TÍTULO
Clustering de Alta Disponibilidad bajo GNU/Linux

AUTOR
Vicente José Aguilar Roselló

TUTOR
D. Manuel Marco Such

DEPARTAMENTO
Departamento de Lenguajes y Sistemas Informáticos

CURSO
2000 – 2001

ÍNDICE

1. INTRODUCCIÓN

Con el actual ritmo de crecimiento del comercio y el movimiento de datos de todo tipo en *Internet* (más de un 100% anual) y la incuestionable importancia de la informática en las empresas actuales de cualquier tamaño, es cada día más importante que los *sistemas informáticos* de éstas puedan funcionar de forma ininterrumpida y sin errores las 24h del día, 7 días a la semana y 365 días al año, ya sea para dar soporte interno (contabilidad, control de personal, desarrollo...) como para ofrecer servicios a través de Internet (comercio electrónico, correo, portales, etc). A esta necesidad de un servicio ininterrumpido y fiable se le conoce como *alta disponibilidad*.

Dos estudios independientes realizados en 1995 por Oracle Corp. y Datamation revelaron que una empresa media pierde entre 80,000 y 350,000 dólares (entre 15 y 70 millones de pesetas) por hora de interrupción no planeada de sus servicios informáticos. Otro ejemplo de la necesidad de la alta disponibilidad es que tras el atentado en el World Trade Center en 1993, 145 de las 350 empresas que allí se hospedaban (algo más del 40%) tuvieron que cerrar sus puertas tras este incidente por no disponer de una infraestructura informática redundante.

La principal técnica para obtener estos sistemas tolerantes a fallos es la redundancia, estrategia utilizada en la industria aeronáutica prácticamente desde sus principios, que consiste en replicar las zonas críticas del sistema, teniendo una unidad activa y varias copias inactivas que, tras el fallo de la principal, sean capaces de retomar su labor en el punto que aquella falló, en el menor tiempo posible y de forma transparente para el usuario.

Existen gran cantidad de servidores altamente redundantes en el mercado fabricados por SUN, IBM y demás empresas del ramo. Son grandes máquinas multiprocesador , con varias controladoras de disco, configuraciones RAID, fuentes de alimentación redundantes, y un largo etcétera de circuitos y controladoras duplicadas para que, en caso de fallo, haya alguna de respaldo. El precio de este tipo de equipos rara vez baja de varias decenas de millones de pesetas. Además, cuando una máquina de este tipo queda obsoleta, no nos queda otro remedio que comprar otra mayor y deshacernos de la antigua.

El presente estudio se centrará en la técnica de obtener una alta disponibilidad por medio de la redundancia, instalando *varios* servidores *completos* en lugar de uno sólo, que sean capaces de trabajar *en paralelo* y de asumir las caídas de algunos de sus compañeros, y podremos añadir y quitar servidores al grupo (*cluster*) según las necesidades. A esta técnica se la denomina *clustering*. Por otra parte, también se abordarán todas las técnicas necesarias para asegurar la estabilidad de cada uno de los servidores del cluster, técnicas que en muchos casos también se basarán en la redundancia de dispositivos. En todos caso los equipos serán PCs normales de los que podemos encontrar en cualquier tienda de informática personal, con procesadores Intel Pentium o AMD, que en ningún caso valdrá cada uno más de doscientas mil pesetas.

Este trabajo está estructurado según el orden que seguiremos a la hora de ir configurando cada uno de los equipos que formarán parte de nuestro cluster: tras una

introducción inicial a las diversas técnicas de clustering, su problemática y sus soluciones, comenzaremos viendo los métodos para asegurar que la información almacenada en los discos de nuestros servidores sea segura, cómo conseguir que éstos compartan información, cómo conseguir que un equipo tome el control de los servicios de otro, cómo organizar y administrar el cluster y cómo dividir el cluster geográficamente en un “cluster de clusters”.

2. GESTIÓN DEL ALMACENAMIENTO

Una de las primeras cosas en las que tendremos que pensar a la hora de implantar un sistema de alta disponibilidad será en cómo asegurar la integridad y fiabilidad de los datos almacenados en los discos de nuestros servidores, que deberán estar disponibles de forma continuada durante largos (indefinidos) periodos de tiempo. Un fallo en un dispositivo de almacenamiento podría llevarnos a dar datos erróneos si el fallo se produce en una zona de datos ,con efectos imprevisibles para nuestra empresa; o a un mal funcionamiento del programa si el fallo se localiza en una zona que almacene ejecutables, con efectos aún más imprevisibles, desde la entrega de datos erróneos, hasta el mal funcionamiento del servidor pasando desde el servicio de datos erróneos hasta la corrupción irreversible de los mismos.

