Instalación y Configuración de un Cluster de Alta Disponibilidad bajo Linux*

Rafael A. García Leiva
Universidad Autónoma de Madrid

12 de abril de 2003

1. Introducción

Tanto en el mundo empresarial, como en el mundo académico, existen ciertas aplicaciones que dada su naturaleza deben proporcionar un servicio ininterrumpido de 24 horas al día, 7 días a la semana. Piénsese por ejemplo en un servidor de bases de datos o en un servidor de páginas web. Para conseguir estos niveles de disponibilidad se suele utilizar una configuración avanzada de hardware y de software denominada *Cluster de Alta Disponibilidad*.

Un cluster de alta disponibilidad es un conjunto de dos o más máquinas, que se caracterizan porque comparten los discos de almacenamiento de datos, y porque están constantemente monitorizándose entre sí. Si se produce un fallo del hardware o de las aplicaciones de alguna de las máquinas del cluster, el software de alta disponibilidad es capaz de rearrancar automáticamente los servicios que han fallado en cualquiera de las otras máquinas del cluster. Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original. Esta capacidad de recuperación automática de servicios nos garantiza la integridad de la información, ya que no hay pérdida de datos, y además evita molestias a los usuarios, que no tienen por qué notar que se ha producido un problema.

No hay que confundir los clusters de alta disponibilidad con los *clusters de alto rendimiento*. Un cluster de alto rendimiento es una configuración de equipos diseñada para proporcionar capacidades de cálculo mucho mayores que la que proporcionan los equipos individuales (véanse por ejemplo los sistemas de tipo *Beouwulf* [?]), mientras que los clusters de alta disponibilidad están diseñados para garantizar el funcionamiento ininterrumpido de ciertas aplicaciones.

Existen muchos paquetes software para la instalación y configuración de un cluster de alta disponibilidad bajo Linux. Podemos encontrar enlaces a la mayoría de estos paquetes en la página web del proyecto Alta Disponibilidad bajo Linux (*The Linux High-Availability Project* [?]). Sin embargo, y con el objetivo de estudiar un caso práctico, en este artículo nos vamos a centrar en cómo instalar y configurar uno de estos paquetes: *Kimberlite* [?].

La tecnología Kimberlite para la instalación de clusters de alta disponibilidad es proporcionada por la compañía *Mission Critical Linux[?]* de manera gratuita y bajo licencia GPL. El software Kimberlite ha sido utilizado como base para otras soluciones de alta disponibilidad más avanzadas, pero de caracter comercial, como son *Covolo*, vendida por la propia Mission Critical Linux, y *Cluster Manager*, vendida por Red Hat como parte del producto *Red Hat Advance Server*.

2. Descripción del Hardware

Todos los equipos que componen un cluster de alta disponibilidad (dos equipos en caso de utilizar Kimberlite) han de estar preparados para ejecutar cualquiera los servicios ofertados por el cluster, y además, utilizando información totalmente actualizada. Una manera de conseguir este requerimiento es mediante el uso compartido de los mismos discos de datos por parte de ambos equipos. Cada servicio ofertado por el cluster dispone de su propia partición dentro de estos discos compartidos, y el equipo

*Trabajo financiado por el MCyT

que tenga arrancado el servicio monta la partición correspondiente. Para evitar la posible corrupción de los datos, por ejemplo porque ambos equipos intenten modificar simultáneamente los datos, el software de alta disponibilidad nos garantiza que sólo uno de los equipos monta a la vez cada partición.

Los discos compartidos también son utilizados por el software de alta disponibilidad para monitorizar el estado de los equipos del cluster. Para ello se utiliza una partición especial denominada *partición de quórum*. Cada miembro del cluster escribe periódicamente su estado (si está activo, que servicios está ofertando, etc.) en la partición de quórum, y lee la información escrita por el otro miembro del cluster. Si un equipo falla durante cierto tiempo al escribir esta información, posiblemente se trate de un problema con el hardware o software de ese equipo. En este caso, se utiliza un segundo mecanismo (llamado canal de latido o *heartbeat*) para comprobar si realmente el equipo está fallando. Este canal puede ser un enlace Ethernet, o simplemente un cable serie entre ambos equipos.

