



UNIVERSIDAD DEL AZUAY

FACULTAD DE ADMINISTRACIÓN DE EMPRESAS

ESCUELA DE INGENIERÍA DE SISTEMAS

Balanceo de Carga en Servidores de Correo Electrónico bajo el mismo Dominio

**MONOGRAFÍA PREVIA A LA OBTENCIÓN DEL TÍTULO
DE**

INGENIERO DE SISTEMAS

**AUTORES: FELIPE ALFREDO QUINDE MARTÍNEZ
DIEGO ARMANDO QUITO RODRÍGUEZ**

DIRECTOR: PABLO PINTADO ZUMBA - MBA

CUENCA, ECUADOR

2007

DEDICATORIA

A mis padres,
por ser la fuerza que me impulsa a
seguir adelante y conseguir mis metas.

Diego Quito Rodríguez

Esta monografía está dirigida a mi familia
que con apoyo constante me han ayudado
a culminar mis estudios, especialmente a mis padres
que con mucho esfuerzo me han dado todo el cariño
y lo que necesito para poder culminar mis metas.

Felipe Quinde Martínez

AGRADECIMIENTOS

Agradezco a Dios por haberme permitido alcanzar este gran objetivo, y a toda mi familia y personas que estuvieron conmigo a lo largo de este proceso.

Diego Quito Rodríguez

Quisiera agradecer por haber culminado mis estudios y la elaboración de esta monografía a mis padres que son quienes me han ayudado y apoyado en todos estos años de estudio.

Felipe Quinde Martínez

ÍNDICE DE CONTENIDOS

Dedicatoria	ii
Agradecimientos	iii
Índice de Contenidos	iv
Índice de Ilustraciones y Cuadros	vii
Resumen	viii
Abstract	ix
Introducción	1
Capítulo 1: Balanceo de Carga	2
1.1 Introducción	2
1.2 Tipos de Balanceo de Carga	3
1.2.1 Balanceo de Carga en Conexiones de Internet	3
1.2.2 Balanceo de Carga en Servidores Web	5
1.2.2.1 Clusters	6
1.2.2.1.1 ¿Cómo funciona un Cluster?	10
1.2.2.1.2 Cluster con Balanceo de Carga	11
1.2.2.1.3 Cluster de Alta Disponibilidad	13
1.2.3 Balanceo de Carga en Servidores de Correo Electrónico ...	16
1.3 Software para Balanceo de Carga	17
1.4 Beneficios del Balanceo de Carga	18
1.5 Conclusiones	18
Capítulo 2: Linux como Plataforma de Implementación	19
2.1 Introducción	19
2.2 Historia	22
2.3 ¿Qué es Linux?	23
2.3.1 ¿Qué significa Código Abierto?	24
2.4 Distribuciones	26
2.5 Servidores en Linux	29

2.6 Ventajas de Linux	32
2.7 Desventajas de Linux	33
2.8 Conclusiones	34
 Capítulo 3: Servidores DNS	 35
3.1 Introducción	35
3.2 ¿Cómo trabaja DNS?	38
3.2.1 Componentes	39
3.2.2 Las partes de un Nombre de Dominio	39
3.2.3 Tipos de Servidores DNS	40
3.2.4 Tipos de Resolución de Nombres de Dominio	41
3.3 Tipos de Registros DNS	45
3.4 Dominios Virtuales	46
3.5 Conclusiones	47
 Capítulo 4: Sendmail	 48
4.1 Introducción	48
4.2 Conceptos	50
4.3 Funcionamiento del Correo Electrónico	52
4.4 El procesamiento de los mensajes	56
4.4.1 Envíos Locales	56
4.4.2 Envíos Remotos	57
4.5 Cuentas en el Sistema	57
4.6 Conclusiones	58
 Capítulo 5: Implementación del Balanceo de Carga en Servidores de Correo Electrónico	 59
5.1 Introducción	59
5.2 Configurando el DNS	60
5.3 Configurando Sendmail	65

5.4 Implementación del Balanceo de Carga en el servidor de Correo Electrónico	70
5.5 Pruebas	72
5.6 Conclusiones	77
Conclusiones y Recomendaciones	77
Glosario	80
Bibliografía	89

ÍNDICE DE ILUSTRACIONES

Figura 1	Arquitectura del DNS	41
Figura 2	Funcionamiento del Correo Electrónico	54
Figura 3	Ejemplo de Cliente de Correo Electrónico.....	76

RESUMEN

El proyecto consiste en configurar dos servidores de correo que sirvan al mismo dominio en dos máquinas diferentes, de tal forma que las cuentas de correo de los usuarios se encuentren distribuidas, así la información no será soportada por un solo servidor, sino que será administrada por dos máquinas, permitiéndonos que la carga de correo esté balanceada en los servidores.

A través de los archivos de configuración del servidor de correo y servidor DNS, implementaremos un método que nos permitirá manejar la información distribuida en los dos servidores de manera transparente al usuario, de tal forma que un servidor sea quien reciba el envío de correo, y este sea quien lo redirija al servidor adecuado que contenga almacenada la información de las cuentas de correo de los clientes.

ABSTRACT

This Project consists in configuring two mail servers for the same domain in two different machines in order to distribute users' mail accounts. That way, the information will not be supported by a single server, but it will be administered by two machines, allowing the mail load to be balanced in the servers.

Through the configuration files of both the mail server and DNS server, we will implement a method that will allow us to handle the information distributed in the two servers in a transparent way for the user so that the server who receives the mail will readdress it to the adequate server that has stored the information of the customers' mail accounts.

INTRODUCCIÓN

La utilización de correo electrónico en una empresa es de vital importancia para el funcionamiento de la misma, de tal forma que esta información crece de manera acelerada, siendo necesario que las tareas de almacenamiento y administración de correo electrónico que soporta un solo servidor, sean compartidas por varios servidores para ganar un mayor rendimiento.

Muchas empresas ofrecen soluciones de balanceo de carga en servidores de correo electrónico que son costosas, y que en gran parte no están al alcance del presupuesto manejado por una institución, por ello es necesario la implementación de nuevos métodos que permitan hacer las mismas tareas que realizan las herramientas ofrecidas por estas empresas a un bajo costo, y que garantice un buen rendimiento de lo equipos, siendo esta la razón principal por la cual vamos a incursionar en el balanceo de carga sobre servidores de correo electrónico en este proyecto.

CAPITULO 1

BALANCEO DE CARGA

1.1. Introducción

En este primer capítulo introduciremos los conceptos más importantes sobre una aplicación con balanceo de carga, con el fin de comprender los beneficios que nos brinda esta técnica aplicada en los diferentes tipos de servidores, y al mismo tiempo tener una noción clara sobre las ventajas que nos brindaría un servidor de correo implementado con un repartidor de carga.

El balance o balanceo de carga es un concepto usado en informática que se refiere a la técnica usada para compartir el trabajo a realizar entre varios procesos, ordenadores, discos u otros recursos. Está íntimamente ligado a los sistemas de multiprocesamiento, o que hacen uso de más de una unidad de procesamiento para realizar labores útiles.

En servicios WEB, el balanceo de carga consiste en utilizar dos o más servidores con configuraciones replicadas capaces de realizar el mismo tipo de trabajo, de modo que un balanceador se ocupa de redirigir el tráfico entrante hacia uno de los servidores disponibles.

Esta distribución se hace de una forma “inteligente” dependiendo de una serie de parámetros como la capacidad, la disponibilidad (ante el fallo o el paro por mantenimiento de uno de los servidores), la carga del servidor, el tiempo de respuesta, el rendimiento histórico o el peso que el administrador le quiera dar a cada uno de los servidores.

Esto nos ofrece beneficios como la escalabilidad y la alta disponibilidad del servicio, que será mayor obviamente al aumentar el número de servidores. Además de asegurar que el servicio se realiza con el mejor tiempo de respuesta posible y la utilización más adecuada de los recursos. Esta optimización de los recursos se traduce de hecho en un ahorro, ya que son necesarios menos equipos para ofrecer el mismo servicio. Por ejemplo, si una empresa ya dispone actualmente de tres servidores dedicados,

balanceándolos puede dar un servicio equivalente a añadir un servidor adicional independiente.

El balanceo de carga simula un único servicio mediante varias máquinas que lo prestan de modo que toda la carga del servicio se reparte entre las máquinas logrando un mayor rendimiento, además de maximizar las capacidades de procesos de datos, así como de ejecución de tareas, por ello es bastante común montar balanceo de carga entre dos o más servidores Web, correo electrónico entre otros.

Para el desarrollo del presente capítulo se utilizará la siguiente bibliografía:

http://es.wikipedia.org/wiki/Balance_de_carga

http://www.terra.es/personal3/fistror/linux/Balanceo_adsl_v1.1.pdf

<http://www.esdebian.org/staticpages/index.php?page=20040921021605992>

<http://talika.eii.us.es/~rosa/clustering.pdf>

<http://www.linalco.com/balanceo.html>

1.2. Tipos de Balanceo de Carga

Dentro de las diversas técnicas difundidas para reducir el trabajo y carga que tienen los equipos ó recursos utilizados dentro de una red de ordenadores, se encuentra el balanceo de carga en conexiones de Internet, balanceo de carga en servidores Web, balanceo de carga en servidores de correo electrónico, que puede ser implementado en diferentes empresas para sacarle el máximo provecho a los recursos tecnológicos que poseen.

