Alta disponibilidad para Linux

Juan Pedro Paredes

juampe@retemail.es

Este documento relata los principios básicos de la alta disponibilidad, además de dar un enfoque a la problemática que ha llevado el desarrollo de estos sistemas.

Luego se enumeraran las características que hacen de Linux un sistema operativo robusto para su uso en estos sistemas. Además se comentaran diversas soluciones ya creadas.

1. Introducción

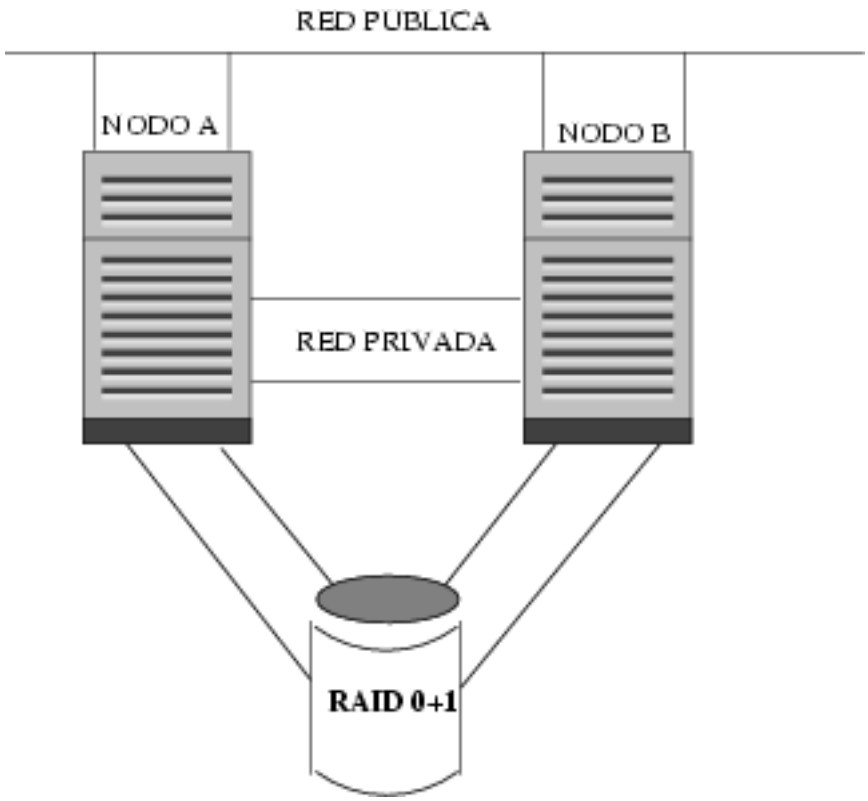
Actualmente Linux es conocido, como un sistema operativo estable; la problemática se genera cuando el hardware, no es tan fiable como se desearía. En la mayoría de los casos, cuando un sistema falla normalmente es debido a un fallo de hardware o a un fallo humano (debido a un error en la administración del sistema).

En los casos en que un fallo hardware provoca graves consecuencias, debido a la naturaleza del servicio (aplicaciones críticas), se implementan sistemas tolerantes a fallos (fault tolerant ó FT); en los cuales, el servicio esta siempre activo. El problema de estos sistemas, es que son extremadamente caros y normalmente no hay presupuesto. Además suelen ser soluciones cerradas, totalmente dependientes de la empresa contratada. Se suele poner un servidor tolerante a fallos, varias interfaces de red, con tomas de alimentación redundantes y climatización especial.

Los sistemas de alta disponibilidad (high availability ó HA), intentan obtener prestaciones cercanas al la tolerancia a fallos, pero a un precio muchísimo más interesante. Esta es una opción, que la ha hecho crecer en importancia dentro del

mundo empresarial. La alta dsiponibilidad está basada en la replicación de elementos, mucho más baratos que un sólo elemento tolerante a fallos. Naturalmente, si hablamos de replicar servidores, hablaremos de un clúster d e alta disponibilidad (ver Figura 2). Sistemas tolerantes a fallos los podemos encontrar en entornos muy críticos, tales como una central nuclear o el sistema de navegación de una aeronave moderna.

Figura 1. Sistema HA de 2 nodos



1.1. Sistemas de alta disponibilidad y sistemas tolerantes a fallos

En un sistema tolerante a fallos, cuando se produce un fallo hardware, el hardware asociado a este tipo de sistema es capaz de detectar el subsistema que falla y obrar en consecuencia para restablecer el servicio en segundos (o incluso décimas de segundo).