Podemos optar por varias configuraciones hardware a la hora de instalar Kimberlite. Estas configuraciones van desde una instalación mínima, con aquellos elementos hardware mínimos que requiere un cluster, hasta una configuración de máxima seguridad, que proporcione redundancia de todos los componentes hardware. El administrador del sistema tiene libertad, según las necesidades y el presupuesto disponible, de seleccionar la configuración más adecuada. En cualquier caso, se recomienda utilizar hardware de calidad, ya que los fallos de hardware suelen ser, en la mayoría de las ocasiones, la causa de la parada de un sistema.

2.1. Instalación Mínima

La configuración mínima de hardware requerida para instalar un cluster bajo Kimberlite se compone de los siguientes elementos (en la Figura 1 se puede ver un esquema de esta configuración):

- dos equipos, encargados de ejecutar las aplicaciones ofertadas por el cluster,

- uno o más discos SCSI, compartidos por ambos equipos del cluster, y que almacenan los datos de las aplicaciones, además de contener las particiones de quórum, y

- un canal de comunicación, por ejemplo Ethernet, que se utiliza para comunicar a los equipos con el exterior, y para que se puedan monitorizar entre sí (canal de latido).

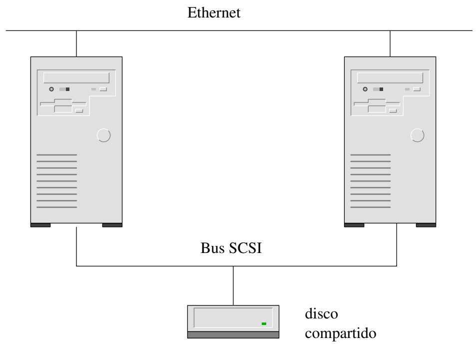

Figura 1.– Configuración Mínima

Para los discos SCSI compartidos se suele utilizar un sistema RAID (*Redundant Array of Independent Disks* - array redundante de discos independientes) que garantice la disponibilidad de los datos en caso de fallo de alguno de los discos. Sin embargo, y para ahorrar costes, en lugar de un RAID se puede utilizar simplemente una pila de discos sin ningún tipo de redundancia. Nótese que las capacidades RAID que implementa el núcleo de Linux no se pueden utilizar junto con Kimberlite.

Otra manera de ahorrar algunos costes consiste en utilizar un conmutador (*switch*) KVM que con un solo monitor, teclado y ratón nos permita gestionar ambos equipos.

2.2. Instalación Avanzada

La configuración hardware descrita en el apartado anterior presenta numerosos puntos débiles que pueden hacer que nuestro sistema falle, incluso utilizando un software de alta disponibilidad. Por ejemplo, si alguno de los discos SCSI compartidos falla, los servicios alojados en dicho disco no se pueden seguir ofertando.

Una configuración hardware más avanzada, tolerante a fallos, incluiría los siguientes elementos (en la Figura 2 se puede ver un esquema de esta configuración):

- discos SCSI compartidos con soporte RAID por hardware, que proporcione un sistema de redundancia de datos,

- dobles controladoras SCSI, que permitan al bus seguir funcionando incluso en el caso de que una de las controladoras falle,
- *single-initiator* SCSI, que nos aísle ambos segmentos del bus, evitando posibles interferencias,
- una tarjeta de red Ethernet adicional, usada como canal de latido,
- un cable serie conectado a ambos equipos, usado como canal serie de latido,
- sistemas de alimentación ininterrumpida UPS, para prevenir posibles cortes en el suministro eléctrico,
- dos *power switch*, que permitan a cada equipo del cluster reiniciar al otro equipo mediante hardware, de esta manera se garantiza la integridad de los datos, ya que tenemos la seguridad de que sólo uno de los equipos monta a la vez cada partición del disco compartido.

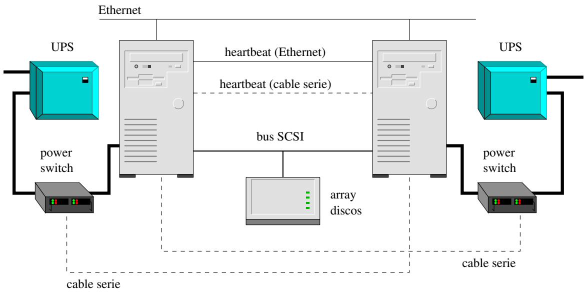

Figura 2.– Configuración Avanzada

2.3. Consideraciones sobre el bus SCSI

Como hemos visto, los discos compartidos SCSI son el elemento central de toda instalación de alta disponibilidad bajo Kimberlite. Por ello, debemos poner especial énfasis en cómo configurar correctamente estos elementos. Aspectos que hay que tener en cuenta a la hora de configurar el bus SCSI son:

