PreOnlining          sys41   0
sys42        ProbesPending        sys40   0
sys42        ProbesPending        sys41   0
sys42        State                sys40   |ONLINE|
sys42        State                sys41   |OFFLINE|
sys42        UserIntLocal         sys40   0
sys42        UserIntLocal         sys41   0
sys42        UserStrLocal         sys40
sys42        UserStrLocal         sys41
```

Para verificar el estado del cluster se ejecuta el comando hastatus

```
#hastatus -sum

-- SYSTEM STATE
-- System               State                Frozen

A  sys40             RUNNING              0
A  sys41             RUNNING              0

-- GROUP STATE
-- Group     System     Probed     AutoDisabled     State
```

Ahora procederemos a dar un comando que nos permitira modificar la configuración del sistema

```
#haconf -makerw
```

Una vez que se permite modificar el sistema agregaremos un grupo donde se incluirá el flisystem a compartir.

```
#hagrp  -add grupo-unam
```

Como recordamos nuestro sistema esta constituido por dos sistemas así que cada uno tiene cierta prioridad sobre otro, y con el comando siguiente daremos dicha prioridad

```
#hgrp -modify grupo-unam SystemList sys41 2 sys40 1
```

A continuación agregaremos el recurso que dará servicio al grupo que previamente ya se había configurado.

```
#hares -add aplmercado vx Filesystem grupo-unam
```

El recurso por definición tiene un formato de comportamiento, por lo que se procede a modificar dichos parámetros con los siguientes comandos.

```
#hares -modify aplmercado Critical 0
#hares -modify aplmercado Enabled 1
```

Una vez terminado este conjunto de pasos pondremos el recurso en servicio mejor conocido como en línea, para lo cual ejecutaremos el siguiente comando.

```
#hares -online aplmercado -sys sys41
```

Ahora verificaremos el estado del sistema donde podremos apreciar con claridad las respuestas que se presentan al ejecutar los anteriores comandos.

```
#hares -display -all
```

```
#Resource Attribute        System    Value

aplmercado    Group
aplmercado    Type              global    DiskGroup
aplmercado    AutoStart         global    1
aplmercado    Critical          global    1
aplmercado    Enabled           global    1
aplmercado    LastOnline        global    sys41
aplmercado    MonitorOnly       global    0
aplmercado    ResourceOwner     global    unknown
aplmercado    TriggerEvent      global    0
aplmercado    ArgListValues     sys41  aplmercado 1 1 0
aplmercado    ConfidenceLevel   sys41  100
aplmercado    Flags             sys41
aplmercado    IState            sys41  not waiting
aplmercado    Probed            sys41  1
aplmercado    Start             sys41  1
aplmercado    State             sys41  ONLINE
aplmercado    DiskGroup         global    aplmercado
aplmercado    Name              global
aplmercado    StartVolumes      global    1
```

```
aplmercado    StopVolumes     global    1
# (io flotante)
ifx_IP    Group           global    sai
ifx_IP    Type            global    IP
ifx_IP    AutoStart       global    1
ifx_IP    Critical        global    1
ifx_IP    Enabled         global    1
ifx_IP    LastOnline      global    sys40
ifx_IP    MonitorOnly     global    0
ifx_IP    ResourceOwner   global    unknown
ifx_IP    TriggerEvent    global    0
ifx_IP ArgListValues sys40  qfe2 192.168.30.50 255.255.255.0
"" 1 0
ifx_IP        ArgListValues      sys41    qfe2  199.168.30.50
255.255.255.0 "" 1 0
ifx_IP    ConfidenceLevel sys40  100
ifx_IP    ConfidenceLevel sys41  0
ifx_IP    Flags           sys40
ifx_IP    Flags           sys41
ifx_IP    IState          sys40  not waiting
ifx_IP    IState          sys41  not waiting
ifx_IP    Probed          sys40  1
ifx_IP    Probed          sys41  1
ifx_IP    Start           sys40  1
ifx_IP    Start           sys41  0
ifx_IP    State           sys40  ONLINE
ifx_IP    State           sys41  OFFLINE
ifx_IP    Address         global    192.168.30.50
ifx_IP    ArpDelay        global    1
ifx_IP    Device          global    qfe2
ifx_IP    IfconfigTwice   global    0
ifx_IP    Name            global
ifx_IP    NetMask         global    255.255.255.0
ifx_IP    Options         global
#
ifx_nic   Group           global    sai
ifx_nic   Type            global    NIC
ifx_nic   AutoStart       global    1
ifx_nic   Critical        global    0
ifx_nic   Enabled         global    1
ifx_nic   LastOnline      global    sys40
ifx_nic   MonitorOnly     global    0
ifx_nic   ResourceOwner   global    unknown
ifx_nic   TriggerEvent    global    0
ifx_nic   ArgListValues   sys40  hme0 "" 1 0
ifx_nic   ArgListValues   sys41  hme0 "" 1 0
ifx_nic   ConfidenceLevel sys40  100
```