El cliente del servicio no notará ningún tiempo de fuera de servicio. En los sistemas de alta disponibilidad existen los tiempos de fuera de servicio; son mínimos pero existen, van desde 1 minuto o menos hasta 5 o 10 minutos, según sea el caso. En teoría esta es la única diferencia entre ambos, pero en los últimos años, se ha ido acercando la idea de alta disponibilidad a la idea de tolerancia a fallos, debido al abaratamiento de hardware, y de ciertas tecnologías que han ido surgiendo. Estas tecnologías han evolucionado de tal forma, que han logrando que subsistemas donde había que recurrir a la alta disponibilidad ahora, se puede lograr tolerancia a fallos a bajo precio. De todos modos en un sistema (como veremos en la siguiente sección) hay muchos elementos y subsistemas, y algunos subsistemas tolerantes a fallos siguen siendo demasiados caros.

En la mayoria de los análisis, que se hacen de un sistema de servicio, si la aplicación puede estar un mínimo tiempo fuera de servicio, y podemos permitir que el cliente pierda la sesión o la conexión, temporalmente, la alta disponibilidad es una opción muy apropiada. Hay soluciones de alta disponibilidad en las cuales las conexiones se mantienen y las sesiones se recuperan.

2. SPOF (Single Point Of Failure ó punto simple de fallo)

La regla básica para hacer un sistema de alta disponibilidad es la replicación de elementos (Recordando la idea de RAID, Redundant Array of Independent Disks). Con SPOF se quiere hacer referencia a cualquier elemento no replicado y que puede estar sujeto a fallos; afectando con ello al servicio. Se debe evitar el SPOF en cualquier subsistema del sistema, ya que en caso de fallo puede peligrar el servicio de datos por culpa de un subsistema.

Por ejemplo, un adaptador de red que da acceso a un servidor a una red es un SPOF, una controladora SCSI también. Si en un entorno de servidores falla uno y no puede ser reemplazado fácilmente por otro, en los tiempos anteriormente mencionados, el servidor se considerará como SPOF.

En el caso de alta disponibilidad (con hardware y software adecuado) se puede reemplazar un adaptador de red o un servidor automáticamente. No sólo los servidores han de ser redundantes si no los elementos que facilitan el servicio, como routers, bridges o la propia red local.

Es conveniente olvidar la redundancia a cierto nivel, pues habrá un nivel en que la alta disponibilidad se hace extremadamente cara, consideremos que sólo tenemos un único

edificio para nuestro sistema, perfectamente podemos considerarlo como SPOF (en caso de incendio en el edificio), pero replicar en otro edificio el sistema, se puede salir de presupuesto.

3. Servicio de datos

El servicio de datos y sus interrupciones es lo que nos ha llevado a adoptar medidas con tolerancia a fallos o alta disponibilidad. En alta disponibilidad, se denomina servicio de datos a un determinado servicio y los recursos necesarios para que el servicio sea ofrecido a un cliente (grupo de recursos). En otros entornos HA, se denominan logical host o software packages a la unión de servicio de datos y grupo de recursos asociados a este.

Estos grupos de recursos han de implementar mecanismos necesarios, para que sean completamente flexibles entre nodos del clúster; es decir puedan ser suplantados o conmutados físicamente entre restantes nodos sin que el servicio de datos cambie en absoluto. Esta “virtualización” del recurso va a permitir que un nodo pueda suplantar la función de otro dentro del clúster. El servicio de datos sólo sera penalizado con el tiempo de conmutación necesario para los grupos de recursos y la puesta en marcha del servicio de datos.

Como ejemplos: una dirección virtual (en TCP/IP, IP Aliasing) para que la dirección del servicio de datos pueda migrar entre nodos. Debe considerarse el servicio de datos del clúster como una entidad única capaz de ofrecer servicios. El software de alta disponibilidad debe ser capaz de abstraer el clúster que lo soporta y ofrecer dichos servicios.