- los discos compartidos han de estar conectados al segmento LVD de las controladoras SCSI,
- ambos extremos del bus han de estar terminados físicamente, es decir, tenemos que terminar el bus por hardware usando terminadores, no podemos utilizar la opción de terminar automáticamente el bus que viene con el software de configuración de la tarjeta,
- los equipos al arrancar no deben reiniciar el bus, por lo que las tarjetas deben tener esta opción desactivada,
- todos los dispositivos del bus han de tener un identificador único, tanto los discos como las propias controladoras,
- no es conveniente que los discos del sistema se encuentren en el mismo bus SCSI que los discos de datos compartidos, y
- en cualquier caso, los discos compartidos han de tener el mismo nombre hardware (`/dev/sd??`) en ambos equipos.

Este último punto es importante. Lo que nos está diciendo es que hay que tener cuidado cuando se utilizan dos equipos con distinto hardware en un mismo cluster. Si por ejemplo uno de los equipos utiliza una controladora SCSI para el sistema operativo, y el otro una controladora IDE, es posible que en primer equipo los discos de datos figuren bajo el nombre `/dev/sdb`, mientras que en el segundo figuren como `/dev/sda`. En este caso, el software Kimberlite no funcionaría correctamente. Lo que se recomienda es usar discos IDE para el sistema operativo y el software de aplicación, y dejar los discos SCSI para los datos compartidos.

3. Instalación y Configuración del Software

Una vez nos hemos decidido por una configuración hardware, comprado los equipos, e instalado físicamente los mismos, podemos continuar con la instalación del software en el cluster. Primero tenemos que instalar y configurar una distribución Linux en ambos equipos del cluster, a continuación hay que instalar y configurar el software Kimberlite, y finalmente, hay que definir y configurar los distintos servicios ofertados por el cluster. En las siguientes secciones se describen en detalle estos pasos.

3.1. Configuración del Sistema Operativo

El primer paso para instalar y configurar el software de alta disponibilidad en nuestro cluster es instalar una distribución estándar Linux. Para este artículo, hemos probado a instalar Kimberlite 2.0.0 bajo la distribución RedHat Linux 7.3, no habiendo encontrado ningún tipo de problemas de incompatibilidades software. Si se quiere utilizar otra versión de la distribución RedHat Linux, o incluso otra distribución de Linux distinta, se recomienda antes consultar la página web de Mission Critical Linux para evitar potenciales problemas de incompatibilidad.

Finalizada la instalación de la distribución Linux, resulta conveniente revisar que se han reconocido y configurado correctamente todos los dispositivos hardware de nuestros equipos. Por ejemplo, podemos revisar la lista de mensajes producidos durante el arranque del sistema (con el comando `dmesg`), o ejecutando `cat` sobre fichero `/proc/devices` y comprobando que existen entradas para los dispositivos SCSI (`sd`), y puertos serie (`ttyS`). También es importante que exista una entrada para los dispositivos de tipo raw (`raw`).

Nótese que para que nuestro cluster funcione de manera adecuada, resulta fundamental tener correctamente configuradas las tarjetas de red en ambos equipos. Antes de continuar, se recomienda revisar que ambos equipos tienen una dirección IP correcta, una máscara de subred correcta (esto es importante), y que disponemos de un servidor DNS (o de algún otro mecanismo) que nos asigne correctamente los nombres de máquinas (*hostnames*) a las correspondientes direcciones IP. En caso de utilizar una segunda tarjeta de red como canal de latido, se recomienda configurar esta tarjeta utilizando las direcciones IP de alguna de las subredes privadas existentes (por ejemplo `192.168.0.0`), y utilizar el fichero `/etc/hosts` para asignar los nombres de máquina a éstas direcciones privadas. Para no complicar la configuración de los servicios de red, se recomienda que la tarjeta de red externa esté asociada al dispositivo `/dev/eth0` y la interna a `/dev/eth1`.