En este capítulo se analizan las distintas técnicas disponibles para asegurar la consistencia de los datos albergados en los dispositivos de almacenamiento de nuestros servidores.

2.1. RAID

RAID (Redundant Array of Inexpensive Disks), como su propio nombre indica, consiste en crear un *array* (cadena) de varios discos simples (“*inexpensive*”, baratos), y tratarlos como un todo a la hora de acceder a ellos. El standard RAID consta de varios niveles, y en cada uno de ellos el acceso a los discos y su contenido se organiza de una forma u otra para conseguir bien mayor capacidad que la de un único disco físico, bien mayor rapidez en el acceso a los datos, bien tolerancia a fallos, o bien alguna combinación de las anteriores.

2.2. LVM

Se trata de una nueva forma de asignar espacio de disco a los sistemas de ficheros: en lugar de utilizar particiones de tamaño fijo, se utilizan particiones “virtuales”, que podrán crecer o disminuir según nuestras necesidades administrativas. Además, el espacio asignado a una partición no tiene por qué pertenecer todo al mismo disco, con lo que se rompe la barrera de únicamente poder tener particiones como mucho del tamaño del mayor de los discos que tengamos instalados (si bien esto también era posible mediante RAID).

Una de las características importantes de LVM es que funciona sobre cualquier

disposivo de bloques, incluso sobre un dispositivo que no sea más que un RAID software como los del apartado anterior. De esta forma, podemos obtener seguridad y flexibilidad montando LVM sobre RAID.

2.3. ext2

El sistema de archivos de Linux por excelencia. En este apartado se analiza a fondo su estructura, funcionamiento y forma de uso.

2.4. ReiserFS

Se trata de un sistema de archivos nuevo para Linux, que viene a reemplazar a ext2. Internamente se estructura en árboles balanceados, un tipo de dato muy utilizado para algoritmos de búsqueda rápidos, con lo que se consigue llevar esta rapidez al mundo de los sistemas de ficheros. Por otra parte, ReiserFS es un sistema de ficheros transaccional, que va llevando una “bitácora” de todas las acciones que realiza para, tras una caída del sistema, poder reparar el sistema de ficheros y conseguir devolverlo a un estado consistente en un mínimo de tiempo.

2.5. xfs y jfs

Son dos sistemas de ficheros nuevos para Linux, que nos llegan de la mano de IBM y SGI. Ambos son sistemas de ficheros transaccionales, como ReiserFS, pero ninguno ha llegado aún al nivel de estabilidad de aquel, y ninguno ha sido integrado todavía en la distribución oficial del kernel de Linux.

3. DISTRIBUCIÓN DE LOS DATOS

Una vez que ya conocemos las diversas técnicas para salvaguardar los datos de nuestros discos duros y posibilitar el cambio de discos en caliente, y los distintos sistemas con los que organizar los sistemas de archivos, se nos presenta otro problema: como ya se avanzó en los primeros capítulos, vamos a conseguir la alta disponibilidad a través de la replicación de servidores, capaces de trabajar en paralelo como uno sólo e incluso sustituirse unos a otros en sus funciones. Esto implica que los datos que tengan que servir o procesar deben estar disponibles para todos y cada uno de nuestros servidores, pero, ¿cómo conseguirlo? Nuestra intención es crear varios servidores, réplicas exactas unos de otros, que sirvan todos el mismo contenido, tendremos que encontrar alguna forma de realizar estas réplicas automáticamente, de forma que para el usuario (en este caso, los desarrolladores o encargados de contenidos) el cluster se comporte como un único ordenador, en el que ellos copian (o trabajan) en un único lugar los ficheros, y el software de control del cluster internamente se encargue de hacer llegar una copia a cada uno de los servidores que

lo componen.

A este respecto tenemos dos estrategias: la replicación física de archivos, en la que cada servidor tendrá una copia de todos los datos en su disco duro; y la distribución de los datos mediante sistemas de archivos distribuidos, en los que tendremos un servidor de ficheros y el resto de equipos del cluster accederán a sus contenidos por la red. Cada estrategia tendrá sus ventajas y desventajas, que en este capítulo estudiaremos.

3.1. rsync

Se trata de un novedoso programa para la replicación de archivos entre servidores. En lugar de instalar un servidor FTP en uno de ellos y que los demás se conecten y tengan que bajar todo el contenido del servidor cada vez, mediante rsync es posible realizar “sincronizaciones”, de forma que tan sólo se transmiten por la red aquellas partes de los ficheros que hayan sido modificados: el resto, no es necesario transmitirlo, con lo que se ahorra tiempo y ancho de banda.

