Cluster Alta Disponibilidad sobre Plataforma GNU/LINUX (VERITAS)

Memoria – TFC – Plataforma GNU/LINUX

Trabajo Fin de Carrera

Ingeniería Técnica de Informática de Sistemas

Autor: Santiago Barrio González

Consultor: Joaquín López Sánchez-Montañes
Profesor responsable de la asignatura: Montse Serra Vizern

UOC
Universitat Oberta de Catalunya

A mi familia, amigos y muy especialmente
a mi esposa y a mis hijos, por confiar en mi
y darme todo su apoyo, cariño y comprensión.
Ellos son los verdaderos autores de este trabajo.
Gracias.

Índice de contenido

1. INTRODUCCIÓN

1.1 MOTIVACIÓN

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.

En la actualidad los sistemas informáticos de las entidades públicas y privadas son una pieza imprescindible para su correcto funcionamiento. Cada vez más empresas y organismos públicos confían procesos de negocio críticos en sistemas informáticos para mejorar la productividad y disponer de esta información crítica en un tiempo mínimo.

Confiar los procesos de negocio en los sistemas informáticos de la organización aporta muchas ventajas pero introduce nuevos puntos de fallo que necesitan estar controlados.

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.

Para evitar fallos de servicio en los sistemas informáticos de las organizaciones aparecen los sistemas de alta disponibilidad (HA). Un sistema de alta disponibilidad (HA) está formado por distintos componentes hardware y software que combinados proporcionan acceso de forma ininterrumpida a los procesos de negocio. En la actualidad, el uso de sistemas de alta disponibilidad esta muy extendido en las organizaciones y esto motiva y justifica el presente estudio.

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.

1.2 OBJETIVO

Este estudio pretende servir de introducción a los sistemas de alta disponibilidad (HA) y 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.

Se abordarán las bases teóricas y los aspectos prácticos de este tipo de sistemas utilizando como ejemplo el diseño e implantación de un sistema real sobre Veritas Cluster Server para un entorno productivo.

Se identificarán los diferentes problemas que aparecen en un proyecto de este tipo y se plantearán soluciones utilizando como ejemplo el diseño e implantación de un sistema real para un entorno productivo.

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.

Una vez alcanzados los conocimientos necesarios para acometer una instalación en cluster de una aplicación empresarial tipo, instalaremos, configuraremos el cluster HA montando sobre el mismo una aplicación tipo. Esta nos servirá como base para la implementación de esta solución sobre cualquier aplicación con una arquitectura similar.

Para alcanzar los objetivos del presente TFC deberemos instalar y configurar el cluster para conseguir que con él, como mínimo, clustericemos lo siguiente:

• Instalar una pequeña aplicación que conecte con una BBDD.
• Montar y configurar los recursos del cluster.
• Servidor de aplicaciones (jboss).
• Balanceador de carga (apache).
• BBDD (Mysql).
• Montar un filesystem en alta disponibilidad para que toda la información importante del aplicativo este protegida por el cluster y siempre disponible para la aplicación.
• Instalar el Veritas Operation Manager (VOM) para controlar el cluster y los recursos del cluster desde interfaz Web.

1.3 PROYECTO

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. Antes de la instalación y configuración del cluster, se llevara a cabo una introducción inicial a las diversas técnicas de clustering, su problemática y sus soluciones, finalizando con la instalación pormenorizada de una aplicación empresarial tipo con la siguiente arquitectura:

• Aplicación.
• Sevidor de aplicaciones (jboss).
• Balanceador (apache).
• Base de datos (MySQL).
• Filesystems compartidos en HA.

En este proyecto presentaremos soluciones sobre como poder montar una arquitectura de este tipo sobre un cluster de alta disponibilidad en plataforma GNU/LINUX.

Hay muchos tipos de soluciones de cluster HA para linux y hablaremos de gran parte de ellas, pero nos centraremos más en poner ejemplos sobre uno en particular, Veritas Cluster Server.

Los cluster de alta disponibilidad son la solución ideal si una aplicación necesita una seguridad sobre su correcto funcionamiento en todo momento. Nos permitirá la recuperación en tiempo real tras desastres desde cualquier lugar bien sea sobre el servidor o sobre cualquier otra parte de la arquitectura de la aplicación, como la BBDD o problemas sobre el balanceador apache por ejemplo.

Linux permite la creación de sistemas en cluster formados por al menos dos máquinas, en el que se pueden crear servicios en alta disponibilidad que pueden superar situaciones en las que una de las máquinas sufra un problema de pérdida de servicio. De esta forma, aunque una de las máquinas deje de estar disponible (por fallo hardware o software), el servicio puede continuar estando disponible desde la otra máquina sin que haya apenas corte en el servicio ofrecido, proporcionando alta disponibilidad de aplicaciones.

2. CONSIDERACIONES PREVIAS

2.1. GNU/Linux y el Software Libre

GNU/Linux es un sistema operativo compatible UNIX, multiusuario y multitarea. Su núcleo, el kernel Linux, fue diseñado a principios de los 90 por Linus Torvalds para los PCs 80x86 y compatibles de la época y, gracias a su código abierto y al desarrollo distribuido en Internet, ha sido adaptado a gran cantidad de arquitecturas, desde estaciones de trabajo RISC hasta PDAs como el IPac de Compaq o incluso a la consola de videojuegos PlayStation de Sony. GNU (acrónimo recursivo de GNU is Not Unix) por su parte, es un proyecto iniciado por Richard Stallman (otro de los "gurús" del software libre) a mediados de los 80 cuyo objetivo es el de conseguir un sistema operativo tipo UNIX completamente gratuito y con el código disponible bajo una licencia abierta. En principio, el kernel para GNU iba (y va) a ser Hurd, todavía en desarrollo, pero cuando Torvalds liberó las primeras versiones de Linux se vio claramente que se necesitaban el uno al otro, ya que el núcleo era la pieza que faltaba para poder "echar a andar" el sistema operativo de GNU, mientras que el kernel Linux de por si, sin utilidades ni librerías ni entorno operativo, no podía valerse por sí mismo. Así nació el binomio GNU (herramientas y entorno) / Linux (núcleo).

Se podría decir que el sistema GNU/Linux es el "buque insignia" del movimiento conocido como Software Libre. Este movimiento (casi una filosofía de vida) promueve el desarrollo cooperativo del software, por medio de la liberación bajo licencias abiertas del código fuente de los programas, de forma que cualquier persona en cualquier parte del mundo pueda aportar su "granito de arena". Existen gran cantidad de licencias dentro del mundo del soft libre, siendo las más importantes y extendidas de ellas la General Public License (GPL) de GNU, que prácticamente da permisos para hacer cualquier cosa con el programa (incluso cobrar por su distribución, siempre que se cumplan el resto de cláusulas) excepto derivar de él trabajos y que éstos no se liberen también bajo la GPL, ni que formen parte de software propietario (un programa propietario no puede enlazarse con una librería GPL); la Lesser General Public License (LGPL) también de GNU, similar a la GPL pero que si permite que un programa con licencia propietaria enlace con librerías LGPL; y la licencia BSD, que elimina prácticamente todas las restricciones de la GPL y LGPL, permitiendo que el código de un programa con licencia BSD sea incluido en un programa comercial sin problemas. Al final de este trabajo se incluyen enlaces a los textos de varias de estas licencias.

Cabe aclarar aquí que todas estas licencias bajo ningún concepto dan derecho a nadie de ADUEÑARSE del código: el concepto de copyright (derechos de autor) sigue presente en todas ellas y se proteje con especial cuidado. Lo que persiguen las licencias abiertas es dar al usuario final una serie de derechos y libertades sobre el software mucho mayores de las que dan las licencias propietarias, pero manteniendo siempre el autor del programa los derechos sobre su obra. Esta es la principal diferencia entre el software libre y el software de Dominio Público (el autor cede TODOS los derechos, incluido el copyright), el Freeware (se puede utilizar gratuitamente pero generalmente no se dispone del código fuente, y cuando se dispone su uso y modificación está restringido) y el Shareware (se puede utilizar libremente con ciertas restricciones o durante un cierto periodo de tiempo tras el que hay que registrarse, y el código fuente no está disponible).

Además del sistema operativo GNU/Linux, otros notables éxitos del software libre son el servidor de HTTP Apache (líder en el terreno de los servidores Web, por delante del IIS de Microsoft), el lenguaje de script en el servidor embebido en HTTP PHP (claro competidor frente al ASP de Microsoft y el JSP de Sun), el navegador multiplataforma Mozilla (derivado del código fuente del Netscape Navigator 4.7x que liberó Netscape), la "suite" ofimática multiplataforma y compatible con MS Office Open Office, y los entornos de escritorio GNOME y KDE (a pesar de los problemas de licencias que tuvo en el pasado por una librería de la que depende).

GNU/Linux y el software libre en general han pasado en los últimos años de ser considerados como poco más que "juguetes" para "locos de la informática", a formar parte clave de la estrategia comercial y la infraestructura de grandes empresas. Como ejemplo cabe destacar la investigación y
desarrollo de aplicaciones que están realizando empresas como IBM o SUN y su adopción del sistema operativo Linux, y el apoyo que está recibiendo también desde el entorno de las instituciones gubernamentales, donde cabe señalar el proyecto GPG (GNU Privacy Guard, una alternativa open source al programa de criptografía de clave privada PGP), que ha sido patrocinado por el gobierno alemán. Aquí en España habría que destacar el proyecto de modernización de los sistemas informáticos del Ministerio de Administraciones Públicas, llevado a cabo por la empresa madrileña Andago y basado íntegramente en software libre bajo GNU/Linux.

Todo el software que se va a analizar y discutir en este trabajo se distribuye bajo licencias abiertas salvo el software ce VERITAS CLUSTER SERVER del cual disponemos de una licencia de evaluación por 60 días. El resto de software se distribuye bajo licencias abiertas, principalmente la General Public License (GPL) de GNU, la licencia BSD y la de Apache. Son, por tanto, programas gratuitos y con el código fuente disponible.

2.2. Introducción a la Alta Disponibilidad

La alta disponibilidad, ya sea de servicios o de datos, puede ser alcanzada tanto haciendo uso de soluciones software como de soluciones hardware. La variedad que existe en ambos grupos es muy grande, pudiendo elegir siempre una solución que encaje perfectamente según nuestras necesidades. Veamos brevemente de forma introductoria en qué consiste cada una de ellas.
Las soluciones software nos permiten alcanzar alta disponibilidad instalando y configurando determinadas herramientas y aplicaciones que han sido diseñadas para tal efecto.

En lo que respecta a alta disponibilidad en servicios son usadas sobre en todo en servicios de ejecución crítica, como por ejemplo un servidor de bases de datos, una web de una tienda online, una centralita de telefonía IP, o el Directorio Activo. Lo normal es que con el software de alta disponibilidad no sea suficiente y haya que hacer uso de herramientas o complementos adicionales para su configuración, tales como dirección IP, reglas de cortafuegos, dependencias, políticas de seguridad, copia y recuperación, etc. Entre las herramientas más importantes dentro de esta categoría en sistemas operativos GNU/Linux podemos citar Heartbeat, HA-OSCAR, KeepAlived, Red Hat Cluster Suite, The High Availability Linux Project, LifeKeeper o Veritas Cluster Server.

De una gran importancia son también las soluciones software para alta disponibilidad de datos, casi siempre requeridos por los servicios críticos de nuestro servidor. Fundamentalmente consiste en la replicación de los datos, disponiendo de ellos en diversas localizaciones, permitiendo su efectiva recuperación en caso de fallos desastrosos o cuando el nodo en el que normalmente se encuentran accesibles no se está disponible. Lo habitual es que las herramientas que ofrecen alta disponibilidad/replicación de datos trabajen de dos formas diferentes: replicación de bloque de datos y de bases de datos. En el primer tipo la información en los sistemas de ficheros es replicada bloque a bloque; algunos ejemplos son DRBD (ya analizado en mayor detalle en capítulos anteriores) o rsync. En cuanto a la replicación de bases de datos la forma de trabajar consiste en la replicación de una base de datos en varias bases de datos, localizadas remotamente, llevando a cabo la replicación instrucción a instrucción. Algunas utilidades que siguen este modelo de replicación en GNU/Linux son MySQL Replication, Slony-I, o las propias de Oracle. Otros importantes proyectos de alta disponibilidad para datos son FreeNAS, NAS Lite-2, u Openfiler. Con los dos primeros por ejemplo usamos un sistema operativo actuando como NAS (Network-Attached Storage), logrando que un equipo pueda desempeñar funcionalidades de servidor NAS soportando una amplia gama de protocolos para compartición de almacenamiento en red y replicación: cifs (samba), FTP, NFS, rsync, RAID software... y otros.

También mediante hardware es posible alcanzar alta disponibilidad tanto para datos como para servicios. Este tipo de soluciones depende en gran medida del tipo de datos o servicios a los que queremos dotar de esta característica, y normalmente suelen usarse para apoyar las soluciones de tipo software que estemos aplicando, para mantener o superar el nivel de seguridad y disponibilidad de las mismas. En la mayoría de los casos entendemos como solución hardware para alta disponibilidad la replicación de recursos hardware, como memoria o almacenamiento, o la puesta en marcha de dispositivos hardware especializados. Por ejemplo, para la alta disponibilidad de servicios de telefonía IP podemos instalar un balanceador de primarios de telefonía E1/T1 (por ejemplo, Redfone), permitiendo que aunque cayera uno de los nodos del clúster de telefonía IP sea posible establecer llamadas a través de la red de telefonía tradicional. En cuanto a alta disponibilidad de datos, soluciones como NAS, RAID, SAN,... son ampliamente conocidas y usadas.

Como podemos ver, dependiendo del proyecto en el que trabajamos disponemos de una gran variedad de soluciones tanto software como hardware que además es posible usar de forma conjunta integrándolas. Las hay propietarias, libres o incluso gratuitas. En el siguiente apartado la que es en la actualidad la solución open source sobre GNU/Linux más usada en los que a alta disponibilidad de servicios críticos se refiere (como es nuestro caso, el de las centralitas de telefonía IP con Asterisk). Heartbeat no sólo es open source sino que además puede ser descargada y usada de forma gratuita, obteniendo un gran soporte por parte de su activa comunidad de desarrolladores.

2.3. Introducción al clustering de servidores

En el verano de 1994 Thomas Sterling y Don Becker, trabajando para el CESDIS (Center of Excellence in Space Data and Informarion Sciencies) bajo el patrocinio del Proyecto de las Ciencias de la Tierra y el Espacio (ESS) de la NASA, construyeron un Cluster de Computadoras que consistía en 16 procesadores DX4 conectados por una red Ethernet a 10Mbps. Ellos llamaron a su máquina Beowulf.

La máquina fue un éxito inmediato y su idea de proporcionar sistemas basados en COTS (equipos de sobremesa) para satisfacer requisitos de cómputo específicos se propagó rápidamente a través de la NASA y en las comunidades académicas y de investigación. El esfuerzo del desarrollo para esta primera máquina creció rápidamente en lo que ahora llamamos el Proyecto Beowulf.

Este Beowulf construido en la NASA en 1994 fue el primer cluster de la historia, y su finalidad era el cálculo masivo de datos. Desde entonces, la tecnología de clusters se ha desarrollado enormemente, apareciendo gran cantidad de estudios, teorías, programas y arquitecturas implantando clusters para diversos fines.

En general, podríamos decir que hay dos tipos de clusters, atendiendo a su finalidad:

- Clusters para el procesamiento masivo de datos:
  El ejemplo más claro de este tipo el el Proyecto Beowulf, del que ya hemos hablado. Este tipo de clusters, por lo general, aprovechan la posibilidad de paralelización de cierto tipo de operaciones matemáticas (en especial, el cálculo matricial) para repartir los datos entre todos los equipos del cluster y poder así operar varios grados de magnitud más rápido. Para este fin se utilizan librerías como las PVM (Parallel Virtual Machine), que facilitan la distribución de datos entre las máquinas, incluso entre máquinas con distintos sistemas operativos, arquitecturas y lenguajes de programación.

  Otro ejemplo de cluster de este tipo sería el caso de MOSIX, unos parches para el núcleo de Linux con los que se consigue poder utilizar de forma transparente toda una red de equipos como si fuera una única supercomputadora, permitiendo el migrado transparente de cara al usuario de procesos de una máquina a otra y la compartición de recursos.

- Clusters de alta disponibilidad:
  En este caso lo que se busca no es exactamente conseguir una gran potencia de cálculo si no conseguir un conjunto de máquinas que todas realicen la misma función y que, ante el fallo de una de ellas, las demás puedan asumir sus tareas de una forma transparente y rápida.

  Por supuesto, la escalabilidad también es importante, ya que siempre podremos añadir más máquinas al cluster para así conseguir más potencia, pero el objetivo prioritario no es este si no la resistencia a cualquier fallo imprevisto.

  Aquí lo que se busca con la replicación de máquinas es evitar los puntos únicos de fallo, del inglés SPF (Single Point of Failure), que serían aquellas máquinas imprescindibles para el correcto funcionamiento del servicio que queremos dar: si únicamente tenemos una instancia de cada máquina de este tipo, se convierte en un SPF y ante cualquier fallo en este equipo, todo el cluster queda inutilizado. La teoría sobre este tipo de clusters gira en torno a estos SPF y cómo evitarlos, mediante redundancia hardware y el software apropiado para controlar el correcto funcionamiento de todos los equipos y, en caso negativo, hacer que una máquina de respaldo suplante a la que acaba de fallar.

La técnica que vamos a explorar e el TFC para obtener alta disponibilidad en nuestros servicios será la replicación de servidores a tantos niveles como nos sea posible. Por lo tanto, el tipo de clusters que nos interesa es el segundo de los expuestos.

Los clusters de alta disponibilidad necesitan de un amplio abanico de componentes que consideren diversos factores, entre otros:

- Control de miembros del cluster.
- Servicios de comunicaciones.
- Control y gestión del clustering, flujo de datos.