4. Dinámica HA

Se considera dinámica HA a todas las reconfiguraciones del clúster que garanticen la máxima disponibilidad del servicio de datos. Esta dinámica esta orientada a los nodos integrantes del clúster y la forma en la cual el clúster responde. Se denomina de la siguiente manera:

Failover

Es un término genérico que se usa cuando un nodo debe asumir la responsabilidad de otro nodo, importar sus recursos y levantar el servicio de datos. Se ha de

entender que una situación de failover es una situación excepcional, para cual la alta disponibilidad ha sido concebida, el fallo de un nodo. Si sólo queda un nodo en el cluster, tras los fallos de los demás, estaremos en un SPOF hasta que el administrador del sistema, verifique y restaure el clúster. También se ha de entender que el servicio de datos sigue levantado, que es el objetivo de la alta disponibilidad.

Takeover

Es un failover automático se produce cuando un nodo nota un fallo en el servicio de datos. Para ello debe haber cierta monitorización con respecto al servicio de datos. El nodo que se declara fallido es forzado a ceder el servicio y recursos, o simplemente eliminado.

Switchover o Giveaway

Es un failover manual, consiste en ceder los recursos de un servicio de datos y este mismo, a otro nodo del clúster, mientras se realizan ciertas tareas administrativas. A este procedimiento se le denomina “Node outage”.

Splitbrain

Para la gestión de un clúster HA es necesario un mecanismo de comunicación y verificación entre nodos integrantes. Por este mecanismo, cada nodo debe gestionar los recursos que corresponden a cada uno, segun el estado del clúster; a su vez cada nodo debe hacer chequeos o latidos (beats) a sus compañeros.

Un Splitbrain (división de cerebros) es un caso especial de failover, en el cual falla el mecanismo de comunicación y gestión de un clúster de dos nodos. Es una situación en la cual cada nodo cree que es el único activo, y como no puede saber el estado de su nodo compañero, tomará acciones en consecuencia, forzando un takeover.

Esta situación es peligrosa, los dos nodos intentaran apropiarse de todos los recursos, incluyendo el servicio de datos. El peligro aumenta sobre todo cuando tenemos recursos delicados, como recursos de almacenamiento, ya que cada nodo podría tomar y escribir por su cuenta, y quebrar a integridad de datos.

Para evitar este problema, cada nodo debe actuar de una forma prudente, y utilizar los recursos compartidos como señal de que se esta vivo. La forma de proceder de

cada nodo es similar, ya que cada uno cree que su compañero ha desaparecido. Después de que un nodo aprecia este problema, tiene indicado que reserve un recurso llamado quorum. Un recurso quorum, es un recurso compartido, que se ha preestablecido en ambos nodos como tal. Este recurso es un recurso exclusivo, sólo un nodo del cluster puede reservarlo. Como este recurso sólo puede ser reservado por un nodo, el nodo que llegue tarde a la reserva del recurso, entiende que debe abandonar el clúster y ceder todos sus recursos. El quórum es utilizado como simplemete, método de decisión.

Otra forma de evitar esta situación, un poco mas violenta, es que un nodo elimine a su compañero; el primero que apague a su compañero se queda con todos los recursos. Es un mecanismo muy brusco, pero muy eficaz. Para este caso existen tarjetas especializadas en esta tarea.

4.1. Grupos de recursos de un servicio de datos

Pensando en un servicio de datos cualquiera, se puede pensar en unas serie de recursos que va a utilizar. Por ejemplo, imaginemos un servicio de datos de servicio web como puede ser apache, este va a necesitar un nodo donde va a ser ejecutado (ciclos de CPU, memoria), un sistema de ficheros donde guardar toda la información web y por supuesto una red donde poder responder a las peticiones de servicio. Estos recursos deben atender a cierta flexibilidad, donde la flexibilidad es la capacidad del recurso de ser estático virtualmente y ser dinámico físicamente. A continuación se enumeraran los distintos tipos de recursos que son interesantes en un clúster HA.

4.1.1. Recursos computacionales

Los recursos computacionales pueden ser considerados a nivel de CPU, nodo o clúster. Son los recursos que permiten que el programa que se encarga de ofrecer servicio de datos pueda ser ejecutado. Si tenemos varias CPU, varios nodos o varios clústers estos deberán tener una copia del programa del servicio de datos en memoria.

En la HA para Linux este recurso se considera a nivel de nodo, donde el servicio de datos va a estar situado en un nodo determinado (como máster del servicio). El software de alta disponibilidad es quien decide que nodo va a alojar que servicio de datos dependiendo del estado del clúster.