1.2.1. Balanceo de Carga en Conexiones de Internet

La proliferación del uso de Internet es cada vez mayor y como consecuencia de ello la necesidad de ancho de banda a precios asequibles y con una alta disponibilidad. Por ello, es cada vez más difícil encontrar una empresa u hogar que no disponga de conexión, con un ancho de banda cada vez mayor y en muchos casos con conexión permanente a la misma.

Sin embargo no en todos los casos se puede tener a disposición una línea de conexión a Internet "profesional" y en otros casos lo que se quiere es disponer de una redundancia en el enlace pudiendo tener conexión en el caso de que un proveedor tenga caído el servicio.

Como se da el caso de una mediana empresa, que dispone de una conexión a Internet basada en un ADSL (Asymmetric Digital Subscriber Line, Línea de Abonado Digital Asimétrica) de 2 Mbps. Con un uso elevado de la misma en la que además no quiere que las descargas de FTP (File Transfer Protocol, Protocolo de Transferencia de Archivos) de sus empleados acaben haciendo cada vez más lento la navegación de los directivos de la misma, aquí podemos ver que el balanceo entre dos ADSL's puede ser una solución válida.

No solo puede utilizar dos conexiones de forma simultanea o incremental, también podrá asignar determinados puertos y servicios a la conexión que considere más óptima para los mismos. Un ejemplo, si mezcla dos tipos de conexiones puede desviar los servicios de telefonía IP a la que tenga menores latencias, y desviar el grueso de la navegación Web a la que sea menos crítica.

Al realizar el balanceo de carga en conexiones de Internet es necesario el ajuste de las rutas que deben ser estáticas, como por ejemplo, todos los paquetes que vayan redirigidos a la LAN (Local Area Network, Red de Área Local), deben ser enrutados siempre por la misma interfaz o habrá determinadas cosas que podemos querer que vengán siempre por determinado interfaz. Cada LAN tendrá sus propias necesidades, como puede darse el caso en que, los servidores DNS (Domain Name System) no están en el mismo segmento de red y se tiene la necesidad de enrutar las peticiones, específicas a los DNS por otro router distinto del que se tiene por defecto.

Al aplicar el balanceo de carga en conexiones de Internet se puede obtener varias ventajas como:

Los usuarios y servicios podrán salir a Internet con una sola IP o con dos, y dispondrá de filtros de acceso a su red local desde el exterior, administración remota segura,

enrutamientos estáticos y dinámicos, servidores virtuales y SNMP (Simple Network Management Protocol, protocolo simple de gestión de redes).

1.2.2. Balanceo de carga en servidores Web

Uno de los principales problemas de los mayores sitios Web en Internet es cómo gestionar las solicitudes de un gran número de usuarios, se trata de un problema de escalabilidad que surge con el continuo crecimiento del número de usuarios.

Esto trae consigo el descontento de los usuarios del sitio, es decir una pérdida en la calidad de servicio, en términos ingenieriles típicamente depende de dos parámetros, la *velocidad de transmisión de la red* y el *tiempo de respuesta del servidor*. Por una parte la velocidad de transmisión de la red depende casi completamente del ancho de banda del enlace y de su disponibilidad, por otra parte el tiempo de respuesta del servidor depende de otros recursos como la velocidad de la CPU (especialmente para programas CGI, Common Gateway Interface, Pasarela de Interfaz Común), memoria RAM, y una buena performance en entrada/salida (I/O) especialmente para discos y tráfico de red.

Se puede instalar más RAM en las maquinas existentes, o reemplazar las CPU por otras más veloces, o instalar controladores SCSI (Small Computer System) y discos con menores tiempos de acceso, quizás sistemas RAID (Redundant Array Of Independent/Inexpensive Disks) con un caché inmenso, optimizar el software, ya sea el sistema operativo o el software del servicio Web, todo esto para mejorar el performance. O se puede utilizar una aproximación alternativa, es decir mejorar el performance aumentando el número de servidores Web, esto involucra distribuir el tráfico en un grupo (cluster) de servidores Web. Aparte del reto técnico esta aproximación es interesante por que con ella los servidores Web no necesitan ser máquinas de gran escala.

El balanceo de carga no solamente protege contra caídas de servidores, evita que picos de acceso, como por ejemplo los generados por campañas publicitarias, afecten el normal funcionamiento de tu sitio.

Además, permite optimizar mejor todos los recursos disponibles dando como resultado una navegación más rápida y estable.

1.2.2.1. Clusters

En muchas ramas de la ciencia, la complejidad de los problemas que se estudian requiere contar con acceso a una máquina que sea capaz de desarrollar varios miles de millones de operaciones por segundo. Las supercomputadoras tradicionales emplean procesamiento en paralelo contienen matrices de microprocesadores muy rápidos que trabajan en sincronía para resolver problemas complejos como pronósticos numéricos del estado del tiempo, o modelar estructuras complejas de la materia. Los fabricantes de supercomputadoras como Cray, IBM, Silicon Graphics, entre otros, producen modelos por diseño especial que cuestan decenas de millones de dólares, precios que van más allá de los presupuestos de inversión de la inmensa mayoría de los grupos de investigación.

En los últimos años, el personal académico de diversas universidades y centros de investigación se ha dado a la tarea de aprender a construir sus propias “supercomputadoras” conectando computadoras personales y desarrollando software para enfrentar tales problemas extraordinarios.

En el ámbito comercial, por otra parte, es creciente la necesidad de mantener los servicios ofrecidos siempre disponibles y con un tiempo de servicio competitivo. Ante esta necesidad, son pocas las empresas que pueden afrontar la inversión necesaria para adquirir una supercomputadora sin puntos de fallo y con capacidad de cómputo alta para servir a gran velocidad.

Dos conceptos muy manejados al tratar con tecnologías de clustering son los de escalabilidad vertical y escalabilidad horizontal. Hablamos de escalabilidad vertical cuando tratamos con un gran sistema, con múltiples CPUs, con una gran velocidad de interconexión entre nodos, tradicionalmente empleando elementos tales como backplanes (Un backplane es un componente de computadoras que facilita la interconexión de los módulos o celdas que componen la computadora).

La mayoría de las aplicaciones escalan muy bien verticalmente hasta cierto número de CPUs. Actualmente, debido a su alto precio, solo se emplea en aplicaciones muy transaccionales y/o minería de datos. El problema de estas máquinas es que suponen una gran inversión que pocas organizaciones pueden afrontar, además de los problemas de escalabilidad que provocan.

La escalabilidad horizontal consiste en emplear muchas máquinas pequeñas e interconectadas para realizar una tarea determinada con un coste proporcionalmente bajo por máquina, es una solución muy barata para algunos tipos de aplicaciones, como servicios Web, correo o supercomputación. Su principal ventaja es que el crecimiento a un mayor número de nodos se puede hacer cuando sea necesario, sin tener que realizar un gran desembolso inicialmente como ocurre en un sistema vertical. Las principales desventajas de un sistema horizontal son que escalan mal para aplicaciones transaccionales (bases de datos, por ejemplo), y que la administración de muchas máquinas interconectadas implica una mayor atención y conocimiento que la administración de una gran máquina con muchos procesadores.

El término cluster se aplica no sólo a computadoras de alto rendimiento sino también a los conjuntos de computadoras, construidos utilizando componentes de hardware comunes y software libre. Entre estos dos extremos situaremos también las soluciones comerciales modulares, que permiten ajustar la inversión a los requerimientos concretos del problema al que nos enfrentemos y expandir el cluster conforme las necesidades vayan aumentando. Estas soluciones, sin embargo, van a imponer modelos específicos de programación que permitan explotar al máximo las capacidades de la máquina y por otra parte, encadenan al comprador a un fabricante concreto. El estudio de la tecnología de clustering dentro del entorno del software libre, sobre todo en el uso del sistema operativo GNU(es un acrónimo recursivo que significa "GNU No es Unix")/Linux y otras herramientas libres asociadas a él nos permiten evitar este problema. Aunque, hemos de señalar que se han presentado soluciones de clustering también para otros sistemas operativos libres (FreeBSD, NetBSD).

Los clusters juegan hoy en día un papel muy importante en la solución de problemas de las ciencias, las ingenierías y en el desarrollo y ejecución de muchas aplicaciones comerciales, los clusters han evolucionado para apoyar actividades en aplicaciones que van desde la supercomputación hasta el software adaptado a misiones críticas, pasando por los servidores Web, el comercio electrónico y las bases de datos de alto rendimiento.

La computación basada en clusters surge gracias a la disponibilidad de microprocesadores de alto rendimiento más económicos y de redes de alta velocidad, y también gracias al desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, todo ello frente a la creciente necesidad de potencia de cómputo para aplicaciones en las ciencias y en el ámbito comercial, así como de disponibilidad permanente para algunos servicios.

Por otro lado, la evolución y estabilidad que ha alcanzado el sistema operativo GNU/Linux, ha contribuido de forma importante, al desarrollo de muchas tecnologías nuevas, entre ellas las de clustering. Según la aplicabilidad de los clusters, se han desarrollado diferentes líneas tecnológicas, la primera surge frente a la necesidad de supercomputación para determinadas aplicaciones, lo que se persigue es conseguir que un gran número de máquinas individuales actúen como una sola máquina muy potente, este tipo de clusters se aplica mejor en problemas grandes y complejos que requieren una cantidad enorme de potencia computacional.