- Gestión y monitorización de recursos.
- Compartición o replicación del almacenamiento:
  - Compartición:
    - Discos SCSI externos.
    - Sistemas NAS.
    - Sistemas de ficheros compartidos (NFS, SMB, Coda).
  - Replicación:
    - Propio de la aplicación (DNS, NIS, etc.)
    - ftp, rsync, etc.

Todos estos detalles habrá que tenerlos en cuenta a la hora de diseñar el cluster y elegir el software que lo gestionará, ya que este software debe ser capaz por sí mismo de atender todos estos puntos de atención y reaccionar a tiempo ante un fallo en cualquiera de ellos.

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

2.4.1. RAID

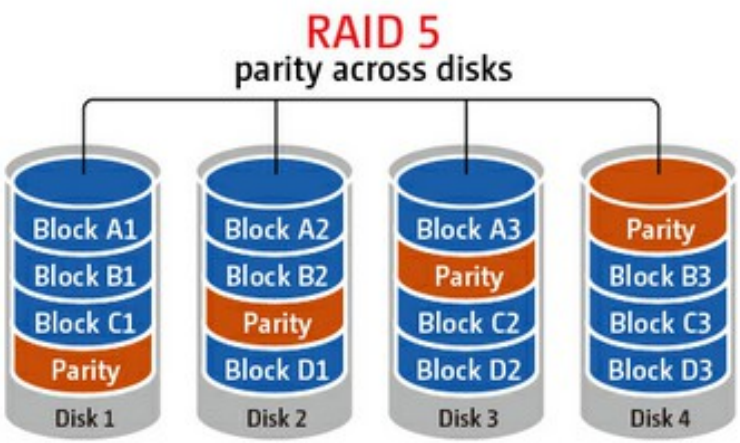
RAID (Redundant Array of Inexpensive Disks), como su propio nombre indica, consiste en crear un array (cadena) de varios discos simples (“inexpensive”, baratos), y tratarlos como un todo a la hora de acceder a ellos. El standard RAID consta de varios niveles, y en cada uno de ellos el acceso a los discos 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.

Los niveles RAID más comúnmente usados son:
- RAID 0: Conjunto dividido
- RAID 1: Conjunto en espejo
- RAID 5: Conjunto dividido con paridad distribuida

Hablemos del uno de ellos, por ejemplo del RAID5, para entender mejor el concepto.

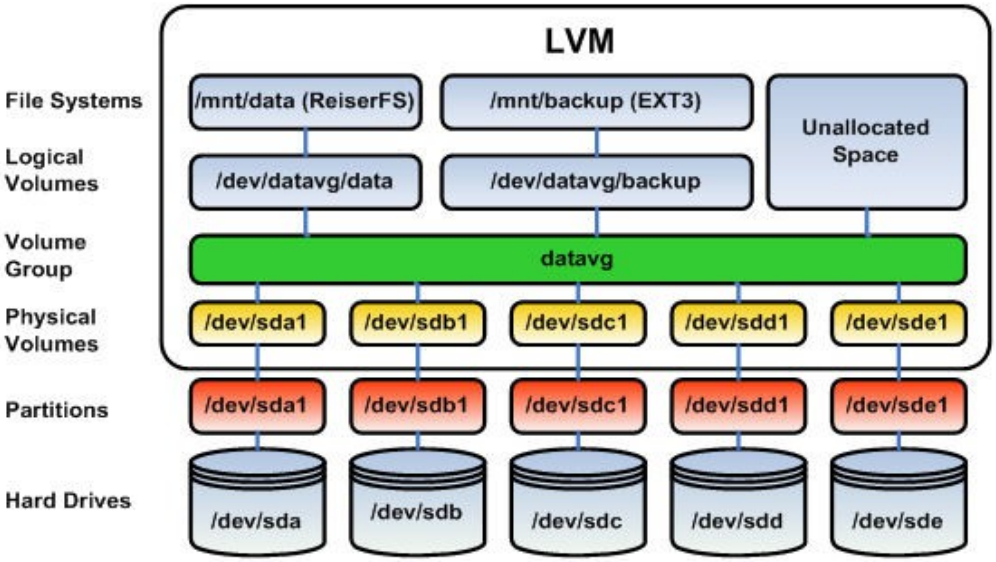
El RAID5 se puede montar sobre tres o más discos, con o sin discos inactivos adicionales. Similar a RAID4, pero la información de paridad se distribuye entre todos los discos, eliminando así el problema del cuello de botella con el disco de paridad. Si falla un disco, la información no se pierde gracias a la paridad, y el contenido del disco dañado se reconstruye en un disco inactivo.

Si fallan dos discos de forma simultánea, o si nos quedamos sin discos inactivos, la información se pierde. Tanto la velocidad de lectura como la de escritura aumentan, al realizarse en paralelo.



2.4.2. LVM

LVM (Logical Volume Manager) es un subsistema para la gestión avanzada de unidades de almacenamiento "en caliente", que se ha convertido en un estándar "de-facto" en varias implementaciones de UNIX. Inicialmente desarrollado por IBM, posteriormente adoptado por la OSF (Open Standards Foundation, ahora OpenGroup) para su sistema operativo OSF/1, que fue la base de las implementaciones de HP-UX, SUSE y Digital UNIX. Otra implementación de LVM es la desarrollada por Veritas, que está disponible para gran cantidad de sistemas UNIX pero funciona de forma distinta al resto. La versión de Linux, desarrollada por la empresa Sistina Software y liberada bajo la licencia GPL, es muy similar a la de HP✗ UX. Logical Volume Manager añade una capa software adicional entre los dispositivos físicos y el interfaz de entrada/salida de bloques del kernel, de forma similar a como lo hace el RAID por software, pero con un objetivo distinto.

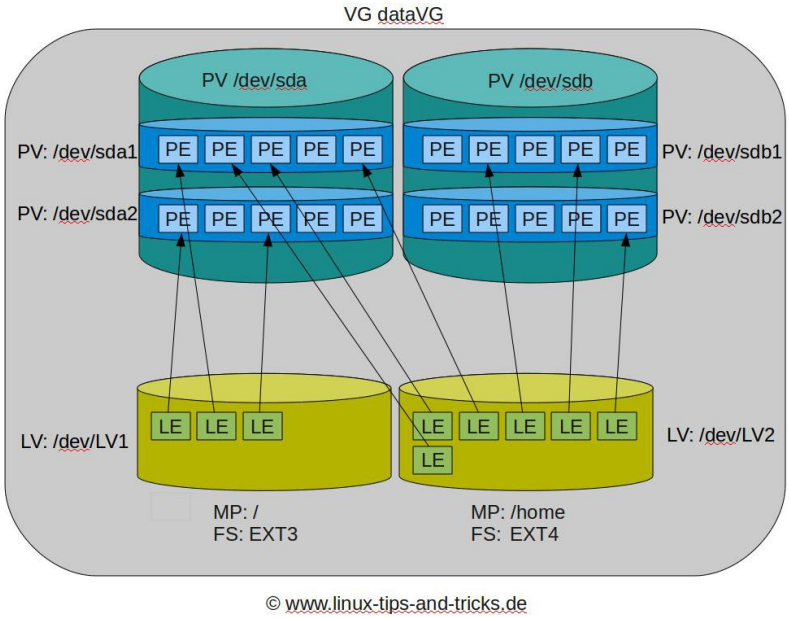


LVM es una nueva forma de asignar espacio de disco a los sistemas de ficheros: en lugar de utilizar particiones de tamaño fijo, se utilizan particiones "virtuales", que podrán crecer o disminuir según nuestras necesidades administrativas.

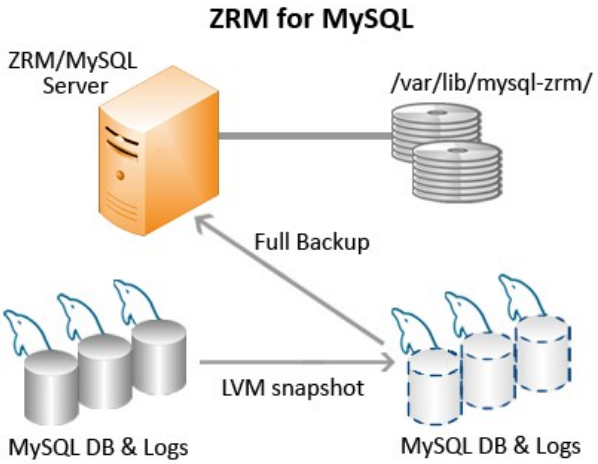
Además, el espacio asignado a una partición no tiene por qué pertenecer todo al mismo disco, con lo que se rompe la barrera de únicamente poder tener particiones como mucho del tamaño del mayor de los discos que tengamos instalados (si bien esto también era posible mediante RAID).

LVM nos ofrece una forma más potente y flexible de asignar en particiones el espacio físico de los discos duros. En lugar de dividir cada disco de forma individual en una o más particiones, como se haría con fdisk, con las

habituales desventajas de no poder tener particiones que ocupen más de un disco (salvo con RAID) y no poder variar el tamaño de las particiones una vez creadas, con lvm agrupamos volúmenes físicos (PV, de Physical Volumes), que pueden ser cualquier dispositivo de bloques (particiones, discos completos o dispositivos /dev/md? del RAID por software) en grupos de volúmenes (VG, Volume Groups). Un VG consiste de uno o más PV, y se dividen en particiones virtuales al estilo de las tradicionales, denominadas volúmenes lógicos (LV, Logical Volumes). Lo novedoso de esta tecnología es que, una vez configurados todos los volúmenes físicos y lógicos, podemos añadir o quitar en cualquier momento y en caliente (si el hardware y software lo permite) más volúmenes físicos a un grupo virtual, o más espacio a un volumen lógico. De esta forma, se elimina de un plumazo el típico problema de tener que parar y reinstalar un sistema porque una partición se ha quedado pequeña y no se puede ampliar.



Otra característica muy interesante de LVM es la posibilidad de crear "snapshots" (fotos) del sistema en un momento dado, algo muy útil a la hora de hacer una copia de seguridad. LVM necesita un LV para almacenar los datos del snapshot, que se recomienda que tenga sobre el 10% del tamaño del LV original que se quiere replicar. Cuando se le dice a LVM que "monte" el snapshot, crea un nuevo sistema de ficheros virtual, que será siempre una copia de sólo lectura del sistema original en el momento en que se creó el snapshot, y va utilizando el espacio que se le ha asignado para almacenar los cambios que se realicen sobre el sistema real. De esta forma, podemos seguir trabajando con el sistema normalmente, y disponemos de una imagen estable del sistema en un momento dado, de la que podemos hacer tranquilamente la copia de seguridad.



2.4.3. Los i-nodos

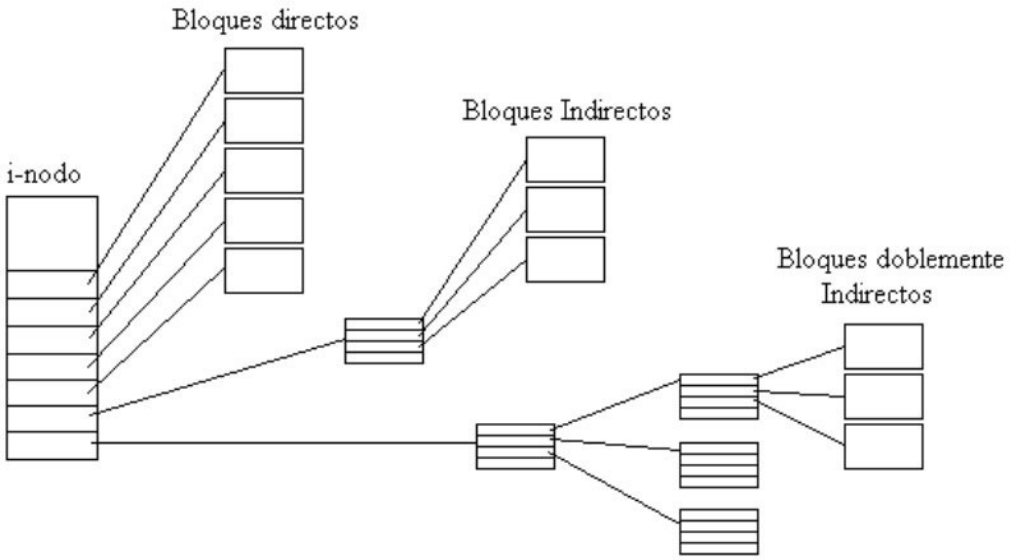
Desde el sistema de ficheros ext2, el i-nodo es el bloque de construcción básico; cada fichero y directorio del sistema de ficheros es descrito por un y sólo un i-nodo. Los i-nodos ext2 para cada Grupo de Bloque se almacenan juntos en la tabla de i-nodos con un mapa de bits (bitmap) que permite al sistema seguir la pista de i-nodos reservados y libres.

La tabla de i-nodos se descompone en varias partes: cada parte está contenida en un grupo de bloques. Esto permite utilizar estrategias de asignación particulares: cuando un bloque debe asignarse, el núcleo intenta asignarlo en el mismo grupo que su i-nodo, a fin de minimizar el desplazamiento de las cabezas de lectura/escritura en la lectura del archivo.

De todos los campos que componen un i-nodo (su estructura se encuentra en el código del kernel de Linux, en el fichero linux/Ext2_fs.h), el campo i_block contiene las direcciones de bloques de datos asociados al i-nodo. Esta tabla se estructura según el método clásico de Unix:

Los primeros doce elementos (valor de la constante EXT2_NDIR_BLOCKS) de la tabla contienen las direcciones de bloques de datos; La posición EXT2_IND_BLOCK contiene la dirección de un bloque que contiene a su vez la dirección de los bloques de datos siguientes; La posición EXT2_DIND_BLOCK contiene la dirección de un bloque que contiene la dirección de bloques que contienen la dirección de los bloques de datos siguientes; La posición EXT2_TIND_BLOCK contiene la dirección de un bloque que contiene la dirección de bloques que apuntan a su vez a bloques indirectos.

Este mecanismo de direccionamiento se ilustra a continuación (limitándose a dos niveles de indirección por razones de claridad):



2.4.4 ext4

ext4 constituye la siguiente etapa en la evolución del sistema de archivos (o ficheros; en inglés, filesystem) denominado extended, que indudablemente se ha convertido en uno de los más utilizados por los usuarios de Linux.
Es importante destacar que las modificaciones introducidas por ext4 han sido más numerosas y significativas en comparación a las realizadas por su antecesor sobre ext2. En otras palabras, ext4 presenta modificaciones en las estructuras internas del mismo sistema (de ficheros), como sucede en el caso de aquellas destinadas a la preservación de los datos propios de cada fichero, mientras que ext3 se caracterizó principalmente por haber introducido la funcionalidad journaling, inexistente en ext2.

En síntesis, el resultado ha sido un sistema mejor diseñado, más eficiente y confiable, y por supuesto con mayores prestaciones.

Las principales mejoras son con respecto a sus antecesores ext2 y ext3 son:

•Soporte de volúmenes de hasta 1024 PiB.
•Soporte añadido de extent.
•Menor uso del CPU.
•Mejoras en la velocidad de lectura y escritura.

2.4.5. ReiserFS

ReiserFS, al igual que JFS y XFS que estudiaremos a continuación, es un sistema de ficheros transaccional que nos asegura que mantiene su contenido en un estado consistente ante una caída del sistema (cuelgue, fallo del suministro eléctrico, etc) en cuestión de segundos y sin necesidad de realizar un fsck. ReiserFS también tiene otras características que lo hacen muy aconsejable en el terreno de los servidores. Antes de pasar a comentar ReiserFS en más profundidad, vamos a estudiar en qué consiste el "journalling"(mecanismo de seguridad de los sistemas transaccionales).

2.4.5.1. Sistemas transaccionales

Cualquier sistema de ficheros permite almacenar, recuperar y manipular datos, almacenados en ficheros y organizados en directorios. Para conseguir esto, el sistema debe almacenar, además de los datos en sí, unas estructuras internas que mantengan la organización de los datos sobre el disco para tenerlos accesibles en todo momento. Estas estructuras de datos internas (como los i-nodos explicados en el punto anterior) son conocidas como "meta-datos". El diseño de estos meta-datos es lo que da su personalidad y características (rendimiento, etc) a cada sistema de ficheros.

Los meta-datos no los maneja el usuario ni el administrador del sistema: se encarga el controlador del sistema de ficheros en el núcleo de Linux, que se programa para tratar con especial cuidado todo este enjambre de datos y punteros a datos. Pero para que el sistema funcione correctamente se necesita una cosa: que los meta-datos estén en un estado consistente. Si en algun momento tenemos problemas la solución clásica es utilizar el fsck, un programa que comprueba el estado de los meta-datos de nuestros sistemas de ficheros y los repara si encuentra algún error.

2.4.5.2. Características de ReiserFS

Sistemas como ext2 o ufs son lentos y desperdician espacio con los ficheros pequeños y con directorios muy llenos, llegando a desaconsejar al usuario utilizarlos e incluso a decir que no es una práctica aconsejable. Por esto, en muchos casos en los que una base de datos no tendría por qué ser necesaria, se pasa a utilizar una en lugar de tener los ficheros directamente en el disco. Esto es lo que tratan de evitar desde el equipo de desarrollo de ReiserFS. Y lo han conseguido. ReiserFS es entre ocho y quince veces más rápido que ext2 al tratar con ficheros de menos de 1k, sin por ello penalizar otros tipos de ficheros (Reiser no es más lento que ext2 con ficheros grandes, en general es siempre algo más rápido).

Otras características interesantes de ReiserFS son:

- Soporta todos los atributos para ficheros UNIX vistos en ext2 (es compatible de cara al sistema y el usuario).
- La organización interna en árboles B* proporciona un rendimiento general superior al obtenido con otros sistemas de ficheros, para ficheros y directorios de cualquier tamaño, en especial con directorios con muchos ficheros que otros sistemas no tratan bien.
- Aprovecha mejor el dispositivo, ya que por un lado no dedica una cantidad fija de sectores para las tablas de i-nodos (se ahorra un 6% de espacio), y por otro se pueden agrupar en un mismo sector ficheros pequeños que no ocupen un sector por sí mismos y "colas" de ficheros (el último trozo que tampoco ocupa todo un sector), en lugar de utilizar un sector por cada fichero pequeño.