4.1.2. Recursos de comunicaciones

Normalmente el servicio de datos va a ser accedido mediante una red de comunicaciones. Las interfaces de red así como la pila de protocolos de red, deben ser capaces de responder a varias direcciones de red con el fin de dar flexibilidad al servicio de datos; es decir virtualizar el servicio. En el caso de redes TCP/IP el servicio de datos sera accedido mediante una dirección IP y un puerto; para que el servicio de datos pueda residir físicamente en cualquier nodo se debe utilizar IP's virtuales (IP aliasing) para que esto sea posible.

Si se utiliza una red Ethernet-TCP/IP y el NIC (network interface card) no es capaz de cambiar la dirección MAC (media access controler), se puede llegar aun problema con las tablas ARP de los demás elementos de red; ya que se debe obligar a los clientes o routers de la LAN a actualizar la nueva dirección MAC para la IP de servicio.

4.1.3. Recursos de almacenamiento

El almacenamiento de los datos del servicio de datos es quizás uno de los puntos mas delicados de la alta disponibilidad. Pues en ellos tendremos la aplicación que se usará para el servicio de datos junto con los datos.

El almacenamiento suele ser el recurso mas complicado de virtualizar en configuraciones clásicas; ya que suele ser un medio SCSI compartido con muchos discos y muchos elementos candidatos a SPOF. En configuraciones hardware más modernas no hay tanta problemática; las arquitecturas SAN (storage area network) permiten que los recursos de computación accedan por red SAN a los recursos de almacenamiento. Los recursos de almacenamiento suelen ser servidores de archivos en con discos en RAID y backup integrado con interfaz FiberChannel. Al estar en un entorno SAN permite acceder a estos recursos a más de un nodo de forma muy flexible.

El recurso de almacenamiento además de ser flexible debe ser independiente para las necesidades de cada servicio de datos. Surge el concepto de grupos de volúmenes; un grupo de volúmenes es un conjunto de volúmenes que pertenecen a un servicio de datos en concreto. Cada volumen es un espacio de almacenamiento que el servicio de datos utilizará para sus propósitos con independencia de otros volúmenes.

Es vital que el recurso de almacenamiento sea capaz de mantener la integridad de los datos y el tiempo de recuperación ante un fallo sea mínimo. Ante estos problemas surge las técnicas de journaling para la gestión de los datos.

5. Entorno HA para Linux

Linux como sistema operativo debe de ofrecer una serie de facilidades, para que los sistemas de alta disponibilidad puedan integrarse correctamente tal y como se hace en otros entornos. Estas facilidades están relacionados directamente con el entorno hardware y el software para las aplicaciones de alta disponibilidad. En esta sección se comentaran los diversos subsistemas que ayudan a “virtualizar” recursos, así como subsistemas para evitar SPOF.

5.1. Sistemas de computación

Actualmente Linux es una plataforma de computación y supercomputación muy consolidada. Se puede afrontar la idea de eliminar SPOF en los sistemas de computación, pero sólo a partir de un nivel que Linux como sistema operativo pueda obrar. Es decir que si consideramos la computación al nivel de nodos y superiores no habrá problemas, pero si consideramos a nivel de procesadores y memorias, es mas un problema muy cercano a una solución hardware tolerante a fallos.

5.2. Sistemas de red

Linux puede presumir de tener una de las pilas TCP/IP mas completas y estables que existen actualmente. La característica de IP Aliasing de Linux, permite asignar varias direcciones IP a una misma interfaz; esto nos permite poder levantar una dirección IP, exclusivamente, en un nodo del clúster. Si algún nodo perece con su IP, cualquier otro nodo del clúster, puede tomar el testigo.

También, Linux es capaz de actualizar sus tablas de rutas dinámicamente, gracias a demonios de enrutamiento tales como ospfd. ospfd es un demonio de enrutamiento que implementa el algoritmo de enrutamiento OSPF. Con estos servicios el clúster puede ser consciente de la topología de la red y reaccionar ante las caídas de líneas de comunicaciones o nodos.

5.3. Sistemas de almacenamiento

En los sistemas de almacenamiento existe una doble problemática:

1. SPOF en los elementos de almacenamiento.

2. Consistencia ante un crash o caída.

El SPOF se soluciona con sistemas de discos capaces de hacer redundantes los datos, así como de sustituir un disco dañado por otro en reserva.

La consistencia se consigue revisando la integridad de los datos; la consistencia debe ser establecida en el mínimo tiempo posible ya que el servicio de datos depende de él.

5.3.1. Sistemas de disco

5.3.1.1. Redundant Array of Independent Disks (RAID)

Un subsistema RAID nos va a permitir eliminar SPOF de los recursos de almacenamiento.