Entre las aplicaciones más comunes de clusters de alto rendimiento (computacionales, de supercomputación) se encuentra el pronóstico numérico del estado del tiempo, astronomía, investigación en criptografía, simulación militar, simulación de recombinaciones entre moléculas naturales y el análisis de imágenes.

Un segundo tipo de tecnología de clusters, es el destinado al balanceo de carga, surge el concepto de "cluster de servidores virtuales", cluster que permite que un conjunto de servidores de red compartan la carga de trabajo y de tráfico de sus clientes, aunque aparezcan para estos clientes como un único servidor. Al balancear la carga de trabajo en un conjunto de servidores, se mejora el tiempo de acceso y la confiabilidad, además

como es un conjunto de servidores el que atiende el trabajo, la caída de uno de ellos no ocasiona una caída total del sistema. Este tipo de servicio es de gran valor para compañías que trabajan con grandes volúmenes de tráfico y trabajo en sus Webs, servidores de correo, etc.

Hemos de pensar que la imagen y el prestigio de una empresa que ofrece sus servicios por Internet se compromete en la velocidad, la calidad y la disponibilidad de estos servicios.

El último tipo importante de tecnología de clustering trata del mantenimiento de servidores que actúen entre ellos como respaldos de la información que sirven. Este tipo de clusters se conoce como "clusters de alta disponibilidad" o "clusters de redundancia".

La flexibilidad y robustez que proporcionan este tipo de clusters, los hacen necesarios en ambientes de intercambio masivo de información, almacenamiento de datos sensibles y allí donde sea necesaria una disponibilidad continua del servicio ofrecido.

Además del concepto de cluster, existe otro concepto más amplio y general que es el Cómputo en Malla (Grid Computing). Una Malla (Network Of Workstation) es un tipo de sistema paralelo y distribuido que permite compartir, seleccionar y añadir recursos que se encuentran distribuidos a lo largo de dominios administrativos "múltiples". Si los recursos distribuidos se encuentran bajo la administración de un sistema central único de programación de tareas, entonces nos referiremos a un cluster.

En un cluster, todos los nodos trabajan en cooperación con un objetivo y una meta común y la asignación de recursos la lleva a cabo un solo administrador centralizado y global. En una malla, cada nodo tiene su propio administrador de recursos y política de asignación.

1.2.2.1.1. ¿Cómo funciona un Cluster?

Desde un punto de vista general, un cluster consta de dos partes. La primera es el software: un sistema operativo confeccionado especialmente para esta tarea (por ejemplo un kernel Linux modificado), compiladores y aplicaciones especiales, que permite que los programas que se ejecutan en el sistema exploten todas las ventajas del cluster.

En el entorno de GNU/Linux hay que destacar la PVM (Paralell Virtual Machine) y la MPI (Message Passing Interface), librerías que abstraen la componente hardware de la componente software.

La segunda componente es la interconexión hardware entre las máquinas (nodos) del cluster. Se han desarrollado interfaces de interconexión especiales muy eficientes, pero es común realizar las interconexiones mediante una red Ethernet dedicada de alta velocidad, gracias a esta red de interconexión los nodos del cluster intercambian entre sí las tareas, las actualizaciones de estado y los datos del programa. En un cluster abierto, existirá una interfaz de red que conecte al cluster con el mundo exterior (Internet).

Cuando se trata de resolver un problema en paralelo, el software debe ser capaz de dividir el problema en tareas más pequeñas, repartirlas entre los nodos y elaborar los resultados, puesto que las subtareas van a ejecutarse en paralelo se consigue un aumento de velocidad, aunque hay que tener en cuenta el retardo que la división, reparto y transmisión de mensajes (resultado, coherencia, estados) supone.

En el caso de los clusters de balanceo de carga, el hardware y el software deben actuar conjuntamente para que el tráfico se distribuya entre los nodos del cluster. De esta forma, se pueden ofrecer los servicios a mayor velocidad o se realiza una tarea más rápidamente.

Los servidores de un cluster de alta disponibilidad normalmente no comparten la carga de procesamiento que tiene un cluster de alto rendimiento, tampoco comparten la carga de tráfico como lo hacen los clusters de balanceo de carga, su función es la de estar

preparados para entrar inmediatamente en funcionamiento en caso de que falle algún otro servidor.

1.2.2.1.2. Cluster con Balanceo de Carga

Con el crecimiento de Internet en los últimos años el tráfico en la red ha aumentado de forma radical y con él la carga de trabajo que ha de ser soportada por los servidores, especialmente por los servidores Web. Para soportar estos requerimientos hay dos soluciones, o bien el servidor se basa en una máquina de altas prestaciones, que a largo plazo probablemente quede obsoleta por el crecimiento de la carga, o bien se encamina la solución a la utilización de la tecnología de clustering para mantener un cluster de servidores. Cuando la carga de trabajo crezca, se añadirán más servidores al cluster.

Hay varias formas de construir un cluster de balanceo de carga. Una solución basada en mantener varios servidores aunque no se trate de un cluster propiamente es utilizar un balanceo de carga por DNS. En este caso, se asigna un mismo nombre a distintas direcciones IP y se realiza un roundrobin (es uno de los algoritmos de planificación de procesos más simples dentro de un sistema operativo que asigna a cada proceso una porción de tiempo equitativa y ordenada, tratando a todos los procesos con la misma prioridad) a nivel de DNS entre ellas. En una situación ideal la carga se repartirá equitativamente entre los distintos servidores.

Sin embargo, los clientes “cachean” los datos del DNS, con lo que la carga no va a repartirse equitativamente y quedaría desbalanceada. Además, aún cuando el balanceo de los accesos de los clientes a los servicios se haga de forma balanceada, estos clientes pueden solicitar cargas muy distintas de trabajo al servidor al que se conectan, mientras un cliente puede querer tan sólo ver una página, otro puede solicitar un buen número de ellas, por otra parte, si un servidor se cae, se van a seguir redirigiendo peticiones a él.

Una solución mejor es utilizar un balanceador de carga para distribuir ésta entre los servidores de un cluster. En este caso el balanceo queda a nivel de conexión, con una

granularidad más fina y con mejores resultados. Además, se podrán enmascarar más fácilmente las posibles caídas de los nodos del cluster.

El balanceo de carga puede hacerse a dos niveles: de aplicación y a nivel IP. Sin embargo, el balanceo a nivel de aplicación puede provocar efectos de cuello de botella si el número de nodos es grande. Cada solución debe elegirse en función de las necesidades del problema al que hacemos frente.

El proyecto Linux Virtual Server (LVS) ofrece parches y aplicaciones de mantenimiento y gestión que permiten construir un cluster de servidores que implementa alta disponibilidad y balanceo de carga sobre el sistema operativo GNU/Linux.

El sistema aparece al usuario externo como un único servidor (en realidad, como un único servidor virtual). Cada uno de los servidores reales que forman el cluster, es controlado por un nodo director que se encarga de realizar el balanceo de carga. Este director corre un sistema operativo GNU/Linux con un kernel parcheado para incluir el código ipvs, que es la herramienta más importante ofrecida por el proyecto LVS. El director puede ser entendido en términos generales como un router con un conjunto concreto de reglas de enrutamiento.

Cuando un cliente solicita un servicio al servidor virtual, el director escoge un servidor real para que lo ofrezca, desde ese momento el director debe mantener la comunicación entre el cliente y el servidor real. Esta asociación entre cliente y servidor real va a durar sólo lo que dure la conexión TCP (Transmission Control Protocol, protocolo de control de transmisión) establecida, cuando se inicie una nueva conexión el director escogerá de nuevo un servidor real que puede ser distinto del anterior. Así, si el servicio ofrecido es una página Web, el cliente podría obtener las imágenes o los textos desde distintos servidores reales ocultos por el servidor virtual.

Con esta arquitectura, además del balanceo de carga, estamos permitiendo que los servidores reales individuales puedan ser extraídos del LVS, actualizados o reparados y

devueltos al cluster sin interrumpir la prestación de servicio. Asimismo, el sistema es fácilmente escalable. La alta disponibilidad en LVS se diseña utilizando software Mon, Heartbeat, Fake y Coda, que se utilizan para gestionar la alta disponibilidad y para mantener una gestión segura, la comunicación entre máquinas y un sistema de ficheros tolerante a fallos, respectivamente.

1.2.2.1.3. Cluster de Alta Disponibilidad

La alta disponibilidad ha sido tradicionalmente un requerimiento exigido a aquellos sistemas que realizaban misiones críticas. Sin embargo, actualmente, está siendo cada vez más importante exigir esta disponibilidad en sistemas comerciales y en áreas académicas donde el objetivo de prestar los servicios en el menor tiempo posible es cada vez más perseguido.

El concepto de cluster de disponibilidad continua, se basa en la idea de mantener la prestación del servicio en todo momento. Esto representa una situación ideal, sería necesario que el sistema estuviera compuesto de componentes perfectos que no fallaran nunca, tanto en hardware como en software.

Realmente no hay sistemas que puedan asumir este tipo de disponibilidad, necesitamos que el cluster sea tolerante a los fallos, en este caso se encubre la presencia de los fallos del sistema empleando redundancia en el hardware, el software e incluso redundancia temporal.