- En el futuro se va a abrir el sistema de ficheros a una arquitectura de “plug-ins”, mediante los cuales el usuario podrá extender fácilmente el sistema con nuevos tipos de ficheros, directorios o atributos.
- Se han implantado las bases para adaptar al mundo de los sistemas de ficheros algunas técnicas hasta ahora únicas de las bases de datos.

ReiserFS se organiza internamente en Árboles B*, en lugar de en i-nodos como ext2 o tablas como la FAT de MS-DOS. Los árboles B* son un tipo de árboles balanceados (de ahí la 'B' del nombre), esto es, se van reorganizando para mantener siempre una altura mínima y así garantizar que las búsquedas sobre ellos tendrán siempre unos tiempos medios buenos, sin casos peores muy alejados de esta media.

2.4.6. xfs y jfs

xfs y jfs son dos ejemplos de tecnologías de nuevo desarrollo o bien adaptadas de otro sistema UNIX a Linux y posteriormente donadas a la comunidad del software libre. También son ejemplos del interés que el software libre está generando en el mundo de las grandes compañías informáticas, ya que estos sistemas han sido desarrollados por SGI e IBM, respectivamente.

Ambos son sistemas completamente compatibles UNIX (atributos, etc) y transaccionales, con características en general muy similares a ResierFS, si bien superiores en ciertos aspectos (soporte de ACLs, etc).

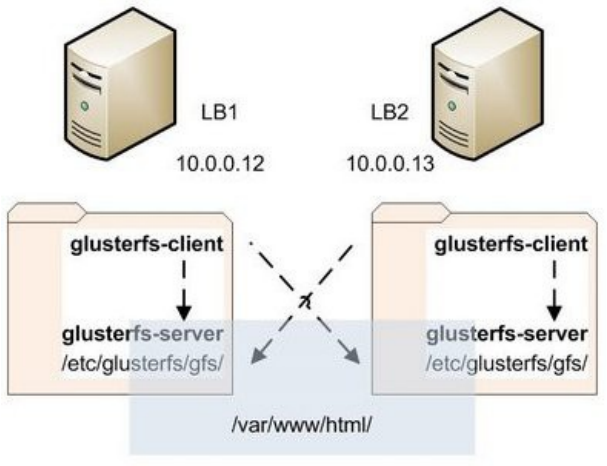
JFS fue diseñado con la idea de conseguir "servidores de alto rendimiento y servidores de archivos de altas prestaciones, asociados a e-business".

XFS es el más antiguo de los sistema de archivos con journaling disponible para la plataforma UNIX, tiene un código maduro, estable y bien depurado.

2.4.7. Gluster File System (GFS)

El Sistema de Archivos Gluster, Gluster File System o GlusterFS, es un multiescalable sistema de archivos para NAS desarrollado inicialmente por Gluster Inc. Este permite agregar varios servidores de archivos sobre Ethernet o interconexine Infiniband RDMA en un gran entorno de archivos de red en paralelo. El diseño del GlusterFS se basa en la utilización del espacio de usuario y de esta manera no compromete el rendimiento. Se pueden encontrar siendo utilizado en una gran variedad de entornos y aplicaciones como computación en nube, ciencias biomédicas y almacenamiento de archivos. El GlusterFS está licenciado bajo la licencia GNU versión 3.

GlusterFS es un sistema de archivos que nos permite replicar y/o distribuir archivos a través de la red, es una forma útil de crear una NAS (Network-Attached Storage) con muy pocos recursos.



GlusterFS está diseñado para reducir la complejidad de un storage replicado y/o distribuido por la red, pero sin reducir el rendimiento y poder utilizar recursos sencillos, como equipos x86.

2.5. 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 el cluster se comporte como un único ordenador, en el que ellos copian 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 punto estudiaremos.

2.5.1. 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.

2.5.2. NFS

El sistema de ficheros compartidos por excelencia del mundo UNIX. Se podría decir que el Network File System de SUN es el pionero de los sistemas de ficheros compartidos tal y como los conocemos hoy en día. NFS permite compartir datos entre varios ordenadores de una forma sencilla. Por ejemplo, un usuario validado en una red no necesitará hacer login a un ordenador específico: vía NFS, accederá a su directorio personal (que llamaremos exportado) en la máquina en la que esté trabajando.

Pero NFS no es un protocolo demasiado eficiente y es muy lento para conexiones mediante módem. Está diseñado para redes locales, siendo muy flexible. Ofrece muchas posibilidades tanto a usuarios como a administradores.

2.5.3. 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. Hay dos cosas que podemos hacer con Samba:

1. Compartir una unidad de Linux con máquinas Windows o unidad de Windows con máquinas Linux.
2. Compartir una impresora de Linux con máquinas Windows o una impresora de Windows con Linux.

2.5.4. 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. EL CONCEPTO DE ALTA DISPONIBILIDAD (HA)

3.1. Introducción

En el siguiente capítulo se va a realizar una descripción del concepto de alta disponibilidad para facilitar la comprensión de los siguientes apartados del proyecto en los que se diseñará e implementará una solución de cluster de alta disponibilidad para una apliación empresarial.

Se comenzará definiendo el concepto de alta disponibilidad y el motivo por el que es una herramienta imprescindible para multitud de organizaciones.

A continuación, se estudiará la importancia de la determinación de las necesidades de disponibilidad reales del cliente para abordar un proyecto de alta disponibilidad lo más coherente posible.

En el siguiente apartado se describirán los distintos niveles de disponibilidad empleando un punto de vista general y un punto de vista cercano al usuario.

El último apartado del presente se describen aspectos importantes para realizar una correcta planificación para asegurar un nivel determinado de disponibilidad en un sistema.

3.2. Alta Disponibilidad en las organizaciones

Un sistema se encuentra disponible si los usuarios pueden realizar operaciones sobre éste (acceder al sistema, someter nuevos trabajos, actualizar o modificar trabajos existentes,…).

En la actualidad los departamentos de tecnología de la información de las organizaciones han ganado mayor protagonismo y operaciones de negocio críticas que antes no dependían de estos departamentos ahora lo hacen. Debido a esta tendencia, es necesario asegurar un servicio continuado y de calidad a los procesos de negocio críticos que se ejecutan en los sistemas informáticos de las organizaciones.

El concepto de alta disponibilidad comprende un diseño de sistema y la implementación de protocolos asociados que aseguren un cierto nivel de continuidad operacional durante un periodo de tiempo determinado.

La disponibilidad de un sistema suele estar medida por año y únicamente tiene en cuenta los cortes de servicio no planificados. No se contemplan las paradas planificadas de los sistemas para el cálculo de la disponibilidad. En niveles muy altos de disponibilidad se pueden introducir estos dos parámetros (paradas planificadas y no planificadas) para el cálculo del valor.

3.3. Determinación de las necesidades de disponibilidad del cliente

El principal aspecto en el diseño de la disponibilidad de un sistema es obtener los requerimientos reales de disponibilidad de los usuarios. Para obtener esta información es necesario estudiar con detenimiento el modo en el que los usuarios utilizan las aplicaciones de la organización, determinar cuales son críticas y necesitan de una mayor disponibilidad.

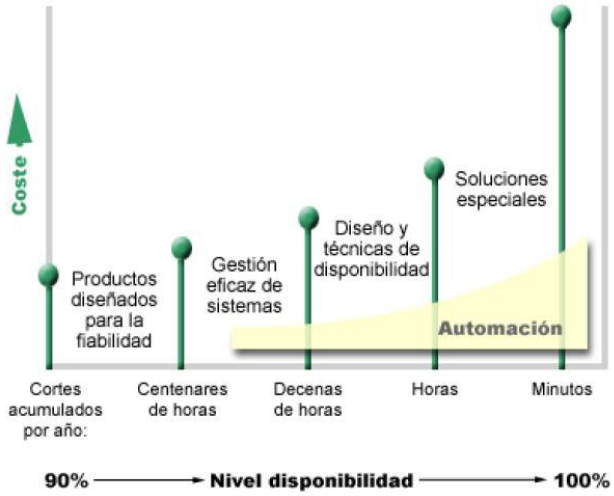
En este primer estudio del entorno es recomendable que el cliente sea consciente del coste de hacer que un sistema junto con sus aplicaciones sea altamente disponible, ya que como norma general, el coste de un sistema se incrementa exponencialmente conforme aumenta su disponibilidad.

Tras realizar el estudio de las necesidades de disponibilidad del cliente, se puede redactar un documento de acuerdo de nivel de servicio. En esta primera fase de estudio el documento únicamente debe reflejar los aspectos relativos a la disponibilidad de los sistemas. Conforme se va avanzando en el diseño de la solución, el documento de acuerdo de nivel de servicio reflejará otros aspectos como el tiempo de respuesta tras un fallo en el

sistema, las paradas planificadas necesarias para el correcto mantenimiento del sistema y otros aspectos relativos al rendimiento o calidad del servicio.

En la mayoría de los casos, puede obtenerse un alto nivel de disponibilidad implementando los procesos y los métodos de gestión de sistemas adecuados. Cuanta más disponibilidad continuada sea necesaria, mayor será la inversión a realizar. Antes de realizar una inversión de este tipo, es necesario asegurarse que ese nivel de disponibilidad es realmente necesario. En la figura siguiente se muestra cómo distintas técnicas pueden mejorar la disponibilidad, aunque también pueden aumentar el precio que se deberá pagar.

En la siguiente figura vemos el coste de diferentes técnicas para mejorar la disponibilidad:



La siguiente tabla muestra los tiempos máximos de caída permitidos anualmente para alcanzar un porcentaje determinado de disponibilidad.

Disponibilidad Necesaria	*Tiempo de caída (min/año)*	*Tiempo de caída (horas/año)*
99.00%	5,256	87.6
99.25%	3,942	65.7
99.50%	2,628	43.8
99.75%	1,314	21.9
99.90%	526	8.76
99.99%	53	0.88
99.999%	5	0.08

3.4. Niveles de disponibilidad

Existen varios niveles de disponibilidad. Estos niveles se diferencian por el tipo y la duración de las interrupciones que admiten. Estos niveles son los siguientes:

- Altamente disponible. El servidor ofrece un nivel aceptable o acordado de servicio durante su período de funcionamiento programado. El objetivo es que el servidor esté disponible cuando el cliente lo necesite.
- Alta disponibilidad. El servidor ofrece un nivel aceptable o acordado de servicio durante su período de funcionamiento programado. El objetivo es que no se produzca ninguna interrupción no planificada; podrían producirse algunas interrupciones planificadas.
- Operaciones continuadas. El servidor ofrece un nivel aceptable o acordado de servicio 24 horas al día, 365 días al año. El objetivo es que el servidor funcione sin interrupciones planificadas; podrían

producirse algunas interrupciones no planificadas.

- Disponibilidad continuada. El servidor ofrece un nivel aceptable o acordado de servicio 24 horas al día, 365 días al año. El objetivo es que no se produzca ninguna interrupción planificada ni no planificada.

En el diagrama siguiente se muestra cómo se relacionan entre sí estos distintos niveles de disponibilidad y qué tipos de empresas son adecuados para cada nivel.

3.5. Causas posibles del tiempo de inactividad o interrupción

En los escenarios más críticos de disponibilidad no se puede permitir que el mantenimiento planificado y los tiempos de parada no planificados interrumpan la disponibilidad del sistema en cualquier momento. A continuación se examinan cuales son esas posibles causas, tanto imprevistas como previstas que se deben tener en cuenta al diseñar una solución altamente disponible.

3.5.1. Interrupciones no planificadas

En cuanto a fallos imprevistos podemos considerar fallos de software, fallos de hardware, errores humanos, y desastres naturales.

- Los fallos del software incluyen sistema operativo, base de datos, middleware, uso y fallos de la red. Un fallo de estos componentes puede causar un fallo del sistema.

- Los fallos del hardware incluyen errores de sistema (CPU, memoria, suministro eléctrico, bus), periféricos (disco, cinta, controladoras), fallos en la red, y apagones.

- Errores humanos, causa principal de fallos, incluyen errores de operador, de usuario, de administrador de la base de datos, o de administrador del sistema. Otro tipo de errores humanos que puede causar el tiempo de inactividad no planificado son los de sabotaje.

- La categoría final es desastres naturales. Aunque infrecuentes, estas causas del tiempo de inactividad pueden tener impactos extremos en las empresas, debido a su efecto prolongado sobre operaciones. Las causas posibles de desastres incluyen fuegos, inundaciones, terremotos, apagones, y bombardeos.

3.5.2. Interrupciones planificadas

Las interrupciones planificadas son necesarias y se cuenta con ellas; sin embargo, el hecho de que sean planificadas no significa que no sean disruptivas. Las interrupciones planificadas suelen estar relacionadas con el mantenimiento del sistema. En estos casos es importante diseñar un sistema para reducir al mínimo las interrupciones. Las causas previstas de tiempo de inactividad incluyen operaciones rutinarias, mantenimiento

periódico y nuevos despliegues.

- Operaciones rutinarias como la instalación de parches o reconfigurar el sistema son ocasionalmente necesarias para actualizar la base de datos, aplicaciones, SO, middleware o la red.

- Mantenimiento periódico de la Base de Datos (del sistema de almacenamiento, parámetros de inicialización, parches de software), de las aplicaciones (administración del esquema, parches de software), del SO, Middleware, de la red.

- Despliegues nuevos significan actualizaciones hardware, SO, base de datos, Middleware, de aplicaciones y de red.

También es importante considerar no sólo el tiempo utilizado en realizar la actualización sino el efecto que el cambio repercute en la aplicación en general.

3.6. ¿Cómo se puede implementar alta disponibilidad?

A continuación se señalan algunas de las técnicas utilizadas para conseguir un sistema con la más alta disponibilidad. La principal técnica para obtener estos sistemas se centra en la redundancia y en replicar las zonas críticas, consiguiendo 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.

3.6.1. Protección contra anomalías en el disco

El almacenamiento en disco es el almacenamiento interno del servidor o un almacenamiento conectado al mismo. El servidor considera que este espacio en disco, junto con la memoria principal del servidor, es un área de almacenamiento de gran tamaño.

Cuando guarda un archivo, no lo asigna a una ubicación de almacenamiento; en lugar de ello, el servidor coloca el archivo en la ubicación que garantiza el mejor rendimiento. Es posible que distribuya los datos del archivo entre varias unidades de disco si ello constituye la mejor opción. Cuando se añaden más registros al archivo, el sistema asigna espacio adicional en una o más unidades de disco. Esta forma de direccionar el almacenamiento se denomina almacenamiento de un solo nivel.

Puesto que los datos se distribuyen entre los discos, es importante considerar cómo pueden protegerse los datos en caso de que se produzca una anomalía en uno de esos discos.

La protección de dispositivo por paridad permite que el servidor pueda seguir funcionando cuando un disco presenta una anomalía o se daña. Cuando se utiliza la protección de dispositivo por paridad, el adaptador de entrada/salida (IOA) del disco calcula y guarda un valor de paridad para cada bit de datos. El IOA calcula el valor de paridad de los datos en la misma ubicación de cada una de las demás unidades de disco del conjunto de paridad de dispositivo. Cuando se produce una anomalía en el disco, los datos pueden reconstruirse utilizando el valor de paridad y los valores de los bits de las mismas ubicaciones de los otros discos. Mientras tiene lugar la reconstrucción de los datos, el servidor sigue en ejecución.

La protección por duplicación es una forma de proteger los datos en caso de que se produzca una anomalía en el disco. Los datos quedan protegidos porque el sistema mantiene dos copias de los datos en dos unidades de disco distintas. Cuando se produce una anomalía en un componente relacionado con un disco, el sistema puede seguir funcionando sin interrupciones utilizando una copia duplicada de los datos hasta que se repara el componente anómalo.

Son posibles distintos niveles de protección por duplicación, en función del hardware que se haya duplicado. Se puede duplicar lo siguiente:

- Las unidades de disco.

- Los controladores de disco.
- La unidad de bus de E/S.
- Los procesadores de E/S de disco.
- Un bus.

Las agrupaciones de discos independientes (también denominadas agrupaciones de almacenamiento auxiliar independientes) permiten evitar que se produzcan interrupciones no planificadas porque los datos de éstas quedan aislados del resto del servidor. Si una agrupación de discos independiente presenta una anomalía, el servidor puede seguir funcionando.

3.6.2. Planificación para una pérdida de alimentación

Para garantizar que el servidor estará disponible cuando se necesite, es necesario asegurar de que dispone de una fuente de alimentación adecuada y de que está protegido en caso de que se produzca una pérdida de la alimentación.

Una parte del proceso de planificación para el servidor consiste en garantizar que dispone de una fuente de alimentación adecuada. Es necesario conocer cuáles son los requisitos de alimentación del servidor y solicitar la ayuda de un electricista cualificado para instalar los cables correctos.

Algunos servidores (gama alta y mainframes) cuentan con baterías de reserva. La unidad de batería de reserva proporciona un tiempo extra de ejecución. Si la alimentación no se restablece transcurrido ese tiempo, el sistema entra inmediatamente en un estado de conclusión controlada.

Algunos servidores disponen de fuentes de alimentación redundante. Una fuente de alimentación redundante es una característica que evita que se produzca una interrupción no planificada proporcionando alimentación en caso de que una fuente de alimentación deje de suministrarla.

Aun con servidores con fuente de alimentación adecuada y redundante, es posible que se produzca un corte en la alimentación, por ejemplo, durante una tormenta. Para evitar las interrupciones no planificadas que tienen lugar como consecuencia de una pérdida de la alimentación, puede que sea necesario realizar una inversión en hardware específicamente diseñado para mantener el servidor en funcionamiento cuando se pierde la alimentación.