```
ifx_nic    ConfidenceLevel sys41   100
ifx_nic    Flags           sys40
ifx_nic    Flags           sys41
ifx_nic    IState          sys40   not waiting
ifx_nic    IState          sys41   not waiting
ifx_nic    Probed          sys40   1
ifx_nic    Probed          sys41   1
ifx_nic    Start           sys40   0
ifx_nic    Start           sys41   0
ifx_nic    State           sys40   ONLINE
ifx_nic    State           sys41   ONLINE
ifx_nic    Device          global     hme0
ifx_nic    Name            global
ifx_nic    NetworkHosts    global
ifx_nic    NetworkType     global
ifx_nic    PingOptimize    global     1
#
migdg  aplmercado    Group           global     sai
migdg      Type            global     DiskGroup
migdg      AutoStart       global     1
migdg      Critical        global     1
migdg      Enabled         global     1
migdg      LastOnline      global     sys40
migdg      MonitorOnly     global     0
migdg      ResourceOwner   global     unknown
migdg      TriggerEvent    global     0
migdg      ArgListValues   sys40   migdg 1 1 0
migdg      ArgListValues   sys41   migdg 1 1 0
migdg      ConfidenceLevel sys40   100
migdg      ConfidenceLevel sys41   0
migdg      Flags           sys40
migdg      Flags           sys41
migdg      IState          sys40   not waiting
migdg      IState          sys41   not waiting
migdg      Probed          sys40   1
migdg      Probed          sys41   1
migdg      Start           sys40   1
migdg      Start           sys41   0
migdg      State           sys40   ONLINE
migdg      State           sys41   OFFLINE
migdg      DiskGroup       global     migdg
migdg      Name            global
migdg      StartVolumes    global     1
migdg      StopVolumes     global     1
#
user_mnt   Group           global     sai
user_mnt   Type            global     Mount
```

```
user_mnt  AutoStart       global    1
user_mnt  Critical        global    1
user_mnt  Enabled         global    1
user_mnt  LastOnline      global    sys40
user_mnt  MonitorOnly     global    0
user_mnt  ResourceOwner   global    unknown
user_mnt  TriggerEvent    global    0
user_mnt  ArgListValues   sys40  /usr1 /dev/vx/dsk/aplmercado
vxfs "" "" 0
user_mnt  ArgListValues   sys41  /usr1 /dev/vx/dsk/aplmercado
vxfs "" "" 0
user_mnt  ConfidenceLevel sys40  100
user_mnt  ConfidenceLevel sys41  0
user_mnt  Flags           sys40
user_mnt  Flags           sys41
user_mnt  IState          sys40  not waiting
user_mnt  IState          sys41  not waiting
user_mnt  Probed          sys40  1
user_mnt  Probed          sys41  1
user_mnt  Start           sys40  1
user_mnt  Start           sys41  0
user_mnt  State           sys40  ONLINE
user_mnt  State           sys41  OFFLINE
user_mnt  BlockDevice     global    /dev/vx/dsk/aplmercado
user_mnt  FSType          global    vxfs
user_mnt  FsckOpt         global
user_mnt  MountOpt        global
user_mnt  MountPoint      global    /usr1
user_mnt  Name            global
user_mnt  SnapUmount      global    0
#
usr2_mnt  Group           global    desa
usr2_mnt  Type            global    Mount
usr2_mnt  AutoStart       global    1
usr2_mnt  Critical        global    0
usr2_mnt  Enabled         global    1
usr2_mnt  LastOnline      global    sys41
usr2_mnt  MonitorOnly     global    0
usr2_mnt  ResourceOwner   global    unknown
usr2_mnt  TriggerEvent    global    0
usr2_mnt        ArgListValues             sys41         /usr2
/dev/vx/dsk/desadg/usr2 vxfs "" "" 0
usr2_mnt  ConfidenceLevel sys41  100
usr2_mnt  Flags           sys41
usr2_mnt  IState          sys41  not waiting
usr2_mnt  Probed          sys41  1
usr2_mnt  Start           sys41  1
```