La redundancia en el hardware consiste en añadir componentes replicados para encubrir los posibles fallos, la redundancia software incluye la administración del hardware redundante para asegurar su correcto funcionamiento al hacer frente a la caída de algún elemento.

La redundancia en el tiempo hace referencia a la repetición de la ejecución de un conjunto de instrucciones para asegurar el comportamiento correcto en caso de que ocurra un fallo.

Definiremos un cluster de alta disponibilidad como un sistema capaz de encubrir los fallos que se producen en él para mantener una prestación de servicio continua, en este caso nos centraremos en los clusters de este tipo que utilizan componentes hardware de forma redundante y software capaz de unir estos componentes y enmascarar los fallos de manera que los servicios ofrecidos al usuario no sean interrumpidos.

El cluster de alta disponibilidad va a poder diseñarse con distintas configuraciones, una posible configuración es activopasivo es decir las aplicaciones se ejecutan sobre un conjunto de nodos, los activos, mientras que los nodos restantes actúan como backups redundantes para los servicios ofrecidos, en el otro extremo, tenemos otra posible configuración que es activoactivo, en este caso, todos los nodos actúan como servidores activos de una o más aplicaciones y potencialmente como backups para las aplicaciones que se ejecutan en otros nodos, cuando un nodo falla, las aplicaciones que se ejecutaban en él se migran a uno de sus nodos backup. Esta situación podría producir una sobrecarga de los nodos de respaldo, con lo que se ejecutarían las aplicaciones con más retardo.

Sin embargo, es aceptable una degradación del servicio y también suele ser preferible a una caída total del sistema. Otro caso particular de cluster de alta disponibilidad sería el de un cluster de un solo nodo, tratándose en este caso, simplemente, de evitar los puntos únicos de fallo.

Los conceptos de alta disponibilidad y de clustering están íntimamente relacionados ya que el concepto de alta disponibilidad de servicios implica directamente una solución mediante clustering. La principal prestación de un sistema de alta disponibilidad es que el fallo de un nodo derive en que las aplicaciones que se ejecutaban en él sean migradas

a otro nodo del sistema. Este migrado puede ser automático (failover) o manual (switchover).

Generalmente este tipo de cluster integra un número relativamente pequeño de nodos (entre 2 y 8), aunque existen soluciones comerciales que trabajan en clusters de mayor tamaño. En este caso, la escalabilidad no venía siendo un objetivo prioritario, en contraste con el caso de los clusters computacionales, las aplicaciones usadas para el diseño y la implementación de este tipo de clusters no tienen porqué escalar, sin embargo, actualmente, existe una cierta tendencia a perseguir la escalabilidad también en los clusters de alta disponibilidad.

Cada vez se busca más tener un cluster de mayor número de nodos pero más pequeños en tamaño y prestaciones, desde un punto de vista general, una solución de alta disponibilidad consiste en dos partes: la infraestructura de alta disponibilidad y los servicios.

La infraestructura consiste en componentes software que cooperan entre sí para permitir que el cluster aparezca como un único sistema. Sus funciones incluyen monitorizar los nodos, los procesos de interrelación entre nodos, controlar el acceso a los recursos compartidos y asegurar la integridad de los datos y la satisfacción de los requerimientos del usuario. La infraestructura debe proveer de estas funcionalidades cuando el cluster está en estado estable y, lo que es más importante, cuando algunos nodos dejan de estar operativos.

Los servicios de alta disponibilidad son clientes de la infraestructura, y usan las facilidades que exporta ese nivel para mantener en todo momento el servicio a los usuarios, normalmente hay una degradación del sistema cuando algún nodo cae, pero no una interrupción del servicio. Los servicios que se mantienen típicamente sobre este tipo de clusters son las bases de datos, los sistemas de ficheros, los servidores de correo y los servidores Web, y en general, serán utilizados para tareas críticas o servicios ofrecidos

por una empresa, grupo de investigación o institución académica, que no puedan, por sus características concretas, interrumpir el servicio.

La adaptación más común que debe sufrir una aplicación para poder ser ejecutada en un cluster de alta disponibilidad implementado sobre GNU/Linux, es añadir scripts. Existen APIs (Application Programming Interface, Interfaz de Programación de Aplicaciones), para trabajar cómodamente con alta disponibilidad, todas ellas incluyen métodos que permiten el switchover y el failover y que permiten arrancar, parar o monitorizar una aplicación por mencionar algunas de sus funcionalidades.

La infraestructura puede ser parte del núcleo del sistema operativo o puede ser una capa situada encima, integrar la infraestructura en el sistema operativo tiene como ventaja la eficiencia y la facilidad de uso.

La principal ventaja de una capa separada es que se independiza la solución de alta disponibilidad del sistema operativo, con lo que resulta más portable. En cualquier caso el sistema debe tener la capacidad de aparecer como un único sistema.

En caso de un cluster de alta disponibilidad los elementos más importantes para asegurar esta apariencia única son el sistema de ficheros global, dispositivos globales y la red global.

Los clusters de alta disponibilidad permiten un fácil mantenimiento de servidores. Una máquina de un cluster de servidores se puede sacar de línea, apagarse y actualizarse o repararse sin comprometer los servicios que brinda el cluster. Cuando el servidor vuelva a estar listo, se reincorporará y volverá a formar parte del cluster.

1.2.3. Balanceo de Carga en Servidores de Correo Electrónico

El balanceo de carga en servidores de correo es una alternativa muy interesante para aquel equipo que soporta toda la información de correo de una empresa.

Esta técnica puede realizarse mediante la implementación de otro servidor de correo que sirva al mismo dominio de tal forma que la carga quede compartida entre dos equipos, permitiéndonos utilizar de una forma más eficiente los recursos de los servidores.

Con esto mejoramos la administración de información de correo de la empresa ya que las cuentas quedan divididas o sectorizadas a criterio del administrador, es decir se puede repartir la carga de correo del departamento de ventas al servidorA, la información del departamento de contabilidad al servidorB, etc. Además el tráfico que soporta cada servidor es mucho menor con lo cual ganamos más rapidez al momento de enviar un correo.

1.3. Software para Balanceo de Carga

Uno de los proyectos más conocidos en este campo es el Linux Virtual Server Project (LVS). Usa balanceo de carga para pasar las peticiones a los servidores y puede virtualizar casi cualquier servicio TCP o UDP (User Datagram Protocol, Protocolo de Datagrama a Nivel de Usuario), tales como HTTP (Hyper Text Transfer Protocol, Protocolo de Transferencia de Hipertexto) (S), DNS, POP (Post Office Protocol, Protocolo de Oficina de Correos), IMAP (Internet Message Access Protocol, Protocolo de acceso a mensajes de correo), SMTP(Simple Mail Transfer Protocol, Protocolo Simple de Transferencia de Correo Electrónico), y FTP. Muchos proyectos de balanceo de carga están basados en LVS (UltraMonkey, Piranha, Keepalived).

El Zeus Load Balancer no está basado en LVS pero ofrece una funcionalidad similar. Pen es otro proyecto no basado en LVS y que es un simple balanceador de carga para servicios basados en TCP como HTTP o SMTP.

Por último, el balanceo de carga y software de monitorización de Turbolinux Cluster Server permiten la detección y recuperación de errores de hardware y software (si la recuperación es posible).

1.4. Beneficios del Balanceo de Carga

Las principales ventajas de la utilización de un balanceador de carga son las siguientes:

- Mayor eficiencia en los servidores.
- Mayor disponibilidad de los recursos de los equipos.
- Mayor capacidad de administración de la información.
- Disminución del tráfico hacia los servidores.

1.5. Conclusiones

Al concluir este capítulo tenemos ya una idea clara sobre las ventajas que nos puede brindar un servidor implementado con balanceo de carga, además de comprender como operan muchas de las técnicas de balanceo expuestas en este apartado, para su posterior implementación en un servidor de correo.

El balanceo de carga tiene como punto fuerte el compartir las tareas que tiene que realizar un solo equipo, logrando un mayor rendimiento en los servidores y de esta forma conseguir un muy buen funcionamiento de los equipos que prestan servicios Web.

CAPITULO 2

LINUX COMO PLATAFORMA DE IMPLEMENTACIÓN

2.1. Introducción

En este capítulo trataremos algunos puntos con el objetivo de tener un concepto general de lo que es Linux como plataforma de implementación para el balanceo de carga en servidores de correo, conoceremos un poco de su historia, cómo se maneja las licencias para el software libre y el código abierto, las diferentes distribuciones, lo robusto que son sus servidores, además de la ventajas y desventajas de este sistema operativo que hoy en día está tomando mucha fuerza.

Linux es la denominación de un sistema operativo y el nombre de un núcleo. Es uno de los paradigmas del desarrollo de software libre (y de código abierto), donde el código fuente está disponible públicamente y cualquier persona, con los conocimientos informáticos adecuados, puede libremente estudiarlo, usarlo, modificarlo y redistribuirlo.

El término *Linux* estrictamente se refiere al núcleo Linux, pero es más comúnmente utilizado para describir al sistema operativo tipo Unix, que implementa el estándar POSIX (“Portable Operating System Interface for Computer Environments, Unix based” es decir “Sistema operativo portable basado en Unix”), que utiliza primordialmente filosofía y metodologías libres (también conocido como GNU/Linux) y que está formado mediante la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU (GNU es un acrónimo recursivo que significa "GNU No es Unix") y de muchos otros proyectos/grupos de software (libre o no libre). El núcleo no es parte oficial del proyecto GNU (el cual posee su propio núcleo en desarrollo, llamado Hurd), pero es distribuido bajo los términos de la licencia GNU GPL (General Public License, Licencia Pública General).