Un dispositivo de hardware de este tipo sería una fuente de alimentación ininterrumpida (UPS). Es posible utilizar una UPS para suministrar alimentación auxiliar al procesador, los discos, la consola del sistema y cualquier otro dispositivo que la necesita. Las fuentes de alimentación ininterrumpida proporcionan las ventajas siguientes:

- Permiten continuar con las operaciones cuando se han producido interrupciones en la alimentación de corta duración.
- Protegen el servidor en caso de producirse picos de voltaje.
- Proporcionan una finalización normal de las operaciones, lo que puede reducir el tiempo de recuperación al reiniciar el servidor.

Si existe la posibilidad de experimentar una anomalía general en la alimentación, es importante considerar la posibilidad de instalar un generador de alimentación. Un generador ofrece más ventajas que una UPS porque permite seguir realizando operaciones normales cuando se producen cortes en la alimentación de mayor duración.

3.6.3. Uso de métodos eficaces de gestión de sistemas

Una de las formas más sencillas de evitar las interrupciones no planificadas es asegurarse de realizar todo lo necesario para garantizar la correcta ejecución del servidor. Esto incluye la realización de tareas básicas de mantenimiento preventivo y de gestión de sistemas que ayuden a maximizar el rendimiento del sistema. La mayoría de estas tareas de gestión de sistemas pueden automatizarse, lo que ayuda a evitar que se produzcan anomalías que pueden ser el resultado de un error humano o de un descuido.

Una forma de garantizar la disponibilidad del servidor es mediante al supervisión de su rendimiento y reaccionar con prontitud ante cualquier problema que pueda detectarse. Es posible emplear herramientas de monitorización para realizar el seguimiento del rendimiento del servidor activamente. Estas herramientas suelen permitir su configuración para enviar notificaciones acerca de cualquier problema que comprometa la disponibilidad del servidor, a tiempo para reaccionar y evitar una interrupción no planificada.

Las actualizaciones de seguridad del sistema y aplicaciones también son un importante componente de la gestión de sistemas que pueden ayudar a mantener la disponibilidad del sistema. Cuando se detectan problemas en los sistemas operativos y aplicaciones, el fabricante suele facilitar una actualización para corregir el problema. Es necesario estar informado acerca de las actualizaciones e instalarlas en el servidor para asegurar que su funcionamiento esté en su nivel óptimo. Es muy recomendable la creación de una estrategia de gestión de actualizaciones y establecer que el proceso de verificación y aplicación de nuevas actualizaciones forme parte del mantenimiento habitual del servidor.

3.6.4. Preparación del espacio para el servidor

Una forma de evitar que se produzcan interrupciones no planificadas es asegurar de que el espacio en el que se coloca el servidor favorezca la disponibilidad. En el rendimiento del servidor participan numerosos factores físicos y del entorno.

En primer lugar, es necesario familiarizarse con el servidor. Los distintos modelos de servidor tienen requisitos distintos en relación con las condiciones en las que deben trabajar, por lo tanto, es imprescindible conocer las necesidades de operación del servidor.

Tras conocer las características físicas del servidor, es necesario tener en cuenta lo siguiente acerca del espacio en el que debe residir el servidor:

- Ubicación. La ubicación física del servidor puede influir en su disponibilidad. Por ejemplo, si la sala no es segura, el servidor podría quedar expuesto a cualquier agresión o, incluso, podría desenchufarse accidentalmente el cable de alimentación.
- Cables. Con frecuencia, no se da mucha importancia a los cables, pero, sin ellos, no podríamos utilizar el servidor. Es importante asegurar que los cables están en perfectas condiciones y de que su utilización es la correcta.
- Entorno. El entorno en el que se instala el servidor también es muy importante para la disponibilidad. El entorno incluye, por ejemplo, la temperatura, la humedad y otros factores que pueden impedir que el rendimiento del servidor sea correcto.

3.6.5. Backups

Los backups son la mejor solución y también la más económica en caso de un desastre natural que pudiera significar horas o días de tiempo de caída. Pueden ser de tres tipos:

· En caliente.
· En frío.
· Incrementales.

Una mención especial merecen los backup online o backups en caliente. Se realizan mientras las transacciones se producen contra la base de datos y permiten que ésta permanezca disponible para la actividad normal mientras se realiza. Los sistemas de ficheros de los sistemas operativos modernos también ofrecen la posibilidad de realizar backups en caliente sobre los ficheros.

3.6.6. Software de replicación remota

Se trata de una solución que funciona incluso en caso de caída completa del Centro de Proceso de Datos principal.

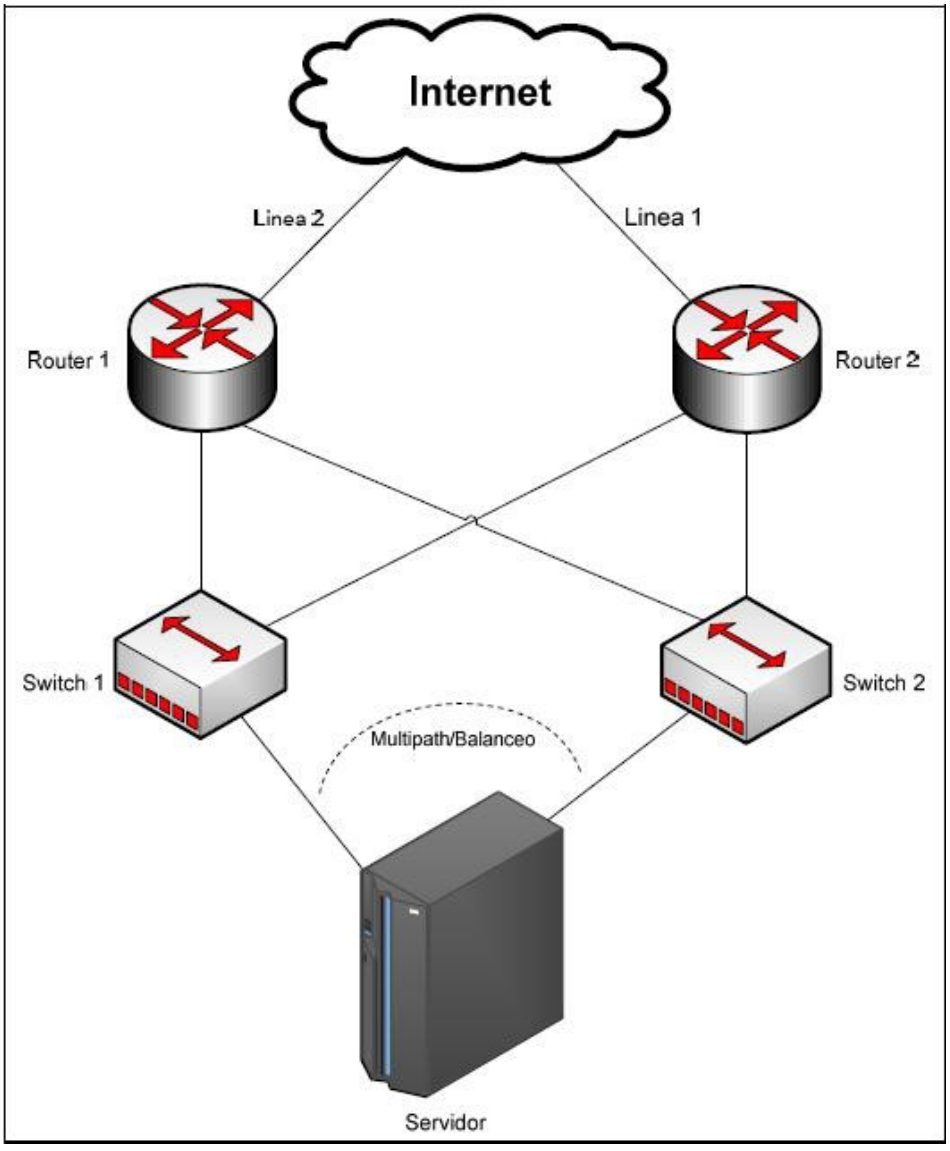
Mientras se está trabajando, los datos son copiados de forma I/O-síncrona en una localización remota, de modo independiente para el sistema y sin alterar el funcionamiento normal del sistema. Si falla el sistema primario, es posible realizar un cambio rápido sobre el sistema secundario. Después de algunos minutos todas las funciones que se ejecutaban en el sistema primario estarán nuevamente disponibles.

Esta solución requiere hardware específico de almacenamiento, software y ancho de banda de la red (la replicación síncrona requiere mayor ancho de banda que la replicación asíncrona, ya que no se puede modificar un nuevo dato hasta que no se ha almacenado en el otro sistema).

3.6.7. Caminos de red redundantes o Multipath / Path Failover

Una solución de comunicación empleando múltiples caminos garantiza que exista un camino de comunicación cuando se produce la caída del camino principal. Además de la seguridad que introduce el empleo de múltiples caminos de comunicación, existe la posibilidad de configurar el sistema para realizar balanceo de carga. Cuando se produce una acumulación de datos por sobrecarga simplemente se desvía por otro camino con menos actividad, la carga se distribuye de forma automática a través del resto de caminos redundantes de la forma más óptima. Esto puede significar un aumento importante del rendimiento de los sistemas y, adicionalmente, una ganancia en disponibilidad.

Un ejemplo de esquema de multipath sencillo seria el siguiente:



3.6.8. Clusteres y programas altamente disponibles

La principal estrategia de disponibilidad para un entorno de varios sistemas es la utilización de clusteres. Un cluster es una colección o un grupo de varios servidores que funcionan conjuntamente como si se tratara de un único servidor. Si una organización necesita alta disponibilidad o una disponibilidad continuada, se debería considerar la posibilidad de utilizar clusteres.

Los servidores de un cluster funcionan en colaboración para proporcionar una única solución de sistemas. Un cluster puede estar compuesto de un número muy elevado de servidores. Esto ofrece la posibilidad de agrupar con eficacia servidores con el fin de establecer un entorno que proporcionará una disponibilidad cercana al 100 por ciento para las operaciones y los datos más importantes. Una configuración en cluster ayuda a garantizar que los servidores y las aplicaciones más importantes están siempre disponibles. Los clusteres también ofrecen una gestión de sistemas simplificada y mayor escalabilidad para poder incorporar nuevos componentes progresivamente a medida que crece la organización.

Aunque las ventajas de los clusteres son numerosas, su coste es significativo. Es necesario comparar el coste de esta solución con el coste del tiempo de inactividad del servidor para determinar si se debe implementar o no la utilización de clusteres en una organización.

Si se opta por la utilización de clusteres en un entorno, es importante considerar los tipos de aplicaciones que se utilizan en la organización. Existen algunas aplicaciones diseñadas para hacer frente a algunos de los efectos de una anomalía.

Como se ha comentado en varias ocasiones, las aplicaciones y los datos son muy importantes para una organización. Si se utilizan clusteres, es recomendable utilizar programas que ofrezcan posibilidad de recuperación ante una interrupción en el sistema.

Estas aplicaciones pueden ser diseñadas y/o desarrolladas por el departamento de desarrollo de la organización, pero también pueden adquirirse aplicaciones que satisfagan los criterios necesarios para funcionar en entornos “clusterizados”. Si se opta por diseñar y/o desarrollar los programas personalmente, es necesario orientar el diseño de programas a la posibilidad de recuperación y a que cumplan los niveles de disponibilidad de la organización.

3.7. Cluster Alta disponibilidad

3.7.1. Ventajas de los clusteres

Los clusteres ofrecen una solución para las organizaciones que necesitan entornos operativos durante 24 horas diarias, siete días a la semana.

Mediante la utilización de clusteres, es posible reducir significativamente el número y la duración de interrupciones no planificadas y la duración de las interrupciones planificadas, garantizando la disponibilidad continua de los sistemas, datos y aplicaciones.

Las principales ventajas que pueden ofrecer un cluster a una organización:

- Disponibilidad continua: Los clusteres aseguran la disponibilidad continua de los sistemas, datos y aplicaciones.
- Administración simplificada: Es posible administrar un grupo de sistemas como un solo sistema o una sola base de datos, sin necesidad de conectar a cada uno de los sistemas individualmente. Se puede utilizar un dominio administrativo de cluster para gestionar con mayor facilidad los recursos que se comparten en un cluster.
- Mayor escalabilidad: Permite la adición sin fisuras de nuevos componentes según las necesidades de crecimiento de la organización.

3.7.2. Cómo funciona un cluster

La infraestructura de cluster aumenta la seguridad para los recursos críticos. Estos recursos pueden incluir datos, aplicaciones, dispositivos y otros recursos a los que acceden múltiples clientes.

Si se produce una interrupción del sistema o un siniestro en las instalaciones, puede accederse a las funciones proporcionadas en un sistema del cluster a través de otros sistemas que se han definido en el cluster.

Existen dos modelos de acceso a estos datos:

- Modelo de copia de seguridad primaria
- Modelo de iguales

3.7.3. Conceptos básicos de los clusteres

Antes de empezar a diseñar y personalizar un cluster que se ajuste a las necesidades de una organización, es necesario entender algunos conceptos básicos de los clusteres.

Existen dos conceptos básicos relacionados con los clusteres:

· Nodos de cluster.
· Grupo de recursos de cluster.

Por ejemplo, un nodo de cluster es un sistema System i o una partición lógica que sea miembro del cluster. Cuando se crea un cluster, se especifican los sistemas o particiones lógicas que se desea incluir en el cluster como nodos.

Un grupo de recursos de cluster (CRG o Cluster Resource Group) se utiliza como el objeto de control para una colección de recursos de recuperación. Un CRG puede contener un subconjunto o todos los nodos del cluster. Un cluster de System i soporta cuatro tipos de CRG: de aplicación, de datos, de dispositivos y de iguales.

En estos tipos de CRG hay dos elementos comunes:

- Dominio de recuperación.
- Programa de salida.

Un dominio de recuperación define el cometido de cada nodo en el CRG. Cuando se crea un CRG en un cluster, el objeto CRG se crea en todos los nodos especificados que deben incluirse en el dominio de recuperación. Sin embargo, se facilita una sola imagen del sistema del objeto CRG, a la que puede acceder desde cualquier nodo activo en el dominio de recuperación del CRG. Es decir, cualquier cambio que se introduzca en el CRG, se aplicará a todos los nodos del dominio de recuperación.

Se llama a un programa de salida durante los eventos del CRG relacionados con el cluster. Uno de estos eventos es pasar un punto de acceso de un nodo a otro.

Existen dos modelos de CRG que se pueden crear en un cluster: modelo de copia de seguridad primaria y modelo de iguales. En el modelo de copia de seguridad primaria, los nodos del dominio de recuperación del CRG se pueden definir como se indica a continuación:

- El nodo primario es el nodo de cluster que sirve de punto primario de acceso para un recurso de cluster.

- Un nodo de reserva es un nodo de cluster que asumirá el cometido de servir de acceso primario si el nodo primario actual sufre una anomalía, o si se inicia una conmutación manual por administración.

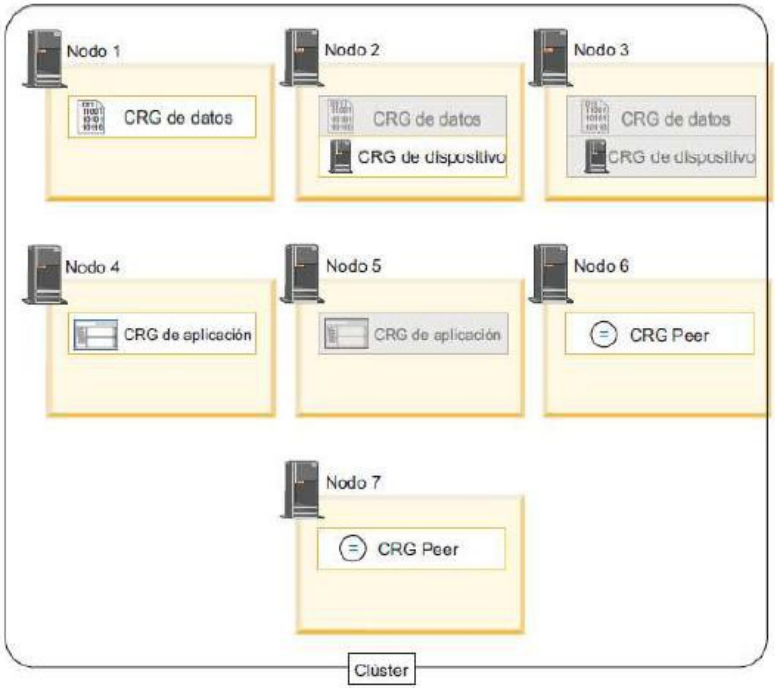
- Un nodo de duplicación es un nodo de cluster que contiene copias de los recursos de un cluster, pero que no puede asumir el cometido de nodo primario o nodo de reserva.

En un modelo de iguales, el dominio de recuperación de un CRG de iguales define una relación de igualdad entre nodos. Los nodos del dominio de recuperación del CRG de iguales se pueden definir como se indica a continuación:

- Un nodo igual es un nodo de cluster que puede ser un punto de acceso activo para recursos de cluster.
- Un nodo de duplicación es un nodo de cluster que contiene copias de los recursos de un cluster. Los nodos definidos como duplicados en un CRG de iguales representan el punto de acceso inactivo de los recursos del cluster.

En un CRG de iguales, los nodos del dominio de recuperación son equivalentes en relación con el cometido que juegan los nodos en la recuperación. Puesto que cada nodo de este CRG de iguales tiene esencialmente el mismo cometido, los conceptos de sustitución por anomalía y conmutación de traspaso no son aplicables. Los nodos tienen una relación de igualdad y cuando uno de los nodos falla, los demás nodos iguales seguirán operando.

En la siguiente figura se muestran los diferentes tipos de CRG que se pueden crear:



3.7.3.1 Tipos de CRG

1. CRG DE DATOS
   El CRG de datos está presente en el Nodo 1, el Nodo 2 y el Nodo 3. Ello significa que el dominio de recuperación del CRG de datos ha especificado un cometido para el Nodo 1 (primario), el Nodo 2 (primer nodo de reserva) y el Nodo 3 (segundo nodo de reserva). En el ejemplo, el Nodo 1 actúa como el punto primario de acceso. El Nodo 2 está definido como el primer nodo de reserva en el dominio de recuperación. Ello significa que el Nodo 2 contiene una copia del recurso que se mantiene actual mediante la duplicación lógica. Si se produjera una conmutación por anomalía o por administración, el Nodo 2 se convertiría en el punto primario de acceso.