```
usr2_mnt   State          sys41   ONLINE
usr2_mnt   BlockDevice    global     /dev/vx/dsk/desadg/usr2
usr2_mnt   FSType         global     vxfs
usr2_mnt   FsckOpt        global
usr2_mnt   MountOpt       global
usr2_mnt   MountPoint     global     /usr2
usr2_mnt   Name           global
usr2_mnt   SnapUmount     global     0
```

En esta última sección ha sido posible ver los elementos y pasos necesarios para construir un sistema de alta disponibilidad, esta sección puede ser usada para referencia de instalación de un cluster con las marcas de hardware y software usados, es posible que haya variaciones mínimas si es que se utilizan otras versiones de software y diferentes modelos de sistemas de computo y arreglo de discos, es muy probable que lo único necesario será instalar parches para adecuar al hardware y tenga la funcionalidad requerida por el software.

CONCLUSIONES

El proceso evolutivo de los sistemas de cómputo sin duda alguna va ligado a las necesidades crecientes y demandantes de la información. Como se mostró en este trabajo la alta disponibilidad se ha convertido en la característica implícita para el diseño y fabricación tanto de los componentes físicos como lógicos de los sistemas de cómputo. Estas características se adecuan a un esquema de datos para organizaciones empresariales quienes son las que mayor utilidad le han dado.

El punto a destacar para llevar a cabo la alta disponibilidad es la redundancia tanto en hardware como en software, y es una solución que ya se aplicaba desde hace poco más de 20 años en el pilotaje aéreo, considerándose desde aquel entonces como una actividad crítica debido al riesgo que implicaba alguna falla y el riesgo de pérdida de vidas humanas. La redundancia en los sistemas de cómputo es una característica básica para la llevar a cabo el esquema propuesto, se protege al elemento más crítico que son los datos y que corren el riesgo de estar indisponibles debido a alguna falla en cualquier elemento del sistema, se expuso la implantación de un cluster siendo éste el ejemplo mas claro de lo que es la redundancia en los elementos susceptibles a fallar.

La criticidad se ha llevado a campos de la informática convirtiendo a los datos en parte crítica de una empresa debido al impacto de estos en su accionar productivo, pues los datos en su conjunto son la herramienta necesaria para una toma de decisiones, la venta de algún producto o servicio. Esto solo por mencionar algunos ejemplos.

El prescindir de la alta disponibilidad puede llevar riesgos considerables, en pérdidas monetarias como primer punto y en casos extremos la desaparición de la misma organización pues no se considera a este esquema como una inversión y por consiguiente resulta menos costoso. Llega a ser sorprendente leer o escuchar aseveraciones donde se menciona: "un sistema basta con instalarlo y dejar que funcione" lo cual deja de prevenir posibilidades de falla o cambios en las necesidades de cómputo. Es de gran importancia no solo plantear y llevar a cabo planes de mantenimiento, actualizaciones, esquemas de seguridad y documentación a fin de adecuar progresivamente al sistema de cómputo a las necesidades de la organización sino también invertir e implantar la alta disponibilidad para responder a necesidades de información en la actualidad.

El sistema operativo UNIX y en caso especifico Solaris de SUN Microsystems hoy día es el sistema más demandado en el mercado junto con los servidores Sun, pues son quienes por su diseño y desempeño en el manejo de grandes masas de datos y alto procesamiento han ganado el mercado actual de la informática, siendo estos los sistemas más vendidos, los productos de software de Veritas han venido a complementarlos desde hace no más de 10 años a la fecha. Trabajando en conjunto en cuanto a las compatibilidades y con esto dan como resultado los