La expresión *Linux* también es utilizada para referirse a las distribuciones GNU/Linux, colecciones de software que suelen contener grandes cantidades de paquetes además del núcleo. El software que suelen incluir consta de una enorme variedad de aplicaciones,

como: entornos gráficos, suites ofimáticas, servidores Web, servidores de correo, servidores ftp, etcétera. Coloquialmente se aplica el término *Linux* a éstas, aunque en estricto rigor sea incorrecto, dado que la distribución es la forma más simple y popular para obtener un sistema GNU/Linux.

La marca *Linux* (Número de serie: 1916230) pertenece a Linus Torvalds y se define como "*un sistema operativo para computadoras que facilita su uso y operación*".

Desde su lanzamiento, Linux ha incrementado su popularidad en el mercado de servidores. Su gran flexibilidad ha permitido que sea utilizado en un rango muy amplio de sistemas de cómputo y arquitecturas: computadoras personales, supercomputadoras, dispositivos portátiles, etc.

Los sistemas GNU/Linux funcionan sobre más de 20 plataformas diferentes de hardware, entre ellas las más comunes son las de los sistemas compatibles con PCs x86 y x86-64, computadoras Macintosh, PowerPC, Sparc y MIPS("millones de instrucciones por segundo", es una forma de medir la potencia de los procesadores.).

Asimismo, existen Grupos de Usuarios de GNU/Linux en casi todas las áreas del planeta.

En español no existe mayor conflicto con la pronunciación del término Linux, puesto que se lee como se escribe y se escribe como se lee, a no ser que se intente pronunciarlo como si estuviera intentando hacerlo en inglés con la i sonando como ai /láí.nuks/ o /lanəks/ (que se ha vuelto popular, porque suena natural en inglés).

Algunas personas anglosajones tienen este problema, por lo que la aclaración en páginas en inglés resulta necesaria, la pronunciación correcta (para cualquier idioma) es muy cercana como se pronuncia en español: /lí.nuks/ o /lnəks/ (Alfabeto Fonético Internacional).

Con la adopción por numerosas empresas fabricantes de PCs, muchas computadoras son vendidas con distribuciones GNU/Linux pre-instaladas, y Linux ha comenzado a tomar su lugar en el vasto mercado de las computadoras de escritorio.

Con entornos de escritorio, GNU/Linux ofrece una interfaz gráfica alternativa a la tradicional interfaz de línea de comandos de Unix. Existen en la actualidad numerosas aplicaciones gráficas, ya sean libres o no, que ofrecen funcionalidad que está permitiendo que GNU/Linux se adapte como herramienta de escritorio.

Algunas distribuciones permiten el arranque de Linux directamente desde un disco compacto (llamados LiveCDs) sin modificar en absoluto el disco duro de la computadora en la que se ejecuta Linux. Para este tipo de distribuciones, en general, los archivos de imagen están disponibles en Internet para su descarga.

Otras posibilidades incluyen iniciar el arranque desde una red (ideal para sistemas con requerimientos mínimos) o desde un disco flexible o disquete.

La creciente popularidad de GNU/Linux se debe a las ventajas que presenta ante otros tipos de software. Entre otras razones se debe a su estabilidad, al acceso a las fuentes (lo que permite personalizar el funcionamiento y auditar la seguridad y privacidad de los datos tratados), a la independencia de proveedor, a la seguridad, a la rapidez con que incorpora los nuevos adelantos tecnológicos (IPv6, que es la versión seis del Protocolo IP (Internet Protocol), microprocesadores de 64 bits), a la escalabilidad (se pueden crear *clusters* de cientos de computadoras), a la activa comunidad de desarrollo que hay a su alrededor, a su interoperabilidad y a la abundancia de documentación relativa a los procedimientos.

Hay varias empresas que comercializan soluciones basadas en GNU/Linux: por lo tanto es así que GNU/Linux es usado por muchas personas a nivel mundial IBM, Novell, Red Hat, Rxtart, así como miles de PYMES (pequeñas y medianas empresas) que ofrecen productos o servicios basados en esta tecnología.

Dentro del segmento de supercomputadoras, la más grande de Europa se llama *MareNostrum*. Desarrollado por IBM, está basado en un *cluster* GNU/Linux (Presentación de MareNostrum en IBM) y junto con esta muchas más supercomputadoras funcionando con Linux.

Linux tiene una amplia cuota en el mercado de servidores de Internet debido, entre otras cosas, a la gran cantidad de soluciones que tiene para este segmento.

Referencias utilizadas para este capítulo:

BURNETT, Steven; PAREDES, Beatriz; TACKETT, Jack, LINUX Cuarta Edición. 2000.

<http://es.wikipedia.org/wiki/Linux>

http://es.tldp.org/FAQ/FAQ_Linux/Html/FAQ_Linux-1.html

<http://www.cosaslibres.com/plinux.html>

http://es.wikipedia.org/wiki/C%C3%B3digo_abierto

http://es.wikipedia.org/wiki/Distribuci%C3%B3n_Linux

<http://www.distro.cl/porque-un-servidor-linux/>

2.2. Historia

LINUX hace su aparición a principios de la década de los noventa, era el año 1991 y por aquel entonces un estudiante de informática de la Universidad de Helsinki, llamado Linus Torvalds empezó como una afición y sin poderse imaginar a lo que llegaría este proyecto, a programar las primeras líneas de código de este sistema operativo llamado LINUX.

Este comienzo estuvo inspirado en MINIX (versión del sistema operativo UNIX desarrollada con fines educativos por el profesor Andrew Tannenbaum). Las primeras discusiones sobre Linux fueron en el grupo de noticias comp.os.minix, en estas discusiones se hablaba sobre todo del desarrollo de un pequeño sistema Unix para usuarios de Minix que querían más.

Linus nunca anunció la versión 0.01 de Linux (agosto 1991), esta versión no era ni siquiera ejecutable, solamente incluía los principios del núcleo del sistema, estaba escrita en lenguaje ensamblador y asumía que uno tenía acceso a un sistema Minix para su compilación.

El 5 de octubre de 1991, Linus anuncio la primera version "Oficial" de Linux, versión 0.02, con esta versión Linus pudo ejecutar Bash (GNU Bourne Again Shell) y gcc (El compilador GNU de C) pero no mucho mas funcionaba. En este estado de desarrollo ni se pensaba en los términos soporte, documentación, distribución.

Después de la version 0.03, Linus saltó en la numeración hasta la 0.10, más y más programadores a lo largo y ancho de Internet empezaron a trabajar en el proyecto y después de sucesivas revisiones, Linus incremento el número de version hasta la 0.95 (Marzo 1992), más de un año después (diciembre 1993) el núcleo del sistema estaba en la version 0.99 y la version 1.0 no llego hasta el 14 de marzo de 1994.

Desde entonces no se ha parado de desarrollar, la version actual del núcleo es la 2.2 y sigue avanzando día a día con la meta de perfeccionar y mejorar el sistema.

2.3. ¿Qué es Linux?

LINUX es un sistema operativo, compatible Unix. Dos características muy peculiares lo diferencian del resto de los sistemas que podemos encontrar en el mercado, la primera, es que es *libre*, esto significa que no tenemos que pagar ningún tipo de licencia a ninguna casa desarrolladora de software por el uso del mismo, la segunda, es que el sistema viene acompañado del *código fuente*. El sistema lo forman el Núcleo del sistema más un gran numero de programas /librerías que hacen posible su utilización.

Linux se distribuye bajo la GNU (General Public License), por lo tanto, el código fuente tiene que estar siempre accesible.

El sistema ha sido diseñado y programado por multitud de programadores alrededor del mundo. El núcleo del sistema sigue en continuo desarrollo bajo la coordinación de Linus Torvalds, la persona de la que partió la idea de este proyecto.

Día a día, más y más programas/aplicaciones están disponibles para este sistema, y la calidad de los mismos aumenta de versión a versión, la gran mayoría de los mismos vienen acompañados del código fuente y se distribuyen gratuitamente bajo los términos de licencia de la GNU Public License.

En los últimos tiempos, ciertas casas de software comercial han empezado a distribuir sus productos para Linux y la presencia del mismo en empresas aumenta rápidamente por la excelente relación calidad-precio que se consigue con Linux.

Las plataformas en las que en un principio se puede utilizar Linux son 386, 486, Pentium, Pentium Pro, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como Alpha, MIPS, PowerPC y SPARC (Scalable Processor Architecture, una arquitectura para el conjunto de instrucciones que es capaz de ejecutar una CPU).

La colección de utilidades para la programación de GNU es con diferencia la familia de compiladores más utilizada en GNU/Linux. Tiene capacidad para compilar C, C++, Java, Ada, entre otros muchos lenguajes. Además soporta diversas arquitecturas mediante la compilación cruzada, lo que hace que sea un entorno adecuado para desarrollos heterogéneos.

Hay varios IDEs (Integrated Development Environment, entorno integrado de desarrollo) disponibles para GNU/Linux incluyendo, Anjuta, KDevelop y Eclipse, además existen editores extensibles como pueda ser Emacs que hoy en día siguen siendo ampliamente utilizados. GNU/Linux también dispone de capacidades para lenguajes de guión (script), aparte de los clásicos lenguajes de programación de shell, la mayoría de las distribuciones tienen instalado Python, Perl, PHP y Ruby.