Actualmente el kernel de Linux soporta RAID 0,1 y 5 con el driver MD. Además de RAID software también soporta gran numero de controladoras SCSI y ATA100 que ofrecen volúmenes de RAID por hardware. Compaq e IBM han colaborado mucho en este aspecto, creando drivers para sus productos. También cabe mencionar los sistemas LAN-Mirror que son una opción barata para conseguir un medio compartido y eliminación de SPOF replicando por red. Como ejemplos cabe destacar DRBD, NBD y ENBD.

5.3.1.2. Logical Volume Management (LVM)

El software de gestión de volúmenes permite exportar los volúmenes de grupos de discos para que el servicio de datos pueda hacer uso de el. Logical Volume Management inicialmente adoptado por IBM luego por la OSF, esta en desarrollo. No sólo es capaz de cambiar el tamaño de discos lógicos sino también de sistemas de ficheros ext2.

5.3.2. Sistemas de ficheros con journal

Journaling es una técnica que nació de las bases de datos y se ha ido incorporando a los sistemas de ficheros. Cuando un sistema de ficheros sufre una caída, dado a un fallo del sistema, este se chequea al completo corrigiendo las inconsistencias. Con journal se lleva una cuenta de que se ha ido modificando en el sistema de ficheros, ya que a la hora de chequearlo sólo comprobara la inconsistencias de unos pocos ficheros y directorios.

El tiempo de puesta en consistencia de un sistema de ficheros disminuye considerablemente. Además se incorporan técnicas como árboles B y Hashes para un acceso mas rápido a ficheros.

A continuación se enumeran los sistemas de ficheros mas familiares y apropiados para la HA en Linux:

ext3

Es el clásico ext2 pero con el añadido de una partición de log, para llevar el journal. Hace journal de datos y metadatos. Por lógica es algo mas lento que ext2 en acceso pero mas rápido en tiempo de recuperación de consistencia. Lo realmente interesante es que hace journal de datos y metadatos.

Reiserfs

Es un sistema de ficheros creado desde 0 con la idea de sacar partido a arboles B, hashes y técnicas de journal actuales. Es un sistema de ficheros realmente rápido en lo que a Linux se refiere. Sólo hace journaling de datos. Actualmente sus creadores están desarrollando Reiser4 bajo el patrocinio de la DARPA.

JFS

IBM lo ha puesto bajo GPL y ha sacado su primera versión con calidad de producción. De momento desarrollan con bastante independencia, de versiones actuales del kernel, ya que sacan parches para versiones especificas.

XFS

Silicon ha sacado releases con calidad de producción bajo GPL. Casi tan rápido como reiserfs. Esta muy bien integrado con Linux, teniendo en consideración otros subsistemas como puede ser quota o NFS. Sacan parches cada una o dos versiones del kernel. Una opción muy a considerar, en un futuro.

6. Soluciones HA para Linux

Con las necesidades anteriormente mencionadas, Linux no podía quedarse atrás. De un tiempo a esta parte han ido surgiendo proyectos y soluciones para Linux, en las cuales algunas destacan su elegancia y sencillez en comparación con sistemas comerciales. En esta sección se va a comentar los mas conocidos y los mas utilizados. Se esta recibiendo mucha cooperación de distribuciones como VA,SuSE,Red Hat o Conectiva, no debería ni decirse que los esfuerzos de Debian no se quedan atrás.

6.1. Heartbeat

Es la navaja suiza de la alta disponibilidad para Linux, es una herramienta que permite crear un clúster de alta disponibilidad. De momento el clúster sólo soporta 2 nodos, permite crear grupos de recursos y cambiar estos grupos de recursos fácilmente entre nodos. Carece de herramientas de monitorización del servicio de datos, que se consigue con otras herramientas como mon.

6.2. ldirectord y LVS (linux virtual server)

LVS permite crear un clúster de balanceo de carga, en el cual hay un nodo que se encarga de gestionar y repartir las conexiones (nodo máster LVS) entre todos los nodos slave del clúster. El servicio de datos debe residir en todos los nodos slave. LVS puede llegar a soportar sin problemas hasta 200 nodos slave.

ldirectord es un demonio que se ejecuta en el máster LVS, que se encarga de testear el servicio de datos de los nodos slave y eliminarlos e insertarlos en el clúster dinámicamente, si surge algún problema o si se repone el servicio según sea el caso.