3.2. NFS

El sistema de ficheros compartidos por excelencia del mundo UNIX. En este apartado analizamos cómo funciona y cómo se instalaría un servidor NFS.

3.3. Samba

Samba es un sistema de ficheros compartido similar a NFS, con la particularidad de que “habla el idioma” de Windows. En efecto, mediante Samba podremos acceder desde Linux a los “recursos compartidos” de Windows, y viceversa: instalar en un Linux un servidor de ficheros al que se pueda acceder desde el “entorno de red” de Windows.

3.4. CODA

Se trata de un servidor de ficheros para Linux con un enfoque distinto al de NFS. Su principal característica es que los clientes van realizando una caché local de los ficheros remotos a los que acceden, y posteriormente en cada acceso se utiliza la caché (si el fichero remoto no ha sido modificado). De esta forma se consiguen mejores tiempos de acceso a los ficheros, y además, gracias a la caché y a las opciones avanzadas para su control, se puede seguir trabajando tras una desconexión de la red, voluntaria (estamos usando un ordenador portátil) o involuntaria (el servidor de ficheros ha caído).

3.5. GFS

Un nuevo concepto en los sitemas de ficheros compartidos, ya que más bien es un sistema de discos compartidos: Utilizando un bus SCSI o una red Fibre Channel, varios equipos acceden de forma simultánea y en igualdad de condiciones a todo el grupo de discos del que disponemos. De esta forma se elimina el modelo cliente/servidor en la distribución de los datos, eliminando el servidor que se convertía en un cuello de botella y en un posible punto de fallo que podría llegar a inutilizar todo el cluster.

4. MONITORIZACIÓN

Otro aspecto muy importante en el clustering de alta disponibilidad es la monitorización de los servicios: si alguno de nuestros servidores “cae”, tendremos que advertirlo de alguna forma y desencadenar las acciones pertinentes (eliminarlo de la lista de servidores activos y hacer que algún otro servidor tome el lugar de este). En este capítulo se analizan varias opciones:

4.1. daemontools

Un grupo de programas similar en concepto al demonio inetd pero más potentes, daemontools se encarga de lanzar un servidor cuando se detecte un acceso a los puertos de su protocolo asociado, y también se encarga de monitorizar el estado del servidor a nivel de proceso: si este muere de forma inesperada, daemontools se encarga de volverlo a lanzar.

4.2. mon

Se trata de un programa para monitorizar servicios. Nos ofrece la infraestructura necesaria para lanzar una serie de monitores en unos momentos programados, de forma similar a como se podría hacer mediante cron. Si alguno de los monitores falla, se pueden lanzar una serie de alertas para avisarnos del problema.

La grandeza de mon radica en que es fácilmente extensible, ya que los monitores y alertas son programas que podemos escribir nosotros en cualquier lenguaje, desde shell-script hasta C, con lo que podremos cubrir cualquier necesidad.

4.3. heartbeat

Si mon monitoriza servicios, heartbeat servidores. Es un programa mediante el cual dos equipos se intercambian “pulsos” por diversos medios: una red ethernet, una red TCP/IP, un cable serie o paralelo... Ante la caída de uno de los dos equipos, el otro es capaz de ocupar su lugar suplantando su dirección IP mediante el método conocido como ARP IP spoofing.

4.4. iANS

iANS en un programa de Intel para poder agrupar tantas tarjetas suyas como tengamos en nuestro equipo y, bien hacer que funcionen todas en paralelo, obteniendo así una mayor velocidad; o bien hacer que cuando una falle, otra tome su lugar de forma automática y transparente, de forma que el equipo no quede inutilizado. A esta práctica se le conoce como “failover”.

5. CLUSTERING DE ALTA DISPONIBILIDAD

En este capítulo ya pasamos a analizar las opciones para montar un cluster. En concreto, analizamos dos programas:

5.1. Linux Virtual Server

Se trata de la solución de clustering de servidores por excelencia del mundo GNU/Linux. Proporciona el software necesario para construir un cluster con balanceo de carga y alta disponibilidad, donde se evitará cualquier punto de fallo duplicándolo y automatizando el proceso de “failover” de un equipo a otro. Dispone de tres métodos distintos de enrutamiento de la carga, del balanceador a los servidores reales, y de varios algoritmos de selección de servidor.

También analizamos en este punto algunos programas que nos ayudarán a instalar, configurar y administrar el cluster.