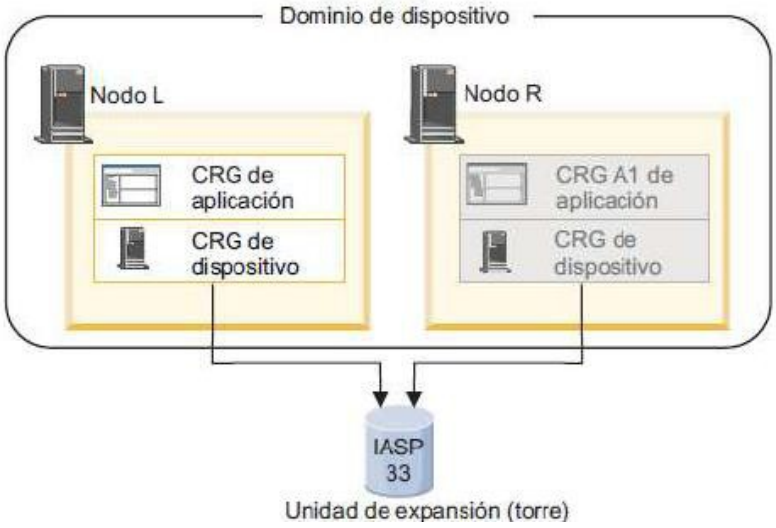
2. CRG DE APLICACIÓN
   El CRG de aplicación está presente en el Nodo 4 y el Nodo 5. Ello significa que el dominio de recuperación para el CRG de aplicación ha especificado el Nodo 4 y el Nodo 5. En el ejemplo, el Nodo 4 actúa como el punto primario de acceso. Si se produjera una conmutación por anomalía o por administración, el Nodo 5 se convertiría en el punto primario de acceso para la aplicación. Requiere una dirección IP de toma de control.

3. CRG DE IGUALES
   El CRG de iguales está presente en el Nodo 6 y el Nodo 7. Ello significa que el dominio de recuperación del CRG de iguales ha especificado el Nodo 6 y el Nodo 7. En este ejemplo, los nodos 6 y 7 pueden ser nodos iguales o duplicados. Si se trata de un dominio administrativo de cluster representado por un CRG de iguales, los recursos supervisados por el dominio administrativo de cluster tendrán sincronizados los cambios en el dominio
   representado por el nodo 6 y el nodo 7, sea cual sea el nodo en el que se ha originado el cambio.

4. CRG DE DISPOSITIVO
   El CRG de dispositivo está presente en el Nodo 2 y el Nodo 3. Ello significa que el dominio de recuperación para el CRG de dispositivo ha especificado el Nodo 2 y el Nodo 3. En el ejemplo, el Nodo 2 actúa como el punto primario de acceso. Por lo tanto, actualmente puede accederse al dispositivo del CRG de dispositivo desde el Nodo 2. Si se produjera una conmutación por anomalía o por administración, el Nodo 3 se convertirá en el punto primario de acceso para el dispositivo.

   Un CRG de dispositivo requiere la configuración de un dispositivo conocido como agrupación de discos independiente (o también agrupación de almacenamiento auxiliar independiente o ASP independiente), en un dispositivo externo, una unidad de expansión (torre) o un procesador de entrada/salida (IOP) en una partición lógica.

Los nodos del dominio de recuperación de un CRG de dispositivo también deben ser miembros del dominio del mismo dispositivo. El siguiente ejemplo ilustra un CRG de dispositivo con el Nodo L y el Nodo R en su dominio de recuperación. Ambos nodos son también miembros del mismo dominio de dispositivo.



4. PROYECTOS DE ALTA DISPONIBILIDAD

4.1. Principales Proyectos HA

Para GNU/Linux existen una gran variedad de proyectos que nos aportan las características de la alta disponibilidad, proyectos tanto para la alta disponibilidad en servicios como en datos.

Algunos de los proyectos destinados a ofrecer alta disponibilidad en servicios son:

- **HA-OSCAR**. Es un proyecto Open Source cuyo objetivo es proporcionar de manera combinada el poder de la alta disponibilidad con un gran rendimiento de cómputo. El proyecto está basado en un sistema de clustering Beowulf enfocándose por tanto a aplicaciones que requieren un grado elevado de disponibilidad. HAOSCAR tiene en cuenta los puntos débiles por lo que adopta redundancia de componentes. HA-OSCAR incorpora detección de fallos, mecanismos de recuperación, etc.

- **The High Availability Linux Project**. Proporciona una solución de alta disponibilidad ofreciendo fiabilidad, disponibilidad. Es una de las soluciones más ampliamente extendidas, estimándose que ha sido instalado en más de treinta mil instalaciones críticas desde 1999. Se integra perfectamente con multitud de servicios y sigue aún mejorando dicha integración gracias a la aportación de su activa comunidad.

- **Kimberlite**. Es considerado un proyecto único ya que es una completa infraestructura de alta disponibilidad de código abierto que incluye incluso garantía sobre la integridad de los datos. Actualmente es un proyecto abandonado o de muy baja actividad.

- **LifeKeeper.** Es una alternativa moderna y flexible comparada con los productos que ofrecen soluciones de alta disponibilidad. Dispone de mecanismos para mantener la integridad de los datos. Una numerosa colección de kits de recuperación para los distintos servicios y/o aplicaciones permiten instalar LifeKeeper en cuestión de horas. Es un producto de la empresa SteelEye.

- **HP Serviceguard**. Es un software especializado para ofrecer alta disponibilidad a las aplicaciones más críticas ante fallos tanto hardware como software. HP Serviceguard monitoriza el estado de cada nodo y responde rápidamente en caso de fallo permitiendo minimizar el tiempo durante el cual no se está ofreciendo el servicio. Está disponible tanto para Linux como para HP-UX.

- **Red Hat Cluster Suite**. Ha sido diseñado específicamente para Red Hat Enterprise Linux. Proporciona además de alta disponibilidad balanceo de carga. Dispone de soluciones listas para implantar para la mayoría de los servicios, pudiendo además proporcionar soporte cuando sea necesario.

- **Veritas Cluster Server**. Es un producto de la empresa Symantec, una solución clustering de alta disponibilidad para la industria.

- **KeepAlived**. El objetivo principal del proyecto es el de proporcionar alta disponibilidad al proyecto Linux Virtual Server. KeepAlived es un servicio que monitoriza el estado del cluster LVS para que en caso de fallo el cluster siga en funcionamiento.

  Algunos de los proyectos destinados a ofrecer **alta disponibilidad en datos** son:

- **DRBD**. Software de replicación de dispositivos de bloque formando un RAID 1 a través de la red, es decir, replica los datos en distintas localizaciones. Datos de un sistema de archivos, de una base de datos, etc. Algunas de sus características son una replicación en tiempo real de manera continua, transparencia en las aplicaciones que estén almacenando datos en la unidad, posibilidad de recuperación de los datos ante un desastre, etc. Se complementa perfectamente con Heartbeat.

- **FreNAS. FreeNAS** es un servidor NAS “Network-Attached Storage” gratuito, que soporta: protocolos Cifs (Samba), Ftp, Nfs, Rsynv, autentificación de usuario local, RAID (0,1,5) por software con un completo interfaz de configuración Web. FreeNAS ocupa menos de 32MB una vez instalado en Compact Flash, disco duro o un dispositivo de memoria usb. FreeNAS está basado en FreeBSD que es un derivado

de Unix, permitiendo clientes tanto de GNU/Linux como Windows y por supuesto Unix.

- **Openfiler**. Es un sistema operativo para compartir en red el almacenamiento de tipo SAN-NAS. Provee de buena integración con iSCSI (Internet SCSI) y "fibre channel", muy útiles en entornos empresariales y en virtualización tal como Xen y Vmware. Cuenta con una interfaz Web para su administración y soporta clientes de todos los sistemas operativos. Dispone de soporte empresarial previo pago.

- **NAS Lite-2**. Es un sistema operativo NAS que permite transformar un computador personal en un servidor de archivos Smb/Cifs, Nfs, Afp, Ftp, Http y Rsync. Su principal característica es que NAS Lite está optimizado para ofrecer el máximo rendimiento manteniendo un mínimo consumo de recursos, es decir, NAS Lite es compacto, estable y muy fiable y lo más sorprendente es que se ejecuta directamente sobre la ram necesitando únicamente 8MB de disco.

- **Mdadm**. Es una utilidad de GNU/Linux que permite crear, borrar y monitorizar un sistema RAID software. Para hacer uso de esta utilidad es necesario habilitar el soporte RAID en el núcleo.

- **MySQL Replication**. Es un módulo del servicio MySQL. Consiste en una replicación asíncrona unidireccional: un servidor actúa como maestro y uno o más actúan como esclavos. El servidor maestro escribe actualizaciones en el fichero de log binario, y mantiene un índice de los ficheros para rastrear las rotaciones de logs. Estos logs sirven como registros de actualizaciones para enviar a los servidores esclavos. Cuando un esclavo se conecta al maestro, informa al maestro de la posición hasta la que el esclavo ha leído los logs en la última actualización satisfactoria. El esclavo recibe cualquier actualización que han tenido lugar desde entonces, y se bloquea y espera para que el master le envíe nuevas actualizaciones.

- **Slony-I.** Es un sistema de replicación de "maestro a múltiples esclavos", el cuál además soporta el concepto de "cascada" (un nodo puede proveer a otro nodo, el cual puede proveer a otro) permitiendo así la recuperación ante errores. Incluye las principales características para replicar bases de datos de gran información a un número límite razonables de esclavos. Es por tanto un sistema diseñado para centros de datos y sitios de backup donde el funcionamiento normal requiere que todos los nodos estén disponibles en un momento determinado.

Algunos de los proyectos aquí listados son de pago, otros son completamente gratuitos, otros ofrecen soporte o disponen de una gran comunidad, etc. En GNU/Linux disponemos de diversas soluciones pudiendo escoger la que más se adapte a nuestras necesidades.

A continuación nos vamos a centrar en la herramienta **Veritas Cluster Server** por ser la solución de alta disponibilidad enfocada a aplicaciones y ser de las más fiables y extendida entre las grandes organizaciones. Además de poseer una comunidad muy activa y un soporte muy efectivo.

5. VERITAS™ CLUSTER SERVER

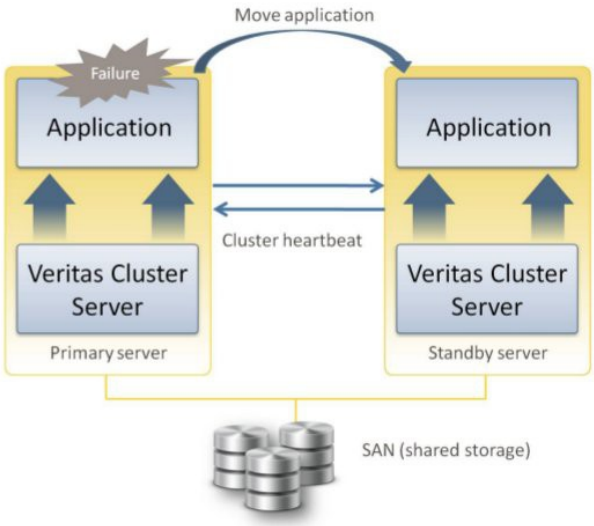
5.1.1. Descripción general VCS

Veritas Cluster Server de Symantec mantiene sus aplicaciones más importantes en funcionamiento 24 horas al día, todos los días del año, sin intervención manual, y automatiza la planificación de la recuperación después de un desastre con eficacia y resultados predecibles. Cluster Server proporciona una solución infalible que detecta los riesgos para la disponibilidad de las aplicaciones y automatiza la recuperación de aplicaciones a fin de proporcionar alta disponibilidad y recuperación después de un desastre.

Veritas Cluster Server supervisa de manera inteligente las aplicaciones y la infraestructura, detecta posibles riesgos de disponibilidad y recupera las aplicaciones de forma automática.

Veritas Cluster Server también puede detectar interrupciones de sites e iniciar una recuperación de aplicaciones en el site de recuperación después de un desastre.

Veritas Cluster Server incluye pruebas integradas para detectar e informar de manera preventiva acerca de problemas potenciales antes de que afecten a los servicios de TI.



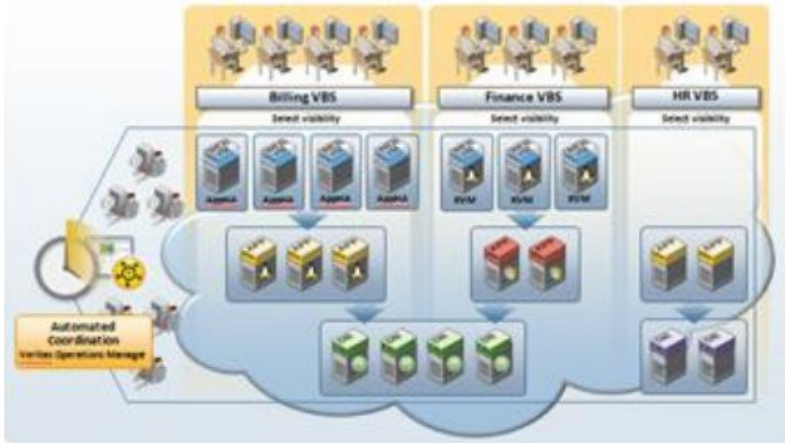
5.1.2. Aspectos clave

- **Servicios Empresariales Virtuales**: proporcionan una recuperación más rápida y un tiempo mínimo de perdida de servicio para aplicaciones compuestas por varios componentes que se ejecutan en diferentes niveles virtuales y/o físicos. Esto añade resistencia y flexibilidad a los servicios empresariales.

  Para adaptarse a las necesidades empresariales en evolución, los centros de datos actuales cuentan con varias capas de entornos físicos y virtuales, cada uno con sus propios requisitos de autorización y sus propias herramientas administrativas. Esto genera una complejidad de gestión de extremo a extremo con aplicaciones o servicios empresariales completos que cuentan con varios componentes que interactúan entre sí a pesar de estar ejecutándose en tecnologías de virtualización y sistemas operativos diferentes.

  Los Servicios Empresariales Virtuales, incluidos con VCS y gestionados por medio de Veritas Operations Manager, tienen en cuenta el servicio empresarial completo y permiten tomar medidas en caso de error. Cuando se produce un error en un componente individual del servicio, VCS recupera la aplicación con errores y organiza automáticamente la conexión a otros recursos informáticos necesarios para mantener disponibles los servicios empresariales. El resultado final es una recuperación más rápida y tiempo fuera de servicio mínimo sin necesidad de intervención manual.

  En la siguiente imagen podemos ver un ejemplo de Servicios Empresariales Virtuales mostrando los componentes básicos de la nube privada.



- **Compatibilidad inmediata con bases de dato datos y aplicaciones**: garantiza la compatibilidad con cientos de aplicaciones y bases de datos, lo que disminuye los tiempos de implementación y los costes de consultoría.

  La instalación de otras soluciones en clúster suele requerir proyectos de consultoría extensos y scripts personalizados para admitir diferentes aplicaciones y bases de datos en el entorno. VCS proporciona compatibilidad inmediata con una amplia gama de aplicaciones, entre ellas SAP, BEA, Siebel, aplicaciones Oracle, Microsoft Exchange y PeopleSoft, así como bases de datos de nivel empresarial, como Oracle, DB2, Microsoft SQL Server y Sybase. Asimismo, Symantec admite nuevas aplicaciones cada trimestre.

  El uso de VCS se traduce en menos scripts, tiempos de instalación más rápidos y mantenimiento continuo más sencillo cuando actualiza las aplicaciones. Asimismo, admite aplicaciones personalizadas con agentes personalizados o genéricos.

- **Amplia compatibilidad de hardware y plataformas:** Usa la misma herramienta en las plataformas físicas y virtuales. Soporta los principales sistemas operativos, incluyendo UNIX, Microsoft Windows, Linux y plataformas virtuales, incluyendo VMware ESX, Red Hat Enterprise Virtualization (RHEV), Oracle VM y Microsoft Hyper-V. Con esto se reducen costes administrativos, de formación y de hardware.

  La existencia de varias plataformas suele significar distintas herramientas de alta disponibilidad y, por lo tanto, aumento de la complejidad. Esta complejidad puede suponer un incremento de los costes y una mayor probabilidad de errores. VCS es la única solución que admite todos los sistemas operativos líderes, entre ellos, UNIX, Microsoft Windows, Linux, y plataformas virtuales, así como la más amplia gama de configuraciones de hardware heterogéneas. Con VCS las empresas pueden añadir alta disponibilidad a su infraestructura actual sin tener que adquirir hardware adicional. Las organizaciones pueden combinar los servidores y el almacenamiento con un solo clúster y compartir la infraestructura de almacenamiento con la misma herramienta en todas las plataformas, lo que reduce los costes administrativos, de formación y de hardware.

- **Compatibilidad avanzada con máquinas virtuales:** Con las tecnologías de virtualización, suelen alojarse varios equipos virtuales en un solo servidor físico. Un error de ese servidor físico puede generar pérdida de disponibilidad en muchas aplicaciones. Como resultado, la necesidad de proporcionar servicios de alta disponibilidad aumenta con el uso de tecnologías de virtualización. VCS ofrece a los entornos virtuales un mayor nivel de disponibilidad al supervisar la aplicación en la máquina virtual y el estado del servidor subyacente y la máquina virtual. Ofrece una única solución para agrupar sistemas físicos y virtuales, lo que significa operaciones más económicas y simples y menos complejidad.

- **Detección de errores más rápida**: detecta errores con mayor rapidez que las soluciones de clústeres tradicionales y prácticamente no requiere sobrecarga de la CPU.