Kimberlite basa su funcionamiento en la tecnología *IP-Aliasing* (para más información sobre IP-Aliasing y cómo funcionan las direcciones IP flotantes consultar la *"IP-Aliasing Mini-HOWTO"* por *Harish Pillay*, disponible en la página web de *The Linux Documentation Project* [?]). En la serie 2.4 del núcleo de Linux se puede activar el soporte de direcciones IP flotantes en tiempo de compilación, de hecho, el núcleo incluido con la distribución RedHat 7.3 Linux trae este soporte ya activado. Aun así, si queremos estar completamente seguros de que nuestro núcleo soporta direcciones IP flotantes, podemos realizar un sencillo experimento, podemos configurar nuestra tarjeta de red `eth0` para que responda a una dirección IP flotante. Para ello probar a ejecutar:

```
ifconfig eth0:0 192.168.0.1 \
netmask 255.255.255.0
```

Cuando tengamos instalado y configurado el sistema operativo, tenemos que crear las particiones en los discos SCSI compartidos. Hay que crear una partición por cada uno de los servicios ofertados, y dos particiones adicionales de unos 10Mb cada una para los servicios de quórum. A continuación hay que crear los correspondientes sistemas de ficheros en las nuevas particiones, excepto en las particiones de quórum, que son tratadas como dispositivos de tipo raw. Para que las particiones de quórum sean vistas como dispositivos raw, tenemos que utilizar el comando `raw`, por ejemplo:

```
raw /dev/raw/raw1 /dev/sda1
raw /dev/raw/raw2 /dev/sda2
```

Podemos comprobar que los dispositivos raw han sido asignados correctamente ejecutando: `raw -qa`.

Finalmente tenemos que modificar el fichero `/etc/sysconfig/rawdevices` para que cuando arranquemos nuestro equipo, se configuren correctamente los dispositivos raw recién creados. Añadir al fichero las siguientes líneas:

```
/dev/raw/raw1 /dev/sda1
/dev/raw/raw2 /dev/sda2
```

Opcionalmente también podemos modificar la configuración del sistema de arranque (grub o lilo) para reducir el tiempo que tardan las máquinas en arrancar (opción *boot time*). De esta manera conseguimos que las máquinas recuperen los servicios perdidos más rápidamente.

3.2. Instalación de Kimberlite

Para instalar el cluster de alta disponibilidad descrito en este artículo hemos utilizado la versión 2.0.0 de Kimberlite. La mejor manera de instalar el software Kimberlite es compilándolo desde el código fuente original. Para ello nos descargamos el fichero `kimberlite-2.0.0.tar.gz` de la página web de Mission Critical Linux, y lo compilamos e instalamos (en ambas máquinas del cluster) utilizando el conocido procedimiento:

```
./configure
make
make install
```

Este procedimiento nos instala el software bajo el directorio `/opt/cluster`, y los ficheros de configuración bajo `/etc/opt/cluster`. Como es habitual, si queremos cambiar el directorio de destino del software, en lugar de la orden `configure`, deberemos utilizar:

```
./configure
--prefix=/otro/directorio
```

Una vez compilado e instalado el software, tenemos que crear un nuevo grupo de usuarios llamado `cluster`, y hacer que todos los ficheros y subdirectorios que cuelgan de `/opt/cluster` pertenezcan a este nuevo grupo. Por ejemplo ejecutar:

```
chgrp -R cluster /opt/cluster
```

A continuación tenemos que configurar correctamente en ambas máquinas el software recién instalado. La configuración de Kimberlite reside en el fichero `/etc/opt/cluster/cluster.conf`, sin embargo, se desaconseja editar a mano este fichero. El problema reside en que ambos equipos han de estar configurados exactamente de la misma manera, y que una copia de dicha configuración ha de ser almacenada en las particiones de quórum.

Para simplificar el proceso de configuración, Kimberlite nos proporciona la utilidad `member_config`, residente en el directorio `/opt/cluster/bin`. Esta utilidad nos hará algunas preguntas sobre cómo queremos configurar nuestro cluster, y realizará la configuración por nosotros. Conviene tener preparada la información que se nos solicita antes de ejecutar la utilidad. La información solicitada es: *hostname* y dirección IP de ambos equipos del cluster, dispositivos raw asociados a las particiones de quórum, número de canales de latido utilizados y tipo (Ethernet o serie), y puerto serie asociado al *power switch* (en caso de existir).