5.2. Super Sparrow

Se trata de un ingenioso programa que automatiza el proceso de selección de un mirror remoto por parte del cliente. Es decir, en lugar de ofrecer al cliente una lista de todos los mirrors dispersos geográficamente de que disponga una página, con Super Sparrow se puede conseguir que, se conecte al mirror que se conecte el cliente, se le redirija de forma automática y completamente transparente al que le resulte más cercano.

6. PROGRAMAS PARA LA INSTALACIÓN Y LA ADMINISTRACIÓN

Una vez que ya hemos revisado todas las posibilidades para implantar un cluster de Alta Disponibilidad, hemos analizado nuestras necesidades, y hemos decidido qué software y qué infraestructura instalar, nos queda el paso más tedioso y susceptible a errores humanos desde el punto de vista del administrador de sistemas: la instalación del sistema operativo y el resto del software en *cada uno* de los servidores del cluster, y su consiguiente configuración. En este apartado se analizan las distintas herramientas disponibles para automatizar en la medida de lo posible esta tarea.

6.1. Linux Utility for cluster Installation

Un programa desarrollado por IBM para la instalación remota de equipos. Si bien las opciones y el acabado son buenos (incluso dispone de un instalador gráfico), presenta el problema de estar íntimamente ligado a la distribución RedHat, siendo imposible utilizar LUI para instalar cualquier otra distribución y mantener programas instalados por nosotros mismos (que no vinieran en formato .rpm).

6.2. FAI

Similar al anterior, pero para la distribución Debian. Presenta los mismos problemas que LUI por estar ligado a una distribución, si bien el aspecto del programa es distinto (la instalación de LUI es gráfica, la de FAI en línea de comandos más orientada a ser automatizada).

6.3. VA System Imager

Un programa desarrollado por VA Linux Systems con el mismo objetivo que los anteriores. En este caso, se consigue que el programa sea independiente de toda distribución de Linux, siendo posible instalar de forma remota cualquier sistema, incluso si ha sido instalado completamente de forma manual.

6.4. webmin

Es un programa para la administración remota de equipos, con una interfaz basada en la web. Con tan sólo un navegador HTTP, se pueden configurar servicios en los equipos remotos.

7. PRUEBAS DE SOFTWARE

Antes de lanzarse a instalar cualquier programa, se deben realizar pruebas para asegurarnos de su buen funcionamiento. En nuestro caso, se decidió probar de forma aislada algunos de los programas y tecnologías expuestos, para “hacernos con ellos” y poder realizar pruebas de rendimiento de cada una de las partes independientemente.

Se realizaron pruebas de los siguientes programas:

7.1. RAID, LVM, ext2 y ReiserFS

Se han realizado pruebas de rendimiento sobre los sistemas de ficheros ext2 y ReiserFS, directamente sobre el disco, y con RAID y/o LVM por debajo. Los resultados se presentan en forma de gráficas comparativas entre las distintas opciones. Los resultados fueron bastante satisfactorios.

7.2. VA System Imager

Se utilizó el programa VA System Imager para replicar la instalación de un equipo con Debian GNU/Linux en otro. Durante el proceso se encontró algún pequeño problema pero que se pudo solucionar. Los resultados, en general, fueron bastante satisfactorios.

7.3. CODA

En la siguiente tanda de pruebas, se comprobó el funcionamiento del sistema de ficheros distribuido CODA. Era un sistema que prometía mucho, pero que tras las pruebas realizadas se quedó tan sólo en eso: promesas. Si bien muchas de sus características avanzadas funcionaron a la perfección (caching local de ficheros), otras no tanto, mientras que su rendimiento no fue tan superior a NFS como se esperaba y además provocó errores durante las pruebas.

7.4. mon

Utilizando el programa para monitorización de servicios mon, se monitorizó un servidor HTTP Apache, y se preparó un par de alarmas para avisar al administrador

por correo electrónico del fallo y para tratar de volver a arrancar el servicio caído. Los resultados fueron muy satisfactorios.

7.5. iANS

En un equipo con tres tarjetas de red, se configuraron en un grupo en modo ATF para que hicieran failover entre ellas cuando alguna tuviera problemas. Este tipo de configuración a la perfección.

8. CONCLUSIONES

Por último, se resume la impresión general del autor sobre el estado de las tecnologías de clustering de servidores bajo GNU/Linux, se da una idea de configuración para un cluster sencillo, y se repasan los aspectos importantes para alcanzar la alta disponibilidad de los servicios ofrecidos por el cluster que, por un motivo u otro, se han dejado de lado a lo largo del trabajo.