- **Detección de puntos finales más rápida:** Los clústeres normales confían en los sondeos de recursos para determinar el estado de los recursos de las aplicaciones. Este proceso de sondeo incrementa la sobrecarga del procesador, pero lo que es más importante, es posible que los errores no se detecten inmediatamente.

  Hace posible la detección más rápida de errores al supervisar de forma asíncrona los recursos seleccionados con su Marco de supervisión inteligente. Esto significa que los errores se pueden detectar instantáneamente en lugar de esperar una ausencia de respuesta de un recurso con errores. También reduce la sobrecarga del CPU asociada con la supervisión tradicional basada en sondeos.

- **Lógica de conmutación por error a avanzada**: garantiza que los recursos se usen de la forma más eficiente posible recuperando las aplicaciones en el servidor más apropiado. La mayoría de las soluciones de clústeres recomiendan usar configuraciones de clústeres de dos nodos activos/pasivos. Esto significa que los servidores de copias de seguridad permanecen inactivos, lo que genera la pérdida de recursos informáticos y la disminución del uso de los servidores. Con VCS se pueden establecer políticas de conmutación por error basadas en la capacidad de los servidores. Después, VCS elige el mejor servidor

para una aplicación específica en el momento que se produce el error según las necesidades de la aplicación y el estado actual de los recursos que existen en el clúster. Cuando se produce un error, VCS puede elegir automáticamente el servidor menos utilizado y vuelve a añadir automáticamente servidores reparados al grupo de selección cuando vuelve a unirse al clúster. La lógica de conmutación por error avanzada de VCS garantiza que el tiempo en servicio de una aplicación se maximice y que los recursos del servidor se utilicen de manera eficaz.

- **Disponibilidad a cualquier distancia**: crea clústeres locales y remotos para la alta disponibilidad y recuperación después de un desastre y funciona con todas las principales tecnologías de replicación de datos.

  La recuperación después de un desastre no consiste solo en replicar datos. Para aplicaciones de uso crítico que deben permanecer online incluso en caso de error en el sitio, las soluciones de recuperación después de un desastre también deberían automatizar la recuperación de aplicaciones.

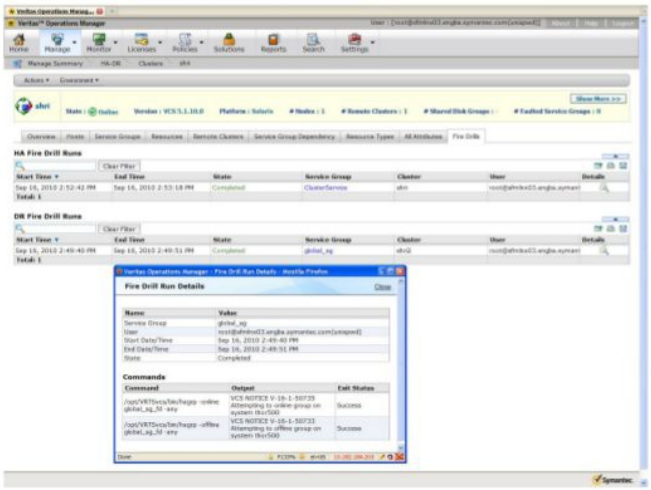
  VCS proporciona recuperación después de un desastre a cualquier distancia. Con un solo clic, Veritas Cluster Server puede migrar aplicaciones entre servidores en un centro de datos local, o transferir todas las aplicaciones a un centro de datos que se encuentra a miles de kilómetros de distancia.

  Veritas Cluster Server admite todas las principales tecnologías de replicación de bases de datos, software y hardware ofreciendo una solución integrada para aplicaciones y disponibilidad de datos. Veritas Replicator, una opción de Veritas Storage FoundationTM, proporciona replicación continua de datos que transfiere información a cualquier distancia. Ofrece compatibilidad total con todas las principales soluciones de replicación de datos de otros fabricantes.

  Su uso para la recuperación después de un desastre se traduce en tiempos de recuperación más rápidos, menos dependencia del personal durante un desastre y gestión más simple de la recuperación de aplicaciones y datos. Automatiza completamente el proceso de gestión de la replicación y el inicio de aplicaciones en el sitio remoto sin complicados procedimientos de recuperación manual que involucran a administradores de aplicaciones y almacenamiento.

- **Pruebas automatizadas de recuperación después de un desastre**: prueba la conmutación por error de las aplicaciones sin afectar al entorno principal.

  Dado que las aplicaciones y los servidores de producción cambian constantemente, las pruebas regulares de una estrategia de recuperación después de un desastre son críticas para garantizar una recuperación correcta en caso de interrupción. Para habilitar la recuperación correcta, VCS incluye Fire Drill, una herramienta que simula pruebas de recuperación después de un desastre mediante el reinicio de la aplicación en el sitio de recuperación después de un desastre como si fuera un desastre real. Dado que es una simulación, Fire Drill no interrumpe las aplicaciones de producción, de modo que puede ejecutarlo con la frecuencia necesaria sin provocar interrupciones ni afrontar el coste de las pruebas tradicionales de recuperación después de un desastre.



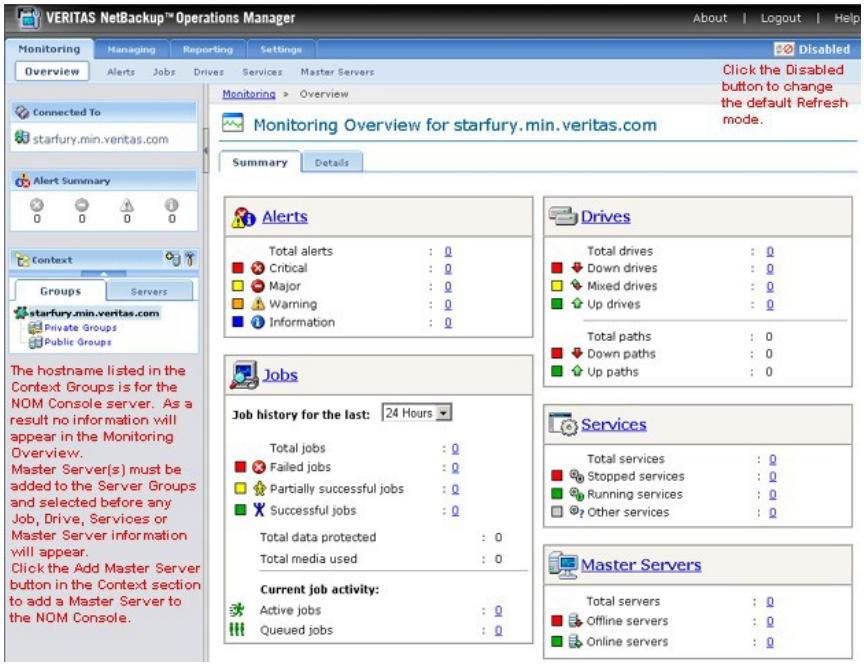
- **Instalación, configuración y gestión sencillas**: proporciona instalación mediante asistente para facilitar la implementación y una consola de gestión central para ver y gestionar varios clústeres remotos y locales.

  La implementación de nuevos clústeres puede ser difícil con otras soluciones de clústeres debido a los excesivos requisitos de scripts y herramientas de instalación deficientes. VCS proporciona administradores con asistentes de configuración fáciles de usar para lograr una implementación de clústeres simplificada. Asimismo, los administradores de clústeres pueden utilizar Symantec Operations Readiness Tool (SORT), una herramienta basada en Web que proporciona servicios de evaluación de riesgos y automatiza muchas de las tareas de instalación. Cluster Simulator, una descarga gratuita, permite a los administradores de clústeres simular situaciones de conmutación por error de aplicaciones y familiarizarse con VVCS sin correr riesgos y sin afectar los entornos de producción.

- **Elaboración de informes y gestión varios clústeres**: permite a los administradores supervisar, gestionar y elaborar informes en las implementaciones de VCS en diferentes plataformas desde una única consola basada en Web.

  Con una cantidad cada vez mayor de aplicaciones y servidores en clúster heterogéneos distribuidos en varios centros de datos, la gestión de clústeres puede resultar muy difícil. El uso de Veritas Operations Manager permite a los clientes supervisar, gestionar y elaborar informes sobre las implementaciones de VCS en diferentes plataformas desde una sola consola basada en Web. Las capacidades de gestión de VOM incrementan la eficiencia del administrador al proporcionar visualización mejorada de los clústeres gestionados, control centralizado de las aplicaciones globales e informes completos de cada estado de disponibilidad de las aplicaciones ayudando a los administradores a evitar errores comunes de configuración de clústeres, auditar cambios inesperados de configuración de clústeres y proporcionar un método estándar para que los administradores detecten e investiguen problemas del clúster.

En la siguiente figura vemos como con Veritas Operations Manager, puede ver el estado de todos los clústeres del centro de datos.



- **Protección de la integridad de los datos**: protege los datos de las aplicaciones mediante la prevención y la mediación avanzada de redes de concurrencia de aplicaciones en entornos de conmutación por error.

  Cuando se interrumpa la comunicación de un clúster, es posible que dos sistemas en un clúster intenten escribir en el mismo almacenamiento y dañen los datos. La lógica de protección de datos avanzada de Veritas Cluster Server protege los datos contra los daños cuando se produce una situación de partición, ya que proporciona mediación por medio de las decisiones de clústeres asociados. Esto garantiza la integridad de los datos y la disponibilidad del servicio.

5.1.3. Sistemas operativos compatibles

HP-UX
IBM AIX
Linux
Microsoft Windows
Oracle Solaris
VMware

5.2. Arquitecturas de recuperación tras desastres con soporte VCS

Veritas Cluster Server, la solución líder de la industria en clústeres de sistemas abiertos, es ideal para reducir el tiempo de inactividad planeado y no planeado, al facilitar la consolidación del servidor y administrar eficazmente una amplia gama de aplicaciones en entornos heterogéneos. Debido a que admite hasta 32 clústeres de nodo, Veritas Cluster Server ofrece el poder y la flexibilidad para proteger todo, desde una única instancia de base de datos crítica hasta grandes clústeres multiaplicaciones distribuidos geográficamente.

VCS ofrece una gran automatización que permite a los administradores de sistemas probar los planes de recuperación tras desastres sin interrupciones. Ofrece una administración inteligente de la carga de trabajo que permite aprovechar al máximo los recursos.

Al utilizar VCS como solución independiente, o en combinación con otros productos, podemos contar con disponibilidad en prácticamente cualquier entorno de sistema abierto. Ofrece protección en tiempo real gracias al soporte para diversas arquitecturas de clúster:

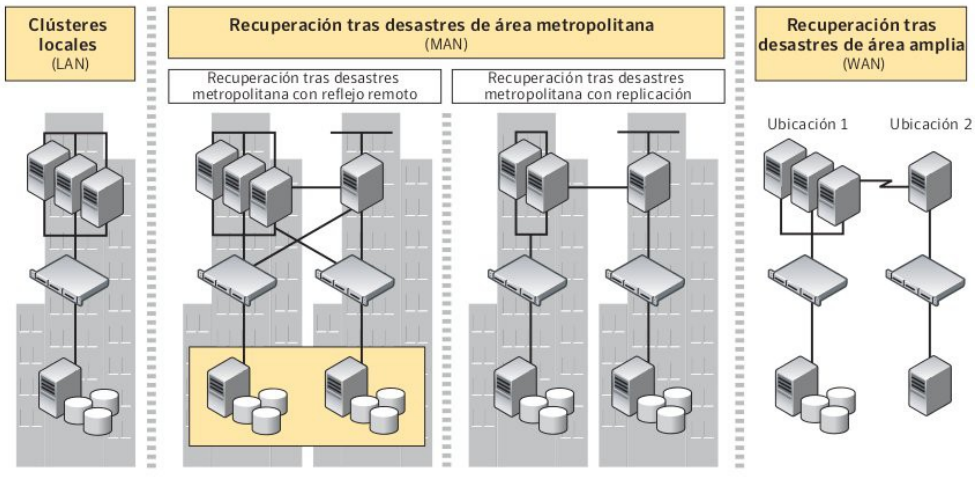
- Clústeres locales para alta disponibilidad
- Clústeres de área metropolitana para recuperación tras desastres de corto alcance (MAN)
- Clústeres de área amplia para recuperación tras desastres (WAN)

Symantec puede ofrecer alta disponibilidad (HA, high availability) y recuperación tras desastres (DR, disaster recovery) para los requisitos de cualquier empresa.

Las principales Ventajas son:

- La arquitectura de clústeres flexible proporciona una alta disponibilidad local para la recuperación de desastres en un área extensa.
- La integración con la tecnología de administración del almacenamiento y replicación Veritas ayuda a garantizar la protección no sólo de las aplicaciones y las bases de datos, sino también de los datos.
- La compatibilidad con tecnologías de administración del almacenamiento y replicación de otros fabricantes permite aprovechar al máximo la inversión.

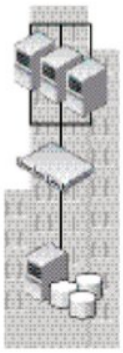
5.2.1. Arquitecturas de VCS



5.2.2.1. Clústeres locales

Un clúster único de VCS consiste en varios sistemas conectados en diversas combinaciones a dispositivos de almacenamiento compartidos. VCS supervisa y controla las aplicaciones y las bases de datos, y puede conmutarlas o reiniciarlas en caso de ocurrir fallas de hardware o software. Un clúster es un conjunto de sistemas conectados mediante interconexiones de red redundantes.

Esta solución ofrece recuperación local de servidores UNIX, Windows® y Linux® en caso de fallas de aplicaciones, sistemas operativos o hardware en un solo sitio.



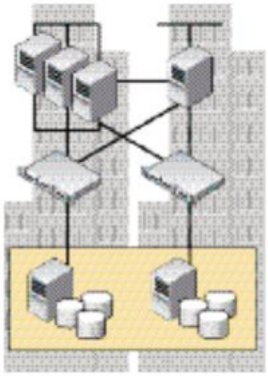
Entorno :

- Clúster único; una o varias subredes; hasta 32 servidores con cualquier configuración de clúster.
- Arquitectura de servidor, red y almacenamiento redundante.
- Cada sistema del clúster tiene acceso, cuando sea necesario, a los datos de las aplicaciones del almacenamiento compartido.
- Todos los servidores del clúster se encuentran en la misma ubicación (centro de datos único).

5.2.2.2. Recuperación tras desastres de área metropolitana

Con reflejo remoto

La recuperación tras desastres de área metropolitana con reflejo remoto (o clústeres de campo) consiste en un clúster único que abarca dos sitios mediante una conexión de canal de fibra (generalmente con tecnología de multiplexación por división de longitud de onda densa [DWDM]), a fin de ofrecer conexiones de SAN para el reflejo de datos y conexiones de red para la comunicación de clústeres. Además, esta arquitectura provee disponibilidad continua para los centros de datos en expansión que deben afrontar limitaciones de espacio para su crecimiento y permite contar con hardware de almacenamiento distinto en cada uno de los dos sitios.

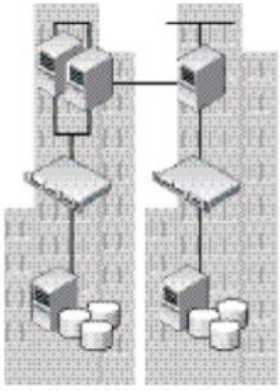


Entorno

- Clúster único que abarca varios emplazamientos, centros de datos o sitios conectados mediante canal de fibra exclusivo. La red pública debe alcanzar todos los sitios.
- Se distribuyen hasta 32 servidores libremente entre los distintos emplazamientos.
- El almacenamiento local se refleja (o se sincroniza) con Veritas Storage FoundationTM v5.0.

Con replicación

La recuperación tras desastres de área metropolitana con replicación es similar a la recuperación con reflejo remoto; sin embargo, no requiere una SAN extendida para el reflejo de datos. En lugar de grabar los datos de forma sincrónica en ambos lados mediante el reflejo remoto, los datos se replican sincrónicamente utilizando VeritasTM Volume Replicator sobre IP o la plataforma de replicación de hardware que el cliente prefiera. Esta arquitectura requiere vínculos de comunicación exclusivos entre los sitios para la comunicación de los clústeres.



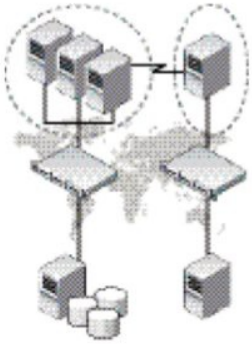
Entorno

- Clúster único que abarca varios emplazamientos, centros de datos o sitios conectados mediante vínculos de comunicación redundantes para la comunicación de los clústeres. La red pública debe alcanzar todos los sitios.
- Se distribuyen hasta 32 servidores libremente entre los distintos emplazamientos, centros de datos o sitios, con cualquier configuración de clúster.
- Se replica el almacenamiento local (sincrónicamente) con Veritas Volume Replicator u otras soluciones de replicación basadas en hardware.

5.2.2.3. Recuperación tras desastres de área amplia

La recuperación tras desastres de área amplia ofrece la mayor protección para los datos y las aplicaciones en caso de ocurrir un desastre. La arquitectura admite la distribución de dos o más centros de datos, clústeres y subredes separados por cualquier distancia. En caso de que haya problemas con un sitio, todos los servicios y los datos se transfieren al "sitio auxiliar" designado, que pasa a estar disponible para los usuarios. Por definición, la recuperación tras desastres de área amplia se utiliza cuando las distancias son mayores que las admitidas por una solución de reflejo sincrónico (más de 100 kilómetros). Puede usarse la configuración activo/auxiliar o activo/activo, donde cada sitio ofrece protección para el otro en caso de ocurrir un desastre.

La conexión por pulsos de clúster entre los sitios suele realizarse sobre IP.



Entorno

- Se implementa un clúster local en cada sitio y se replican los datos en uno o varios sitios secundarios.
- Las comunicaciones de clúster a clúster se realizan sobre IP. La replicación generalmente se realiza sobre IP, debido a las distancias implicadas.