Cuando hayamos terminado de configurar el software, sólo nos queda inicializar las particiones de quórum. Para ello utilizamos la orden `diskutil -I`. Y una vez inicializadas las particiones, podemos arrancar el software de cluster.

El procedimiento de configuración descrito en los párrafos anteriores puede resultar un poco confuso, sobre todo la primera vez que se realiza. Por ello, para simplificar la tarea del administrador de sistemas, se propone seguir el siguiente procedimiento paso a paso:

En el servidor 1:

- ejecutar la utilidad `member_config` y responder a las preguntas que nos haga,
- inicializar la partición de quórum,

  ```
  diskutil -I
  ```

- comprobar que la partición ha sido inicializada correctamente,

  ```
  diskutil -t
  diskutil -p
  ```

- arrancar los servicios de cluster,

  ```
  service cluster start
  ```

En el servidor 2:

- ejecutar el la utilidad `member_config` indicándole que lea la configuración ya existente en la partición de quórum,

  ```
  member_config --init=/dev/raw/raw1
  ```

- arrancar los servicios de cluster,

  ```
  service cluster start
  ```

Para comprobar que los servicios de cluster están funcionando correctamente, podemos ejecutar la orden `clustat -i 2`, que nos mostrará en pantalla, cada 2 segundos, el estado de nuestro cluster.

3.3. Configuración de los Servicios

El objetivo final de instalar un cluster bajo Kimberlite es poder configurar un conjunto de servicios que ofrezcan datos y aplicaciones en alta disponibilidad. Para crear estos servicios tenemos que indicar los recursos que utilizan las aplicaciones y las propiedades de las mismas. Durante el proceso de configuración, la información que se nos solicitará es: el nombre del servicio, el *script* de arranque y parada, la partición o particiones (en los discos compartidos) donde residen los datos, los correspondientes puntos de montaje, y la máquina en la que se va a ejecutar por defecto la aplicación. A un servicio también se le puede asignar una dirección IP (flotante) que proporcione a los clientes un acceso transparente.

Kimberlite nos permite configurar nuestro cluster de dos maneras distintas, utilizando una configuración de tipo *activo-activo*, o una configuración de tipo *activo-espera*. En la configuración de tipo activo-activo ambos miembros del cluster ofertan servicios y ejecutan aplicaciones, mientras que en la configuración de tipo activo-espera sólo uno de los miembros del cluster ejecuta las aplicaciones, mientras que el segundo equipo queda en espera, preparado para arrancar el software en caso de que el otro equipo falle.

Son muchas las aplicaciones que se pueden utilizar bajo Kimberlite, por ejemplo un servidor de páginas web, una base de datos, un servidor de colas de impresión, etc. En esta sección veremos cómo instalar y configurar una de estas aplicaciones; vamos a ver cómo instalar un servidor web de alta disponibilidad.

Instalación de un servidor web Apache

Como ya hemos indicado, antes de definir el servicio tenemos que identificar algunos elementos de configuración. Para este caso concreto, estos elementos son:

- nombre del servicio: `httpd`
- *script* de arranque y parada: `/etc/init.d/httpd`
- partición de datos en el disco compartido: `/dev/sda1`
- punto de montaje: `/var/www`
- máquina en la que va a ejecutar el servicio por defecto: *servidor1*
- dirección IP flotante: 192.168.1.101

Una vez tenemos identificada toda la información necesaria para definir el servicio podemos empezar con la instalación y configuración del mismo. El primer paso consiste en instalar y configurar el software Apache en ambos equipos del cluster. Podemos comprobar que el servidor web está funcionando correctamente accediendo a la página de prueba que se instala por defecto. Cuando esté instalado el servidor web en ambos equipos, paramos el servicio (por ejemplo con `service httpd stop`), y realizamos la siguientes pasos:

En el servidor 1:

- montar la partición correspondiente a Apache en el disco compartido en un punto de montaje temporal,

  ```
  mount /dev/sda1 /mnt/tmp
  ```

- copiar los datos de nuestra página web al disco compartido,

  ```
  cp -r /var/www/* /mnt/tmp
  ```

- borrar los datos originales

  ```
  rm -rf /var/www
  ```

- desmontar la partición y volverla a montar en su destino final,

  ```
  umount /mnt/tmp
  mount /dev/sda1 /var/www
  ```

- comprobar que el servidor web sigue funcionando, y

  ```
  service httpd start
  ```

- parar el servicio y desmontar la partición.