sistemas de alta disponibilidad más vendidos a nivel empresarial dentro del área de sistemas abiertos.

Con los elementos y características de los sistemas expuestos en el presente trabajo podemos concluir que la alta disponibilidad es y será una característica imprescindible para cubrir la exigencia actual en los sistemas de cómputo obtener información pronta y útil en el lugar y momento que es requerido.

BIBLIOGRAFÍA

AGARWAL, A. Análisis of Cache Performace for Operting Systems and Multiprogramming. Kluwer Academic Publishers, Boston, 1989.

ANDREWS, M. Principlesof Fimware Engineering in Microprogram Control. Computer Science Press, Silver Spring, MD, 1993

ARDEN, B. What Can Be Automated? The MIT Press, Cambridge, MA, 1998.

BANERJI, D., y RAYMOND, J. Elements of Microprogramming. Prentice-Hall, Englewood Cliffs, Nj. 1987.

BLAHUT, R. Theory and Practice of Error Control Code. Addison-Wesley, Reading. MA. 1986.

BRADLEE,D.; EGGERS,S.; y HENRY, R. "The Effect on RISC Perdormace of Register Set Size and Structure Versus Code Generation Strategy." Proceeding, 18th Annual International Symposium on Computer Architecture, mayo de 1991

CLINE, B. Microprogramming Concepts and Techniques. Petrocelli, New York, 1991.

COMER E. "Redes globales de información con internet TCP/IP", Prentice Hall, México 1996.

DATTATREYA, G. "A Systematic Approach to Teaching Binary Arithmetic in a First Course." IEEE Transactions on Education, febrero de 1993.

ECKERT, R. "Communication Between Computers and Peripheral Devices-An Analogy." ACMSIGSCE Bulletin, septiembre de 1994

ENSLOW, P. "Multiprocessor Organization-A Survey." ACM Computing Surverys, marzo de 1987.

ESPONDA, M., Y Rojas, R. "A Graphical Representation of RISC Processors." Computer Architecture News, septiembre de 1999.

GOLDBERG,D."GAT Every Computers Scientist Should Know About Floating-Point Arithmetic."ACM Computing Surveys, marzo de 1991

HAWKINIS M., PIEDAD "High Availability Design, Techniques, and proceses." Prentice Hall, Nj. 2001.

MASSIGLIA, P., ed. The RAIDbook: Asourcebook for Disk Array Technology. The Raid Advisory Board, St Peter, MN, 1994.

MILENKOVIK M. "Sistemas operativos" Prentice Hall, México 1986.

NOVITISKY, J.; AZIMI, M.; y GHAZNAVI, R. "Optimizing Systems Performace Based on Pentium Processors." Processdings COMPCON´92, febrero de 1993

PATTERSON, D.; GIBSON,G.; y KATZ,R." A Case for Redundant Arrays of Inexpensive Disks (RAID)." Processdings, ACM SIGMOD Conference of Management of Data, junio de 1998.

POUNTAIN, D. "Pentium: More RISC than CISC." Byte, Septiembre de 1993.

PRESSMAN R. "Ingeniería del software", McGraw-Hill, España 1996.

PRISTER F. "In search of cluster" , Prentice Hall New Jersey 1998.

STALLINGS W. "Organización y arquitectura de computadoras", Prentice may, Madrid 1997.

STONE, H. High-Performace Computer Architecture. Addison-Wesley, Reading, MA. 1993

CURTIS, P Unix Backup and Recovery. O'Reilly November 1999

COLOURIS, G.; DOLLIMORE, J.; KINDBERG, T. Distributed Systems. Addison-Wesley 1994

PEEK, J.;O'REILLY, T.; LOUKIDES M. Unix Power Tools. O'Reilly 1997

RUSSELL, D.; GONGEMI, G. Computer Security Basics. O'Reilly 1992

SHOOMAN, M. Reliability of Computer Systems and Networks. Wiley 2002

EVAN, M.; HAL, S. Blueprints for High Availability Designing Resilient Distributed Systems. Wiley 2000