6. PREREQUISITOS DE INSTALACIÓN VCS

Se definen a continuación requisitos previos a la instalación de Veritas Cluster Server 6.0.

6.1 Hardware

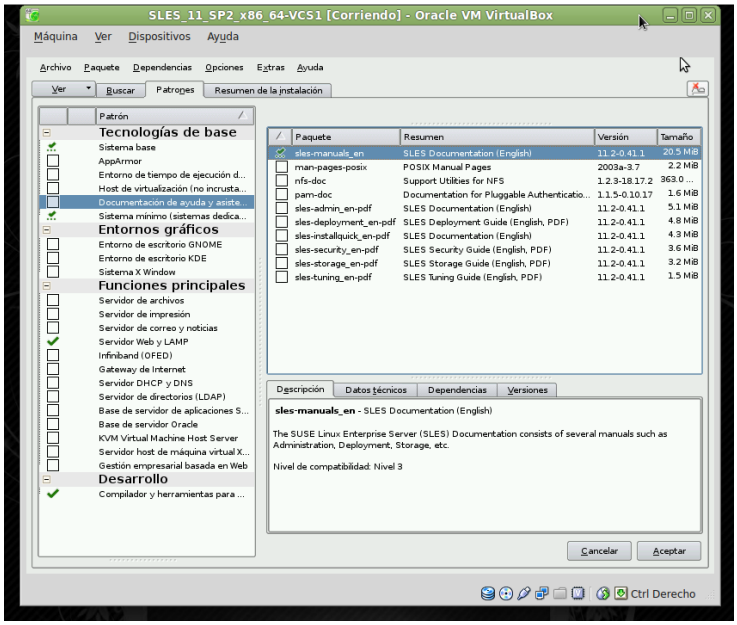
Requisitos previos:

- 2 o más servidores. Lo usual es un servidor en un CPD y otro en otro CPD lo bastante alejados para tener seguridad en catástrofes bien sea atmosféricas o de otra índole. Dependiendo del proyecto puede solicitarse la instalación en BladeCenters.
- Recomendado, 4 tarjetas ethernet por servidor. Para TFC al visualizar por Virtual Box se montan 3 tarjetas de red, una de ellas para comunicación entre máquinas al montar el cluster.
- Recomendado 2 tarjetas de fibra por servidor. garantizando el multipathing a través de sus switch de fibra. Para TFC se monta mediante VirtualBox con discos compartidos.
- Se recomienda mínimo de 50Gb HDD SCSI en RAID-1 o disco de alta velocidad en la SAN, para instalación de SO.
- Mínimo de 2Gb de RAM.
- Acceso mediante RSA o BladeCenter.

6.2 Sistema Operativo

SUSE SLES 11 SP2 (no instalar SP3, no está soportado para 6.0) o RHEL de 64bits. La máquina deberá estar correctamente validada y subida a último nivel de paquetes.

Las opciones de instalación básicas se muestran en la siguiente pantalla:



6.3 Comunicaciones

Para cada servidor se necesitan al menos 3 IP's para Veritas.

- 2 IP's para el heartbeat.
- 1 IP de servicio para cada nodo. Deben de estar en la misma VLAN extendida.
- IP's virtuales del cluster. 1 IP por cada recurso que queramos que balancee. Esto será optativo, ya que proyectos accederán a los nodos de Veritas a través de un balanceador de carga (Apache)
- Nos la proporcionará el grupo de comunicaciones.

6.4 Almacenamiento

La cantidad de discos con los que trabajar dependerán de cada proyecto. Pata el TFC se simula un proyecto virtualizando máquinas mediante Oracle VM Virtualbox.

Oracle VM VirtualBox es un software de virtualización para arquitecturas x86/amd64, creado originalmente por la empresa alemana innotek GmbH. Actualmente es desarrollado por Oracle Corporation como parte de su familia de productos de virtualización. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como «sistemas invitados», dentro de otro sistema operativo «anfitrión», cada uno con su propio ambiente virtual.

Para una instalación tipo, en empresa con CPD, BladeCenter y discos de cabina, se recomienda lo siguiente:

- Discos independientes para datos y/o sistema operativo (en caso de que no se usen discos SCSI internos en RAID1) exclusivos para cada servidor.
- Discos de datos visibles desde ambos nodos y dados de alta en ambos CPD's, en la SAN extendida.
- Datos de las cabinas y de los discos para configurar los recursos.

6.5 Software

Software necesario para el cluster.

VRTS_SF_HA_Solutions_6.0_[RHEL|SLES].tar.gz.

Disponible en el repositorio de software de la empresa propietaria, Symantec:

http://www.symantec.com/es/es/cluster-server

Debemos descargar también las posibles actualizaciones disponibles para la versión de VCS que estámos instalando.

Las versiones se nombran mediante Maintenance Packs (MP) o Rolling Packs (RP). Las podemos descargar desde:

https://sort.symantec.com/patch/finder

NOTA: También debemos descargar los últimos parches disponibles P-patch.

6.6 Configuración de los sistemas

6.6.1 Tamaños de los FS

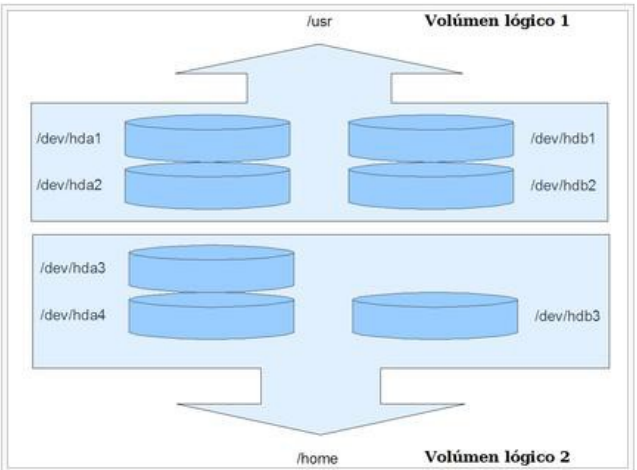
Debemos ampliar algunos FS ya que el tamaño que aplicamos por defecto en la instalación del sistema operativo no es suficiente. Los FS que ampliaremos son opt, usr y tmp. Para ello utilizaremos filesystem LVM.

LVM es una implementación de un administrador de volúmenes lógicos para el kernel Linux. Se escribió originalmente en 1998 por Heinz Mauelshagen, que se basó en el administrador de volúmenes de Veritas usado en sistemas HP-UX.

LVM incluye muchas de las características que se esperan de un administrador de volúmenes, incluyendo:

- Redimensionado de grupos lógicos
- Redimensionado de volúmenes lógicos
- Instantáneas de sólo lectura (LVM2 ofrece lectura y escritura)
- RAID0 de volúmenes lógicos.

LVM no implementa RAID1 o RAID5, por lo que se recomienda usar software específico de RAID para estas operaciones, teniendo las LV por encima del RAID.En la siguiente imagen podemos observar cómo trabaja LVM.



Esquema básico de LVM.

Los volúmenes lógicos agrupan particiones físicas de disco, y éstos a su vez, aunque no está representado en la figura, se engloban en un grupo lógico. De esta forma, /home se compone de hda3, hda4 y hdb3, y a su vez, /usr engloba a hda1, hda2, hdb1 y hdb2.

Utilizando LVM para la ampliación de nuestros FS:

```
lvextend -L 2.5G /dev/system_vg/opt_lv
lvextend -L 3G /dev/system_vg/usr_lv
```

El FS temporal lo ampliaremos en la maquina principal desde la que haremos la instalación a 8G y en el resto de máquinas del cluster, a 2Gb.

```
lvextend -L 8G /dev/system_vg/tmp_lv        #Nodo donde realizaremos la instalación
lvextend -L 2G /dev/system_vg/tmp_lv        #Resto de nodos del cluster
```

Una vez hayamos concluido la instalación, se puede liberar el espacio extra asignado al /tmp en la máquina de la instalación, dejándolo como en el resto, a 2Gb.

Hacemos efectivo el cambio:

```
resize2fs /dev/system_vg/opt_lv
resize2fs /dev/system_vg/usr_lv
resize2fs /dev/system_vg/tmp_lv
```

Hay que tener en cuenta que las operaciones de ampliación de FS se tienen que realizar en los dos nodos del cluster para que en los balanceos los filessytems de las máquinas sean idénticos. De no ser así es posible que no tengamos consistencia de datos y produciremos errores tanto a nivel de VCS como de aplicación.

6.6.2 Paquetes necesarios

Paquetes necesarios instalados en todos los sistemas del cluster son los siguientes:

- libacl-32bit
- pam-32bit
- libstdc++43-32bit

Para instalar directamente en sistemas SLES11, ejecutar:
zypper in libacl32bit pam32bit libstdc++4332bit

6.6.3 Configuración del PATH del sistema

Los siguiente procedimientos los tendremos que efectuar en todos los nodes del cluster.

La instalación, así como otros comandos de VCS, están localizados en los siguientes directorios:
/sbin, /usr/sbin, /opt/VRTS/bin, y /opt/VRTSvcs/bin.

Vamos a añadir estos directorios al PATH de root:
export PATH=$PATH:/sbin:/usr/sbin:/opt/VRTS/bin:/opt/VRTSvcs/bin:/opt/VRTSsfmh/bin

Los manuales del Volume Manager se encuentran en /opt/VRTS/man por lo que es conveniente añadirlos al path del MAN, del siguiente modo:
export MANPATH=$MANPATH:/opt/VRTS/man

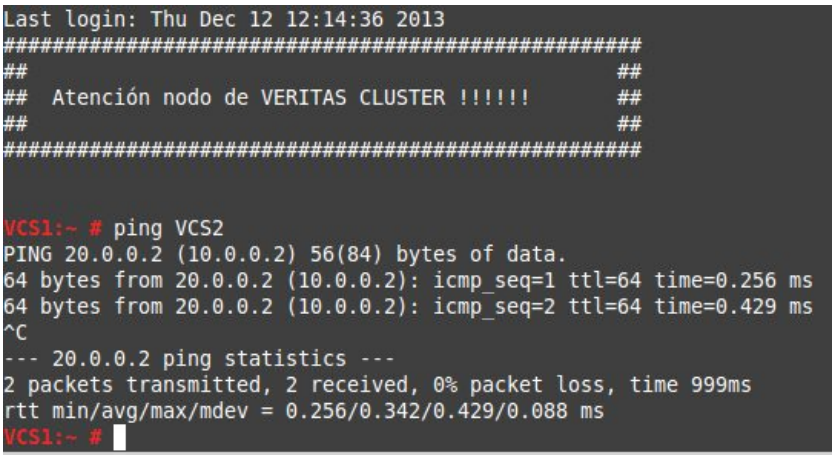
Incluir también en el .profile un "banner" para tener la certeza de que la persona que se logue en el sistema sepa que es un nodo de un cluster Veritas.

Contenido de /root/.profile en todos los nodos:

```
VCS1:~ # cat .profile
echo "#################################################"
echo "##                                             ##"
echo "##        Atención nodo de VERITAS CLUSTER !!!!!!      ##"
echo "##                                             ##"
echo "#################################################"
echo
echo
```

6.6.4 Configuración de la RED

Se debe comprobar mediante la herramienta de ping o telnet, que los nodos se ven entre sí.



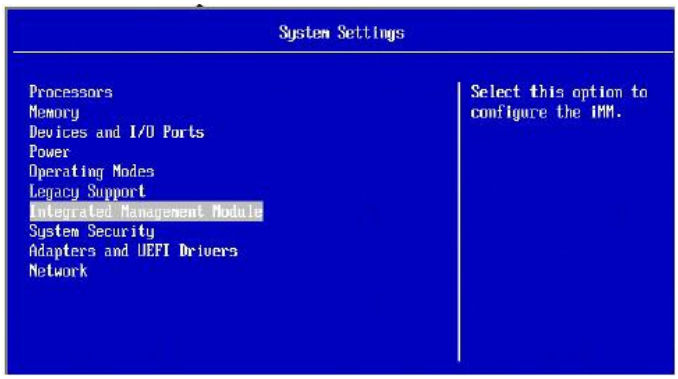
Es muy importante que las interfaces de red de las mismas VLAN's se llamen de la misma manera en ambos nodos. Por ejemplo, la ETH0 del NODO1 y NODO2, corresponde a la VLAN 1251, la ETH1 del NODO1 y NODO2, corresponden a la VLAN 1252, etc. Esto lo podemos realizar modificando las reglas de udev, en el path /etc/udev/rules.d/ o mediante YaST.

Tampoco deben de aparecer dispositivos de red extraños, del tipo:
ifconfig -a

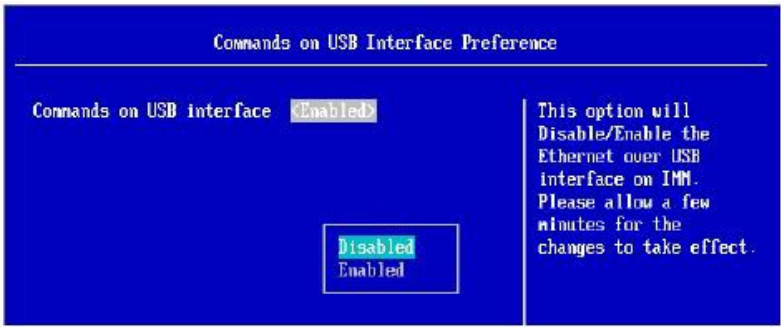
```
usb0      Link encap:Ethernet HWaddr E6:1F:13:7D:CE:DB
          BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 b) TX bytes:0 (
```

Para eliminar este dispositivo en concreto, hay que entrar en la BIOS del servidor. En la sección IMM, encontraremos la opción:



Deshabilitar “Commans on USB interface



6.6.5 LVM y UDEV

Para que los discos se vean de manera correcta en el sistema y evitar que de forma accidental se usen discos asignados al cluster, es necesario modificar las reglas de UDEV, y modificar los filtros de LVM.

Crearemos una regla nueva en /etc/udev/rules.d/, con el siguiente contenido:

```
vi /etc/udev/rules.d/99veritas
rules
# Renombrar dispositivos para Veritas.
ENV{ID_SERIAL}=="36005076303ffc1f40000000000009c00",
NAME="VERITAS_YECO_ID_36005076303ffc1f40000000000009c00"
ENV{ID_SERIAL}=="36005076303ffc1f40000000000009d00",
NAME="VERITAS_YECO_ID_36005076303ffc1f40000000000009d00"
ENV{ID_SERIAL}=="36005076309ffc1380000000000009c00",
NAME="VERITAS_TORR_ID_36005076309ffc1380000000000009c00"
ENV{ID_SERIAL}=="36005076309ffc1380000000000009d00",
NAME="VERITAS_TORR_ID_36005076309ffc1380000000000009d00"
```

Modificar los filtros de LVM, editando el archivo /etc/lvm/lvm.conf y añadiendo:

```
filter = [ "a|/dev/VERITAS.*|",
"a|/dev/sda|",
"r|/dev/disk/byid/
scsi36005076303ffc1f40000000000009c00|",
"r|/dev/disk/byid/scsi36005076303ffc1f40000000000009d00|",
"r|/dev/disk/byid/scsi36005076309ffc1380000000000009c00|",
"r|/dev/disk/byid/scsi36005076309ffc1380000000000009d00|",
"a|/dev/disk/byid/scsi.*|",
"r|.*|" ]
```

Para el TFC, se han seguido los siguientes pasos:

1. Obtenemos identificador

```
VCS2:~ # udevadm info -q all -n /dev/sdb | grep -i serial=
E: ID_SERIAL=VBOX_HARDDISK_VB1ba82f7a-cc1724b1
VCS2:~ # udevadm info -q all -n /dev/sdc | grep -i serial=
E: ID_SERIAL=VBOX_HARDDISK_VB3ead7b39-16700b64
```

O bien con:

```
# udevadm info -q all -p /sys/block/sdb | grep -i serial=
# udevadm info -q all -p /sys/block/sdc | grep -i serial=
```

2. Creamos fichero para las reglas de Veritas

```
VCS2:~ # cat /etc/udev/rules.d/99-veritas.rules
ENV{ID_SERIAL}=="VBOX_HARDDISK_VB1ba82f7a-cc1724b1",      NAME="VERITAS_VBOX_HARDDISK_VB1ba82f7a-
cc1724b1"
ENV{ID_SERIAL}=="VBOX_HARDDISK_VB3ead7b39-16700b64",      NAME="VERITAS_VBOX_HARDDISK_VB3ead7b39-
16700b64"
```

Posteriormente reiniciamos el demonio UDEV. Es posible que para que las reglas surtan efecto, haya que reiniciar la máquina (RHEL).

Tras el reinicio, los discos nos salen ya etiquetados:

Ejecutamos lvmdiskscan:

```
VCS2:~ # lvmdiskscan
  /dev/root                                              [       5.00 GiB]
  /dev/sda2                                              [1022.00 MiB]
  /dev/VERITAS_VBOX_HARDDISK_VB1ba82f7a-cc1724b1         [       1.00 GiB]
  /dev/VERITAS_VBOX_HARDDISK_VB3ead7b39-16700b64         [       1.00 GiB]
  /dev/sdd                                               [       1.00 GiB]
  2 disks
  3 partitions
  0 LVM physical volume whole disks
  0 LVM physical volumes
```

NOTA: La idea es que los discos que se van a usar desde VCS y que no van a llevar etiqueta de LVM, no aparezcan como discos que aparentemente “no están en uso”. Estas acciones hay que realizaras en todos los nodos del cluster.

6.6.6 Activar la comunicación SSH entre los sistemas

Cuando se instala VCS, este ofrece la opción de realizar la instalación en múltiples sistemas a la vez, evitando posibles equivocaciones, o modificación de configuración. Para ello es necesario configurar el comando rsh para que no pida password durante la comunicación entre nodos.