En el servidor 2:

- borrar los datos originales,
- montar la partición en el directorio destino, y comprobar que el servicio funciona correctamente, y

- parar el servicio y desmontar la partición.

En el servidor 1:

- arrancar la herramienta de configuración de Kimberlite,

  ```
  /opt/cluster/bin/cluadmin
  ```

- añadir un nuevo servicio, y

  ```
  cluadmin>service add
  ```

- responder a las preguntas que se nos hacen, con la información sobre la configuración que habíamos preparado.

Cuando definimos un nuevo servicio con la herramienta `cluadmin`, Kimberlite añade la información de configuración del servicio a una base de datos de configuración del cluster, residente en el disco compartido y accesible para ambos equipos. Y a partir de ese momento podemos empezar a trabajar con el nuevo servicio.

Incompatibilidad con las scripts de arranque y parada de RedHat 7.3.

Existe un problema de incompatibilidad entre Kimberlite y las scripts de arranque y parada de servicios instaladas por RedHat 7.3. El problema reside en que estas scripts consideran como un error intentar parar un servicio que ya está parado. Por otro lado, cuando Kimberlite arranca un servicio, primero se asegura de que está correctamente parado invocando explícitamente una orden de parada. Al recibir un código de error por parte del script, Kimberlite asume que no se pudo realizar correctamente la parada, y por tanto, que el servicio no puede ser arrancado.

Para solucionar este problema, se puede crear un *script puente* que nos filtre esta condición de error. Por ejemplo, el script puente en caso del servicio `httpd` podría ser:

```
#!/bin/sh
/etc/init.d/httpd $1
exit 0
```

4. Mantenimiento

Para las tareas de mantenimiento del cluster, Kimberlite proporciona la herramienta en modo texto `cluadmin`. Con esta herramienta podemos administrar de una manera fácil y cómoda tanto el software de alta disponibilidad, como los servicios que hayan sido definidos.

La herramienta `cluadmin` nos permite arrancar y parar el software Kimberlite, crear una copia de seguridad de la configuración actual del cluster, y recuperar esta copia en caso de que algo haya salido mal. En cuanto a los servicios, `cluadmin` nos permite parar o arrancar cualquier servicio, modificar su configuración, o incluso eliminarlo. Además de lo anterior, `cluadmin` también nos permite monitorizar el estado del cluster.

Junto a la herramienta `cluadmin`, Kimberlite también nos proporciona una herramienta gráfica, basada en páginas web, para monitorizar de manera dinámica el estado de nuestro cluster. Sin embargo, en este artículo no vamos a entrar en los detalles de cómo configurar y usar esta herramienta gráfica (para más información, consultar la documentación que acompaña a Kimberlite).

También es interesante destacar la posibilidad que ofrece Kimberlite de migrar los servicios desde un equipo del cluster hacia el otro, de una manera totalmente transparente para los usuarios. Esto nos permitiría, por ejemplo, actualizar el sistema operativo, o incluso el hardware, de los equipos del cluster sin necesidad de parar los servicios, y sin que exista el peligro de que se corrompan los datos.

5. Conclusión

Kimberlite es uno de los mejores paquetes software de alta disponibilidad, distribuidos bajo licencia GPL, que existen actualmente para Linux. Es cierto que presenta algunas limitaciones importantes, como por ejemplo que no se garantice la alta disponibilidad de servicios individuales (sólo se garantiza la disponibilidad para el caso de fallos de las máquinas). Aun así, Kimberlite resulta adecuado para muchas aplicaciones, tales como servidores de base de datos, o servidores de World Wide Web con contenidos dinámicos.

Por último, es interesante mencionar que Kimberlite puede ser utilizado junto con otros paquetes software de alta disponibilidad para Linux, como el proyecto *Linux Virtual Server* [**?**], para proporcionar servicios de red de alta disponibilidad que sean escalables.

Referencias

[1] The Beowulf Project: http://www.beowulf.org

[2] The Linux High-Availability Proyect: http://www.linux-ha.org

[3] Kimberlite: http://oss.missioncriticallinux.com

[4] The Linux Documentation Project: http://www.tldp.org

[5] Linux Virtual Server: http://www.linuxvirtualserver.org/