6.3. Pirahna

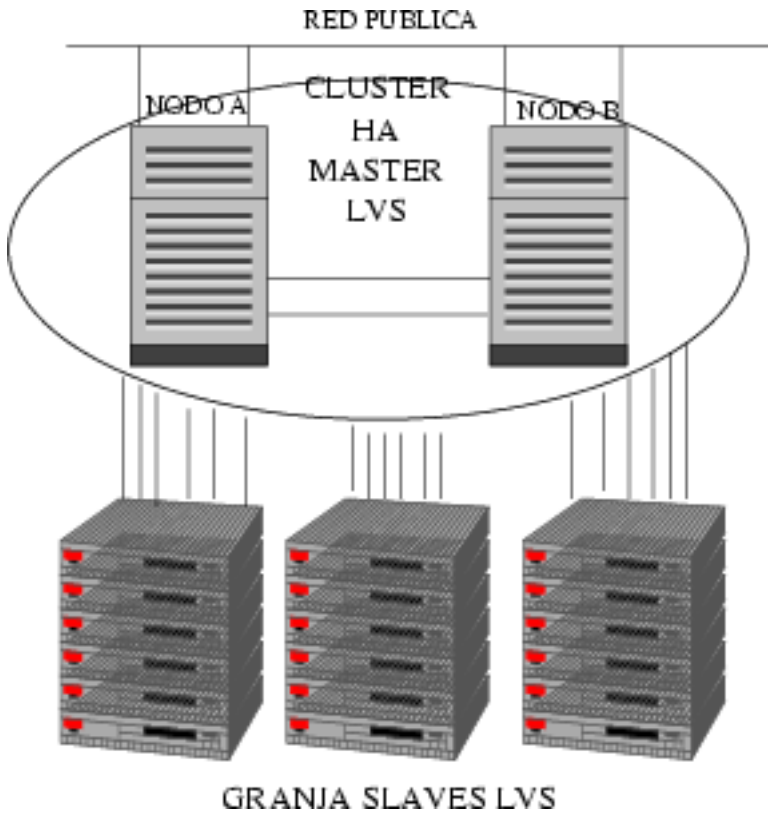
Es el nombre que Red Hat ha dado a su solución basada en LVS, el añadido es una interfaz para configurarlo.

6.4. UltraMonkey

Es una solución creada por VA Linux que se basa en LVS y Heartbeat para ofrecer clústers de alta disponibilidad y balanceo de carga (ver Figura 2). El nodo máster LVS

se pone en alta disponibilidad ya que es el único SPOF. Además incorpora una interfaz para configurar el clúster.

Figura 2. Sistema HA con LVS



6.5. Kimberlite

Creada por Mission Critical Linux, es una solución que soporta un clúster de 2 nodos. Permite fácilmente, definir un dispositivo de quórum, monitorizar los servicios de datos, así como gestionarlo. Una solución completa bajo GPL.

7. Copyleft

Bibliografía

Harald Milz, *Linux High Availability HOWTO*, 22 de diciembre de 1998.

Linux-HA Web Site: http://linux-ha.org.

Mon Web Site: http://www.kernel.org/software/mon.

Reiser fs Web Site: http://www.namesys.com.

Philipp Reisner, *DRBD Web Site: http://www.complang.tuwien.ac.at/reisner/drbd*.

Phil Lewis, *A high-availability cluster for Linux: http://linuxjournal.com/issue64/3247.html*.

NBD Web Site: http://www.it.uc3m.es/~ptb/nbd/.

LVM Web Site: http://www.sistina.com/lvm/.

GFS Web Site: http://www.sistina.com/gfs/.

EXT3: ftp://ftp.uk.linux.org/pub/linux/sct/fs/jfs/.

JFS Web Site: http://www-124.ibm.com/jfs/index.html.

XFS Web Site: http://linux-xfs.sgi.com/projects/xfs/.

UltraMonkey Web Site: http://ultramonkey.sourceforge.net/.

Pirahna Web Site: http://ha.redhat.com.

Kimberlite Web Site: http://oss.missioncriticallinux.com/projects/kimberlite/.

LVS Web Site: http://www.linuxvirtualserver.org.

Gracias al equipo de desarrollo de linux-ha, en especial a Alan Robertson y a Horms quienes han hecho un trabajo excelente y constante en el área de la alta disponibilidad para Linux.

Gracias al grupo SERA, sin el cual este documento no sería posible.