2.3.1. ¿Qué significa Código Abierto?

Código abierto (del inglés *open source*) es el término por el que se conoce el software distribuido y desarrollado libremente. Este término empezaron a utilizarlo en 1998 algunos usuarios de la comunidad del software libre, tratando de usarlo como reemplazo al ambiguo nombre original en inglés del software libre (*free software*).

Free en inglés puede significar diferentes cosas: gratuidad y libertad. Por ello, por un lado, permite pensar en "software por el que no hay que pagar" (software gratuito) y, por

otro, se adapta al significado que se pretendió originalmente (software que posee ciertas libertades).

El término para algunos no resultó apropiado como reemplazo para el ya tradicional *free software*, pues eliminaba la idea de *libertad* (incluso hay algunos que usan en inglés el término *libre software* para evitar la ambigüedad de *free*).

Desde el punto de vista de una "traducción estrictamente literal", el significado obvio de "código abierto" es que "se puede mirar el código fuente", por lo que puede ser interpretado como un término más débil y flexible que el del software libre. Basado en ello se argumenta que un programa de código abierto puede ser software libre, pero también puede ser semilibre o incluso completamente no libre. Sin embargo, por lo general, un programa de código abierto puede ser y de hecho es software libre, como igualmente un programa software libre es open source. Esto ocurre dado que ambos movimientos reconocen el mismo conjunto de licencias y tiene principios equivalentes.

Un término que pretende resolver posibles ambigüedades o confusiones que ambos términos generan es FOSS (Free Open Source Software).

En la actualidad *open source* es utilizado para definir un movimiento nuevo de software (la Open Source Initiative), diferente al movimiento del Software Libre, incompatible con este último desde el punto de vista filosófico, y completamente equivalente desde el punto de vista práctico, de hecho, ambos movimientos trabajan juntos en el desarrollo práctico de proyectos.

La filosofía del *Open Source* centra su atención en la premisa de que al compartir el código, el programa resultante tiende a ser de calidad superior al software propietario, es una visión meramente técnica.

2.4. Distribuciones

Una distribución es un modo de facilitar la instalación, la configuración y el mantenimiento de un sistema GNU/Linux. Al principio, las distribuciones se limitaban a recopilar software libre, empaquetarlo en disquetes o CD-ROM y redistribuirlo o venderlo.

Ahora las grandes distribuciones RedHat, SuSE, Caldera, Mandrake, Corel Linux, TurboLinux, entre otros, son potentes empresas que compiten entre sí por incluir el último software, a veces también software propietario, con instalaciones gráficas capaces de auto detectar el hardware y que instalan un sistema entero en unos cuantos minutos sin apenas preguntas.

Entre las distribuciones de GNU/Linux, destaca el proyecto Debian/GNU. Debian nace como una iniciativa no comercial de la FSF (Free Software Foundation), aunque luego se independiza de ésta y va más allá del propio sistema GNU/Linux. Es la única de las grandes distribuciones que no tiene intereses comerciales ni empresariales. Son sus propios usuarios, muy activos, quienes mantienen la distribución de modo comunitario, incluidas todas sus estructuras de decisión y funcionamiento. Su objetivo es recopilar, difundir y promover el uso del software libre. Reúne el mayor catálogo de software libre, todos ellos probados, mantenidos y documentados por algún desarrollador voluntario.

Existen numerosas distribuciones Linux. Cada una de ellas puede incluir cualquier número de software adicional (libre o no), como algunos que facilitan la instalación del sistema y una enorme variedad de aplicaciones, entre ellos, entornos gráficos, suites ofimáticas, servidores Web, servidores de correo, servidores FTP, etcétera.

La base de cada distribución incluye el núcleo Linux, con las bibliotecas y herramientas del proyecto GNU y de muchos otros proyectos/grupos de software, como BSD (Berkeley Software Distribution).

Entre algunas de las distribuciones no comerciales tenemos las siguientes:

- Archlinux (optimimizada para procesadores i686)
- Aurox (basada en Red Hat Linux)
- BestLinux
- Bluewall
- ByzantineOS
- Debian (x86/PPC)
- CentOS (basada en Red Hat Enterprise Linux)
- Fedora Core (x86/PPC) (basada en Red Hat Linux)
- Gentoo Linux (x86/PPC)
- Gnoppix (basada en Ubuntu, antes en Debian, de tipo CD autónomo)
- gNewSense (distribución oficial de la GNU)
- Jarro Negro (basada en Slackware)
- Knoppix (basada en Debian, de tipo CD autónomo)
- Kubuntu (x86/PPC/x86-64) (Ubuntu con KDE)
- Mandriva Linux (x86/PPC/x86-64) (antes Mandrake Linux)
- MkLinux (para todo tipo de modelos Apple Macintosh)
- Pardus
- ROCK Linux
- Slackware
- Slax (live basada en Slackware)
- Scientific Linux (basada en Red Hat Enterprise Linux)
- OpenSuSE
- Trinux (basada en Debian, de tipo CD autónomo)
- Trustix Secure Linux
- Ubuntu (x86/PPC/x86-64) (basada en Debian)
- VectorLinux (basada en Slackware)
- White Box (basada en Red Hat Enterprise Linux)
- Xubuntu (basada en Ubuntu pero con el escritorio Xfce)
- Admelix (una distribución chilena)
- ASLinux Desktop (distribución para escritorios de descarga gratuita basada en Debian y KDE mantenida por la empresa andaluza Activa Sistemas)

- EduLinux (una distribución educativa chilena)
- Espactra (Estación Práctica de Trabajo)
- gnUAMix (patrocinada por la Universidad Autónoma de Madrid, basada en Debian y de tipo CD autónomo)
- GuadaLinux (x86/PPC) (impulsada por la Junta de Andalucía (España) basada en Ubuntu, antes en Debian)
- Gobierno GDF/Linux (creada por la Delegación Tlalpan del Gobierno del Distrito Federal (México), basada en Fedora)
- Jarro Negro (creada por la Comunidad Linux UNAM Naucalpan CLUN, por estudiantes del Colegio de Ciencias y Humanidades plantel Naucalpan, basada en Slackware y Debian)
- JuegaLinux (x86/PPC) (Hermana de Guadalinux, pero con muchos juegos)
- Kwort (distribución creada en Rosario - Argentina y basada en Slackware)
- Linedux (distribución educativa creada en Lima - Perú y basada en Debian)
- LinEspa (creada por el foro LinuxenEspañol, basada en Debian)
- LinEx (creada por la Junta de Extremadura, España)
- Linuxin (basada en Debian GNU/Linux 3.0 (Woody) y realizada para novatos)
- Linuxinf (una distribución educativa/informática chilena)
- LliureX (creada por la Generalitat Valenciana (España) y orientada al sistema educativo. Soporta 2 idiomas: español y valenciano)
- LUC3M (distribución de la Universidad Carlos III de Madrid)
- Meduxa (creada por la Comunidad Autónoma de Canarias (España), basada en Ubuntu)
- Molinux (creada por la Comunidad Autónoma de Castilla-La Mancha (España), basada en Ubuntu)
- Musix GNU+Linux (100% Libre. Destinado a músicos, técnicos sonidistas y usuarios en general)
- Pequelin (distribución educativa para niños y jóvenes, basada en Knoppix)
- Trisquel (distribución creada por la Universidad de Vigo, en idiomas gallego y castellano)
- Tuquito (distribución creada en Tucumán - Argentina y basada en Debian)

- UTUTO XS (distribución 100% libre creada en Argentina por el Proyecto UTUTO con repositorios compilados para diversos procesadores Intel y AMD)
- ULAnix (distribución venezolana, basada en Debian, de tipo CD autónomo)
- WENDYX (distribución Chilena orientada a Desktop, basada en knoppix/Debian/biglinux).

Entre las distribuciones comerciales podemos mencionar:

- ASLinux Desktop (distribución para escritorios de descarga gratuita basada en Debian y KDE mantenida por la empresa andaluza Activa Sistemas)
- Caldera Linux
- Corel Linux (basada en Debian)
- Linspire (basada en Debian) (antes Lindows)
- Lycoris Desktop/LX
- Mandriva Fusion de Conectiva y Mandrake.
- Tumix GNU/linux
- Red Hat Linux
- RXART(distribución de linux basada en Debian y orientada al mercado latino comercializada en Perú, Uruguay, Chile y Argentina).
- Red Hat Enterprise Linux
- SUSE Linux (x86/PPC)
- Turbolinux
- Xandros (basada en Corel Linux e inspirada en Debian)
- Yellow Dog Linux (para PPC, basada en Fedora Core PPC)

2.5. Servidores en Linux

La escalabilidad y disponibilidad de recursos es una característica importante de cualquier servicio. El hardware que pudo ser potente unos años atrás hoy en día puede no manejar más los grandes volúmenes de datos a los cuales es sometido, es por eso que las soluciones de balanceo de carga entran en escena con Linux.

En el caso de un servidor de correo, la rapidez es fundamental, largas colas de mensajes pueden saturar hasta el más potente de los servidores. Por eso, los servidores de correo más grandes del mundo con balanceo de carga utilizan plataformas basadas en GNU/Linux, al ser un sistema operativo que ofrece mucha flexibilidad al momento de configurar los servidores con repartidores de carga que sobre él se montan, permitiendo que el tráfico no esté dirigido a un único servidor, si no que las respuestas a peticiones de clientes sea compartida por varios servidores.

El costo de instalación y mantenimiento de este tipo de servidores en Linux es mucho menor que un servidor montado sobre una plataforma de software propietaria como Windows de Microsoft, por ello, muchas personas piensan que los servidores de correo basados en GNU/Linux son la mejor opción, además Linux tiene software como el LVS para implementar varias aplicaciones con balanceo de carga además de un servidor de correo.

Por otro lado, Linux tiene disponible todos los servicios habituales para el funcionamiento de una red como son: bases de datos, servicios de Internet., servicio de ficheros e impresión, utilidades necesarias para mantener el nivel de seguridad requerido.

Pero además hay que reseñar que cada servicio funciona sin afectar al resto de los servicios, es decir puede modificar la dirección IP de su equipo, las rutas, añadir, parar o reiniciar un servicio concreto sin que el resto de los servicios se vean afectados. Sólo es necesario detener el equipo para realizar operaciones con el hardware, como añadir un disco duro, o utilizar un nuevo núcleo. No se tendrá, pues, la necesidad de ser uno mismo el atacante de nuestro propio sistema, a diferencia de lo que ocurre en otros sistemas operativos.

Por el costo de un sencillo PC, se puede tener un servidor de alto rendimiento y confiabilidad sin tener que pagar altísimos costos por licenciar software, el costo de licenciar es prácticamente nulo, el costo de mantenimiento del hardware es mínimo (el costo de mantener un PC). Además Linux corre en una variedad impresionante de diferentes plataformas de hardware, desde un modesto PC casero armado con un viejo

procesador DX4-100 Intel y 16 megas de memoria RAM, hasta un super moderno equipo basado en tecnología Alfa, una estación de trabajo de alto rendimiento, un Laptop personal o sobre un equipo servidor configurado con varios procesadores Intel P4, es decir, prácticamente en cualquier equipo que se seleccione para algún propósito.

Si es necesario por cuestiones de costos, se puede configurar la misma caja Linux para implantar simultáneamente varios servidores y hasta enrutar o implementar un necesario cortafuegos utilizando la herramienta netfilter, iptables.

Los servidores Linux son super flexibles, es decir no tenemos que someternos a una determinada configuración de servidor preseleccionada ni pagar una fortuna por un disco duro, más memoria, una tarjeta ethernet adicional o algún aditamento que se requiera para cumplir con algunas necesidades. Con Linux se puede seleccionar a gusto la configuración que desee y se puede inclusive, si las circunstancias lo ameritan, reutilizar un PC de su inventario para instalar el servidor que tanto necesita a un costo realmente irrisorio.

Los servidores Linux son super estables, Linux no se cuelga, funciona por días, semanas, meses, años y más, la estabilidad del sistema operativo Linux y los servidores implementados sobre este es un hecho notorio ampliamente conocido.

Los servidores de Google, Yahoo y Altavista corren sobre Linux, exactamente, esta gente no puede darse el lujo de equivocarse pues manejan miles de transacciones por día, y también confían en Linux como plataforma para sus servidores.

Los servidores Linux están basados en una tecnología probada y ampliamente difundida, debido a que tanto el “hardware” como el “software” es producido por una gran cantidad de empresas y organizaciones en todo el mundo, creando de esta forma un enorme mercado, los costos son bajísimos y el ciclo de desarrollo es tan corto como es posible.

2.6. Ventajas de Linux

Entre muchas de las ventajas que nos ofrece este robusto sistema operativo tenemos las siguientes:

- En Linux pueden correr varios procesos a la vez de forma ininterrumpida como un servidor de red al tiempo que un procesador de textos, una animación, copia de archivos o revisar el correo electrónico.
- Seguridad porque es un sistema operacional diseñado con la idea de cliente - servidor con permisos de acceso y ejecución a cada usuario. Esto quiere decir que varios usuarios pueden utilizar una misma máquina al tiempo sin interferir en cada proceso.
- Linux es software libre, casi gratuito. Linux es popular entre programadores y desarrolladores e implica un espíritu de colaboración.
- Linux integra una implementación completa de los diferentes protocolos y estándares de red, con los que se puede conectar fácilmente a Internet y acceder a todo tipo de información disponible.
- Su filosofía y sus programas están dictados por el movimiento “Open Source” que ha venido creciendo en los últimos años y ha adquirido la suficiente fortaleza para hacer frente a los gigantes de la industria del software.
- Linux incorpora una gama de sistemas de interfaz gráfica (ventanas) de igual o mejor calidad que otras ofrecidas en muchos paquetes comerciales.
- Posee el apoyo de miles de programadores a nivel mundial.
- El paquete incluye el código fuente, lo que permite modificarlo de acuerdo a las necesidades del usuario.
- Utiliza varios formatos de archivo que son compatibles con casi todos los sistemas operacionales utilizados en la actualidad.
- Invulnerable a los virus (por lo tanto, no necesita antivirus).
- Invulnerable a los spywares (programa que se instala en tu PC y que roba tus datos para enviarlos a un tercero, por lo tanto no necesita antispyware).
- Su sistema de archivos no se fragmenta (Linux es muy ordenado con el uso del disco duro, por lo tanto, no es necesario defragmentarlo como en Windows).

- Se puede cancelar un programa, pero no se cancela Linux (Linux es muy eficiente con el uso de los recursos de la máquina como la memoria, procesador, disco duro, etc., por lo tanto, si un programa se cancela o se vuelve inestable, no afecta al funcionamiento del resto del sistema).
- Multitarea real (puede ejecutar múltiples procesos al mismo tiempo consumiendo menos memoria y procesador).
- Multiusuario real (muchos usuarios pueden usar el mismo PC al mismo tiempo).
- No requiere una máquina tan poderosa como Windows para funcionar (de hecho, los requerimientos mínimos son: Procesador 486 y 8Mb de RAM).
- Puedes convertir tu PC en una Estación de Trabajo o Multimedia (PC de escritorio), Servidor Web, Servidor de Correos, Servidor de Bases de Datos, Servidor de impresión, Servidor de archivos, etc.
- Mucha documentación de todo tipo.

2.7. Desventajas de Linux

Como desventajas podemos mencionar:

- Algunos Linux no cuentan con una empresa que lo respalde, por lo que no existe un verdadero soporte como el de otros sistemas operativos.
- La pendiente de aprendizaje es lenta.
- No es tan fácil de usar como otros sistemas operativos, aunque actualmente algunas distribuciones están mejorando su facilidad de uso, gracias al entorno de ventanas, sus escritorios y las aplicaciones diseñadas específicamente para él, cada día resulta más sencillo su integración y uso.
- Documentación y terminología muy técnica.
- Para usuarios corrientes, todavía no es un sistema de escritorio.
- Funciona únicamente con proveedores de hardware que accedieron a la licencia GPL y en algunas instancias no es compatible con variedad de modelos y marcas.
- Requiere consulta, lectura e investigación en lista, foros o en bibliografía dedicada al tema.

- La configuración de dispositivos de entrada y salida no es trivial.
- Muy sensible al hardware.
- Muchas distribuciones e idiomas.
- Hay que leer y entender código

2.8. Conclusiones

Luego de haber investigado algunos temas importantes sobre Linux hemos comprendido de manera general lo que es este sistema operativo como plataforma de implementación para el balanceo de carga, además hemos sacado las siguientes conclusiones:

- Linux tiene todo el software necesario para llevar a cabo una implementación de servidor de correo con balanceo de carga.
- Los servidores que se implementan bajo Linux son muy robustos, y de muy bajo costo tanto en hardware como en software.
- El software libre representa un nuevo modelo de desarrollo de software, un verdadero paradigma.
- La Licencia Pública General (GPL) de la Free Software Foundation, desempeña un papel clave en la promoción y desarrollo del software libre, ya que protege al software que se distribuye bajo esta licencia, contra patentes y restricciones desfavorables para los usuarios, lo que constituye una garantía para ellos.
- El sistema operativo GNU/Linux representa un terreno inexplorado que ha conseguido captar la atención de muchas empresas, lo que se está traduciendo en rentabilidad para ellas y en beneficio para los usuarios.
- Hace unos años GNU/Linux era un sistema sólo para universitarios e investigadores, ahora ya está ocupando un lugar a nivel empresarial.

CAPITULO 3

SERVIDORES DNS

3.1. Introducción

En este apartado describiremos en si la arquitectura de un servidor DNS y su modo de operación planteándonos como objetivo el comprender su funcionamiento. No abordaremos temas para la configuración de este tipo de servidor ya que la implementación de un DNS se tratará en el capítulo 5.

El Domain Name System (DNS) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. La asignación de nombres a direcciones IP es ciertamente la función más conocida de los protocolos DNS. Por ejemplo, si la dirección IP del sitio FTP de prox.ve es 200.64.128.4, la mayoría de la gente llega a este equipo especificando ftp.prox.ve y no la dirección IP, además de ser más fácil de recordar, el nombre es más fiable. La dirección numérica podría cambiar por muchas razones, sin que tenga que cambiar el nombre.

Inicialmente, el DNS nació de la necesidad de recordar fácilmente los nombres de todos los servidores conectados a Internet. En un inicio, SRI (ahora SRI International, instituto de investigación de Stanford) alojaba un archivo llamado *Hosts* que contenía todos los nombres de dominio conocidos (técnicamente, este archivo aún existe, la mayoría de los sistemas operativos actuales todavía pueden ser configurados para revisar su archivo hosts).

El crecimiento explosivo de la red causó que el sistema de nombres centralizado en el archivo Hosts no resultara práctico y en 1983, Paul Mockapetris publicó los RFCs (Request For Comments, conjunto de archivos de carácter técnico donde se describen los estándares o recomendaciones de cualquier cosa, entre otros los de la propia Internet) 882 y 883 definiendo lo que hoy en día ha evolucionado al DNS moderno. (Estos RFCs han quedado obsoletos por la publicación en 1987 de los RFCs 1034 y 1035).

La raíz de la base de datos del DNS está representada por el nodo "." y cada uno de los nodos que descienden de ella reciben el nombre de dominios. En el sistema DNS cada dominio se hace cargo de la base de datos que depende de él.

En cada dominio puede haber a su vez servidores y otros dominios, cada nombre de dominio se construye escribiendo los sucesivos nombres de dominio a los que pertenece el dominio hasta llegar al dominio raíz. Cada nombre se separa del siguiente mediante un punto y se escriben colocando a la izquierda los nodos inferiores. Por ejemplo el departamento imasd de la compañía acme, que operase en España recibiría el nombre: imasd.acme.es

El punto raíz no se pone, a un nombre de dominio que incluye todos los nodos hasta el raíz se le denomina nombre de dominio completamente cualificado (FQDN Full Cullified Domain Name, nombre de dominio completamente cualificado). En Internet por debajo del nodo raíz los primeros nodos corresponden normalmente a países u organizaciones internacionales, cada país tiene su propio dominio, y además existen otros para otro tipo de organizaciones.

En el caso de Estados Unidos, la mayoría de los dominios pertenecen por razones históricas a los dominios edu, com, mil y gov. Otros dominios que dependen del raíz son:

- int para organizaciones internacionales
- org para organizaciones no gubernamentales
- net para otras redes que se han unido a Internet
- arpa para la transición de Arpanet a Internet

y los diferentes países, identificados por el código de dos letras ISO3166, salvo Gran Bretaña, que usa uk. Dentro de cada dominio hay una organización que decide la estructura de dominios, por ejemplo en España es Es-NIC (Network Information Center, centro de información sobre la red), dependiente de RedIRIS (red de investigación Española).

El DNS se utiliza para distintos propósitos. Los más comunes son:

- Resolución de nombres: Dado el nombre completo de un *host* (por ejemplo *blog.smaldone.com.ar*), obtener su *dirección IP* (en este caso, *208.97.175.41*).
- Resolución inversa de direcciones: Es el mecanismo inverso al anterior. Consiste en, dada una *dirección IP*, obtener el nombre asociado a la misma.
- Resolución de servidores de correo: Dado un *nombre de dominio* (por ejemplo *gmail.com*) obtener el servidor a través del cual debe realizarse la entrega del correo electrónico (en este caso, *gmail-smtp-in.l.google.com*).

Por tratarse de un sistema muy flexible, es utilizado también para muchas otras funciones, tales como la obtención de claves públicas de cifrado asimétrico y la validación de envío de e-mails (a través de mecanismos como SPF “Sender Policy Framework”).

En redes de tamaño medio (quizás más de 5 equipos) es conveniente la utilización de DNS, esto nada tiene que ver con el DNS de *Internet* (aunque el servidor local puede estar vinculado a este sistema).

Básicamente, es conveniente montar un servidor local de DNS por los siguientes motivos:

- Agilizar el acceso a Internet, al tener un servidor de nombres en nuestra propia red local (que acceda al DNS de nuestro proveedor o directamente a los root servers) se agiliza el mecanismo de resolución de nombres, manteniendo en *caché* los nombres recientemente usados en la red y disminuyendo el tráfico hacia/desde Internet.
- Simplificar la administración de la red local, al contar con un DNS propio (ya sea uno o varios servidores de nombres) es posible definir zonas locales (no válidas ni accesibles desde *Internet*) para asignar nombres a cada uno de los hosts de la LAN. De esta forma es posible, por ejemplo, referirnos a la impresora de red como “*hplaser.mired.local*” en vez de “192.168.0.2” y a nuestro servidor de correo interno como “*smtp.mired.local*” en vez de “192.168.0.3”. (Pensemos, por

ejemplo, qué ocurriría con las configuraciones de las aplicaciones si un día decidimos cambiar el esquema de direcciones IP de nuestra red.)

El principal problema que presenta el DNS es que, al estar basado en UDP (protocolo de transporte que no garantiza la recepción de la información enviada), tanto las consultas como las respuestas pueden perderse (por ejemplo, a causa de congestión en algún enlace de la red). Es común apreciar cómo, en el caso de servidores y redes no muy bien configuradas, la resolución de nombres se resiente sensiblemente ante cualquier anomalía (saturación de tráfico o del servidor de nombres local).

Además DNS tiene como inconveniente la lentitud de la propagación de las modificaciones en el sistema, producto de la propia arquitectura del mismo.

Referencias utilizadas para este capítulo:

http://es.wikipedia.org/wiki/Domain_Name_System

<http://blog.smaldone.com.ar/2006/12/05/como-funciona-el-dns/>

<http://web.uservers.net/soporte/docview.php?articulo=58>

3.2. ¿Cómo trabaja DNS?

Antes de entrar al funcionamiento de DNS es necesario introducir algunos términos básicos para evitar confusiones y ambigüedades.

- Host Name: El nombre de un *host* es una sola "*palabra*" (formada por letras, números y guiones). Ejemplos de nombres de *host* son "www", "blog" y "obelix".
- Fully Qualified Host Name (FQHN): Es el "*nombre completo*" de un *host*. Está formado por el *hostname*, seguido de un punto y su correspondiente *nombre de dominio*. Por ejemplo, "blog.smaldone.com.ar"
- Domain Name: El *nombre de dominio* es una sucesión de nombres concatenados por puntos. Algunos ejemplos son "smaldone.com.ar", "com.ar" y "ar".

- Top Level Domains (TLD): Los *dominios de nivel superior* son aquellos que no pertenecen a otro dominio. Ejemplos de este tipo son "com", "org", "ar" y "es".

3.2.1. Componentes

Para la operación práctica del sistema DNS se utilizan tres componentes principales:

- Los Clientes DNS (resolvers), un programa cliente DNS que se ejecuta en la computadora del usuario y que genera peticiones DNS de resolución de nombres a un servidor DNS (Por ejemplo: ¿Qué dirección IP corresponde a nombre.dominio?).
- Los Servidores DNS (name servers), que contestan las peticiones de los clientes, los servidores recursivos tienen la capacidad de reenviar la petición a otro servidor si no disponen de la dirección solicitada.
- Y las Zonas de Autoridad, porciones del espacio de nombres de dominio que almacenan los datos. Cada zona de autoridad abarca al menos un dominio y posiblemente sus subdominios, si estos últimos no son delegados a otras zonas de autoridad.

3.2.2. Las Partes de un Nombre de Dominio

Un nombre de dominio usualmente consiste en dos o más partes (técnicamente etiquetas), separadas por puntos cuando se las escribe en forma de texto. Por ejemplo, www.mahomedalid.org o es.wikipedia.org

- A la etiqueta ubicada más a la derecha se le llama “dominio de nivel superior” (en inglés Top Level Domain). Como org en www.mahomedalid.org o es.wikipedia.org
- Cada etiqueta a la izquierda especifica una subdivisión o subdominio, nótese que "subdominio" expresa dependencia relativa, no dependencia absoluta. En teoría, esta subdivisión puede tener hasta 127 niveles, y cada etiqueta contener hasta de 63 caracteres, pero restringido a que la longitud total del nombre del dominio no

exceda los 255 caracteres, aunque en la práctica los dominios son casi siempre mucho más cortos.

- Finalmente, la parte más a la izquierda del dominio suele expresar el nombre de la máquina (en inglés hostname). El resto del nombre de dominio simplemente especifica la manera de crear una ruta lógica a la información requerida. Por ejemplo, el dominio es.wikipedia.org tendría el nombre de la máquina “es”, aunque en este caso no se refiere a una máquina física en particular.

El DNS consiste en un conjunto jerárquico de servidores DNS, cada dominio o subdominio tiene una o más zonas de autoridad que publican la información acerca del dominio y los nombres de servicios de cualquier dominio incluido. La jerarquía de las zonas de autoridad coincide con la jerarquía de los dominios. Al inicio de esa jerarquía se encuentra los servidores raíz: los servidores que responden cuando se busca resolver un dominio de primer y segundo nivel.

3.2.3. Tipos de Servidores DNS

Tenemos los siguientes:

- Bind
- PowerDNS
- MaraDNS
- djbdns
- pdnsd
- MyDNS

El software más utilizado en los servidores de nombres de Internet es Bind (Berkeley Internet Name Domain), creado originalmente en la Universidad de California, y actualmente propiedad del Internet Systems Consortium.

Este programa, distribuido bajo una licencia libre, es utilizado en prácticamente todos los sistemas Unix del mundo. Esto ha sido considerado un problema de seguridad, al punto que se ha propuesto la migración de algunos root servers a otro sistema, ya que la