- En el nodo desde donde se vaya a lanzar la instalación, generamos la DSA key con el siguiente comando:
  ssh-keygen -t dsa
- Aceptamos la localización por defecto **~/.ssh/id_dsa**. Cuando se nos pregunte por la passphrase, confirmamos con un simple intro.
- Una vez generada la llave continuamos con:
  cp ~/.ssh/id_dsa.pub ~/.ssh/authorized_keys
  chmod 400 ~/.ssh/authorized_keys
- Por ultimo se procede a comprimir la carpeta .ssh y pasarla al otro servidor para ser descomprimida en la carpeta /root y el el HOME de usuario con el que se quiera tener esta relación:

```
[root@node1 ~]# tar cvfzp ssh.tgz /root/.ssh
[root@node1 ~]# scp ssh.tgz 192.168.0.2:/root
[root@node1 ~]# ssh 192.168.0.2
[root@node2 ~]# tar xvfzp ssh.tgz
```

- Activamos el acceso a root por ssh en el/los servidores remotos, ya que este queda desactivado en la validación. Activamos root en el archivo **/etc/ssh/sshd_config**:
  PermitRootLogin yes

- Configuramos el cliente de ssh para que el servidor desde donde hacemos la instalación ataque por el puerto 3106. Para ello, editar el archivo **/etc/ssh/ssh_config**:
  Port 3105 # si queremos que sea distinto a 22 por defecto

- Copiamos el ID generado en el primer nodo al resto de nodos, ejecutando:
  ssh-copy-id -i /root/.ssh/id_dsa.pub root@HOSTNAME2

- Para verificar que puedes conectar a los sistemas en donde VCS vaya a instalarse, ejecuta:
  ssh -x -l root HOSTNAME2 ls
  ssh -x -l root HOSTNAME2 ifconfig

Estos comandos deberían devolver la ejecución en el servidor remoto de los comandos "ls" e "ifconfig", sin pedir password o passphrase (excepto la confirmación de acceso por primera vez).

Si todos los pasos anteriores se han ejecutado con éxito, ya tenemos el entrono preparado para la instalación de VCS en múltiples sistemas al mismo tiempo.

7. INSTALACIÓN VCS

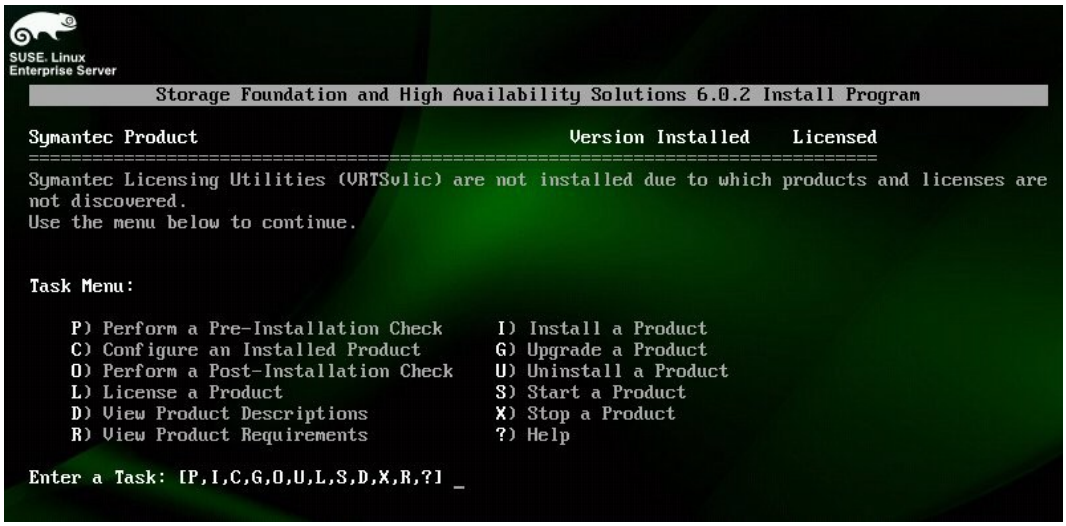
7.1 Instalación

El proceso de instalación del VCS unicamente hay que realizarlo en el nodo principal del cluster.

El proceso de instalación es el siguiente:

- Subir el software de VCS (sles11_x86_64) al directorio **/tmp**, descomprimirlo y ejecutar el instalador correspondiente a la versión del sistema operativo:

  ./installer



- Desde el menú, elegir la opción **I** que corresponde con "**Install a Product**".

- De la lista de productos a instalar, elegimos la opción **4** para "**Veritas Storage Foundation and High Availability (SFHA)**".

- Aceptar las condiciones de licencia de uso.

- Seleccionar la opción **3** para instalar todos los paquetes "**Install all rpms - 779 MB required**".

- Introducir los nombres de los nodos en donde se quiere realizar la instalación, separados por espacios.

En este momento se hace una comprobación de comunicación con el resto de nodos que van a formar el cluster. Si todo ha ido bien, nos mostrará una pantalla con los paquetes que va a instalar en todos los nodos. Pulsamos **INTRO** y esperamos a que termine la instalación.

A continuación, nos pedirá que introduzcamos los siguientes datos:

- Pulsamos **1** para introducir una licencia de uso válida para cada uno de los nodos.
  32ZUIDR&&C3X6F6OSB88@@#8KPO44O63P

  NOTA: Esta licencia mostrada no es real. Con la descarga del software en http://www.symantec.com/, se facilita una licencia de evaluación de 60 días.

- Pulsamos **N** para indicar que no queremos introducir licencias adicionales.

- Pulsamos **Y** para indicar que queremos iniciar la configuración del cluster.

- A continuación nos pregunta si queremos configurar IO Fencing:
  Do you want to configure I/O Fencing in enabled mode? [y,n,q,?]

  Seleccionamos **NO**. Lo configuraremos más tarde.

- A continuación nos solicita en nombre que queremos dar al cluster:
  Enter the unique cluster name: [q,?] NOMBRECLUSTER

  El formato de "NOMBRECLUSTER" será: [PROYECTO]_[PRODUCTO]_[ENTORNO]
  Ej.: SAN_CLUSTER_TFC

- Nos solicita que elijamos la manera en que vá a funcionar el HeartBeat:
  How would you like to configure heartbeat links? [1-3,b,q,?]

  Elegiremos la opción **1** por defecto "**Configure heartbeat links using LLT over Ethernet**"

- Posteriormente nos solicita que elijamos las interfaces de red sobre las que se va a gestionar el HeartBeat. Ya hemos mencionado en los prerequisitos que es imprescindible que en todos los nodos del cluster, las Ifaces de una misma VLAN se llamen de la misma manera. Configuraremos al menos dos interfaces para HeartBeat y otra de baja prioridad (la de servicio). El diálogo sería el siguiente:

  Enter the NIC for the first private heartbeat link On **NODO1**: [b,q,?] (eth1) **eth2**
  Would you like to configure a second private heartbeat link? [y,n,q,b,?] (y)
  Enter the NIC for the second private heartbeat link on **NODO1**: [b,q,?] (eth1) **eth3**
  Would you like to configure a third private heartbeat link? [y,n,q,b,?] (n)
  Do you want to configure an additional lowpriority heartbeat link? [y,n,q,b,?] (n) Y
  Enter the NIC for the lowpriority heartbeat link on **NODO1**: [b,q,?] (eth0) **eth0**
  Are you using the same NICs for private heartbeat links on all systems? [y,n,q,b,?] (y)

  Checking media speed for eth2 on NODO1 .....1000Mb/s
  Checking media speed for eth3 on NODO1 ….....1000Mb/s
  Checking media speed for eth2 on NODO2 ..... 1000Mb/s
  Checking media speed for eth3 on NODO2 ..... 1000Mb/s

- A continuación no solicita el ID único del cluster. En Linux, tenemos reservado el rango del 1000 al 2000.
  Enter a unique cluster ID number between 065535: [b,q,?] XXXX

Nos solicitará a continuación si queremos realizar un chequeo, para ver si el ID elegido ya está en uso. Seleccionamos **Y** para realizarlo como medio de seguridad.

Checking cluster ID ..... Done
Duplicated cluster ID detection passed. The cluster ID XXXX can be used for the cluster.

Una vez elegido el ID, editar dicho artículo y actualizar la info con el ID del nuevo cluster, para que quede documentado.

- A continuación nos aparecerá un resumen de la información recopilada para la instalación. Si todo está correcto, seleccionar Y para comenzar la instalación:

Cluster information verification: El resultado de la instalación del VCS para el TFC, es el siguiente:

```
Cluster   information verification:

        Cluster Name:         SAN_CLUSTER
        Cluster ID Number: 999
        Private Heartbeat NICs for vcs1:
                link1=eth1
                link2=eth2
        Low-Priority Heartbeat NIC for vcs1:
                link-lowpri1=eth0
        Private Heartbeat NICs for vcs2:
                link1=eth1
                link2=eth2
        Low-Priority Heartbeat NIC for vcs2:
                link-lowpri1=eth0

Is this information correct? [y,n,q,?] (y)
```

- A continuación nos pregunta si queremos configurar la IP del cluster. Elegiremos **NO**. Dependiendo de la arquitectura del cluste, nos hará falta o no. En caso de ser necesaria, se configura más adelante:
  Do you want to configure the Virtual IP? [y,n,q,?] n
- Ahora nos pregunta si queremos configurar los servicios de seguridad de Symantec. Seleccionaremos **NO**.
  Would you like to configure the VCS cluster in secure mode? [y,n,q,?] n

- Posteriormente, nos pregunta si aceptamos las credenciales por defecto a VCS. Seleccionaremos **SI**.
  Do you wish to accept the default cluster credentials of 'admin/password'? [y,n,q] y

- A continuación nos preguntan si queremos añadir más usuarios. Seleccionaremos **NO**.
  Do you want to add another user to the cluster? [y,n,q] n

- Aceptamos la información mostrada por pantalla relativa a usuarios y passwords.
  Is this information correct? [y,n,q] y

- Una vez llegados a este punto, nos pregunta si queremos configurar las notificacines vía SMTP. Seleccionamos **NO**.
  Do you want to configure SMTP notification? [y,n,q,?] n

- Finalmente nos pregunta si queremos configurar las notificacines vía SNMP. Esto, aunque es útil para la monitorización, de momento no lo configuramos: Seleccionamos **NO**.
  Do you want to configure SNMP notification? [y,n,q,?] n

  Finaliza la instalación básica.
  Veritas Storage Foundation and High Availability Startup completed successfully

Si ejecutamos de nuevo el ./installer, y pulsamos sobre la opción “I”, veremos la siguiente captura de pantalla con el software instalado:

Symantec Product	Version Installed	Licensed
Veritas File System	6.0.000.000	yes
Veritas Dynamic MultiPathing	6.0.000.000	yes
Veritas Volume Manager	6.0.000.000	yes
Veritas Cluster Server	6.0.000.000	yes
Veritas Storage Foundation	6.0.000.000	yes
Veritas Storage Foundation and High Availability	6.0.000.000	yes

7.2 Actualización al último RP

Coprobar en la URL https://sort.symantec.com/patch/finder si hay parches actualizados para la versión de VCS 6.0 y VSFHA 6.0. Descargarlos en caso afirmativo.

Descomprimimos el tar.gz (**sfha-sles11_x86_64-6.0RP1-patches.tar.gz**) con el ultimo RP disponible en el nodo principal, desde el que se ha hecho la instalación. Un buen lugar es /tmp. Entramos a la carpeta /tmp/sles11_x86_64. Ejecutamos:
./installrp

Esto iniciará un asistente que nos solicitará que confirmemos que queremos instalar los paquetes de actualización en ambos servidores:
Enter the 64 bit SLES11 system names separated by spaces: [q,?] **NODO1 NODO2**

Tras una comprobación de los sistemas, aparecerá:
The following Veritas Cluster Server patches will be installed on all systems:

Patch	Rpm
VRTSllt5.1.132.0	VRTSllt
VRTSgab5.1.132.0	VRTSgab
VRTSvxfen5.1.132.0	VRTSvxfen
VRTSamf5.1.132.0	VRTSamf
VRTSvcs5.1.132.0	VRTSvcs
VRTScps5.1.132.0	VRTScps
VRTSvcsag5.1.132.0	VRTSvcsag
VRTSvcsdr5.1.132.0	VRTSvcsdr
VRTSvcsea5.1.132.0	VRTSvcsea

Press [Enter] to continue:

Presionamos **INTRO**. Nos solicitará que confirmemos la parada de todos los servicios en ambos nodos. Seleccionamos **SI**.
All SFHA processes that are currently running must be stopped
Do you want to stop SFHA processes now? [y,n,q,?] y

A partir de este momento, realizará las acciones necesarias para aplicar el RP.
Finalmente aparece el mensaje:
Veritas Storage Foundation and High Availability Startup completed successfully

Desde el link anteriormente comentado https://sort.symantec.com/patch/finder, comprobaremos que no hay más parches para aplicar.

Por último reiniciaremos todos los nodos del cluster y comprobamos que arrancan correctamente.

8. VCFS Y VERITAS VOLUME MANAGER

8.1. Instalación VCFS

Lo primero que debemos hacer es asegurarnos que tenemos instalado el software de Veritas "Veritas Storage Foundation Cluster File System". Para ello, ejecutaremos el installer que hay en el path donde hemos descomprimido el instalador.

Nos aparecerá el siguiente menú:

```
Task Menu:
   P) Perform a Pre-Installation Check          I) Install a Product
   C) Configure an Installed Product            G) Upgrade a Product
   O) Perform a Post-Installation Check         U) Uninstall a Product
   L) License a Product                     S) Start a Product
   D) View Product Descriptions                 X) Stop a Product
   R) View Product Requirements                 ?) Help
```

Seleccionamos la opción I "Install a Product". Nos aparecerá el siguiente menú:

```
    1)  Veritas Dynamic Multi-Pathing (DMP)
    2)  Veritas Cluster Server (VCS)
    3)  Veritas Storage Foundation (SF)
    4)  Veritas Storage Foundation and High Availability (SFHA)
    5)  Veritas Storage Foundation Cluster File System HA (SFCFSHA)
    6)  Symantec VirtualStore (SVS)
    7)  Veritas Storage Foundation for Sybase ASE CE (SFSYBASECE)
    8)  Veritas Storage Foundation for Oracle RAC (SF Oracle RAC)
    b)  Back to previous menu
```

Seleccionaremos la opción 5 "Veritas Storage Foundation Cluster File System HA (SFCFSHA)". Aceptamos los términos de licencia. Nos aparecerá el siguiente menú:

```
1)  Install minimal required rpms - 466 MB required
2)  Install recommended rpms - 759 MB required
3)  Install all rpms - 781 MB required
4)  Display rpms to be installed for each option
```

Seleccionaremos la opción 3 "Install all rpms - 781 MB required ". A continuación, nos solicitará que introduzcamos los sistemas en los que se va a realizar la instalación:

```
Enter the system names 64 bit SLES11 separated by spaces:  HOSTNAME1 HOSTNAME2
```

A continuación veremos como progresa la instalación del producto. Una vez instalados todos los paquetes, nos aparecerá el siguiente menú:

```
1)  Enter a valid license key
2)  Enable keyless licensing and complete system licensing later
```

Seleccionaremos la opción 1 "Enter a valid license key ".

```
Enter a SFCFSHA license key: [b,q,?] AJZGS3__JTFUEZ8____68O4O4CPCPXIC6
```

No introduciremos licencias adicionales.

A continuación, nos preguntará si deseamos configurar SFCFSHA. Diremos que no.

A continuación, nos preguntará si deseamos reconfigurar VCS. Diremos que no, ya que lo hicimos en el punto 7 INSTALACIÓN VERITAS CLUSTER SERVER, durante la instalación del producto básico.

8.2. Inialización de VERITAS VOLUME MANAGER

Todos los pasos que afectan al VVM, se realizarán en uno sólo de los nodos.

La primera vez que arrancamos el volume manager puede que necesitemos inicializarlo. Si existe el archivos /etc/vx/reconfig.d/state.d/install-db significa que no esta inicializado y tendremos inicializar el Veritas Volume Manager. En caso de que no exista, podemos omitirlo y pasar al siguiente punto.

Al no estar inicializado , si intentamos ver los discos nos dará un error:
vxlisk list
VxVM vxdisk ERROR V-5-1-684 IPC failure: Configuration daemon is not accessible

Para solucionarlo, debemos ejecutar los siguientes comandos en todos los nodos del cluster:

- Eliminar el fichero /etc/vx/reconfig.d/state.d/install-db. Ejecutamos:
  vxdctl mode

  Deben de aparecer el mensaje “mode: disabled” o “mode: not-running”.

- Ejecutamos:
  vxconfigd

- Inicializar vxvm:
  vxdctl init <hostname>

- Ejecutamos:
  vxdctl enable

- Ejecutamos:
  vxconfigd

- Reiniciamos el servidor y comprobamos que funciona ejecutando un listado de discos con vxdisk list:

```
DEVICE       TYPE      DISK         GROUP        STATUS
aluadisk0_0  auto      -            -            error
aluadisk0_1  auto      -            -            error
aluadisk0_2  auto:none -            -            online invalid
```

Ahora listamos todos los discos de sistema desde VERITAS. Con vxlist podemos listar la información de los discos.

Si nos aparece un error del tipo:
VxVM DCLI vxlist ERROR V-5049971-158 Authentication or communication could not be established with the server.

debemos ejecutar en todos los nodos del cluster:

/opt/VRTSsfmh/adm/dclisetup.sh

Aunque nos salgan errores o warnings, no haremos caso y comprobaremos que ya se listan los discos:

```
vxlist
TY              DEVICE          DISK        DISKGROUP     SIZE   FREE STATUS
disk sda            -           -           -             -      uninitialized
disk sdaa           -           -           -             -      notsetup
disk sdab           -           -           -             -      notsetup
disk sdac           -           -           -             -      notsetup
disk sdad           -           -           -             -      notsetup
```

En este ejemplo, los discos se ven por el nombre que les ha dado el kernel. Para poder ver el nombre de los discos por su ID (es más cómodo), ejecutaremos el siguiente comando desde el nodo principal:
vxdiskadm

Seleccionaremos la opción "**20 Change the disk naming scheme**", y diremos que sí a "**Do you want to change the naming scheme?**"

Ahora, veremos los discos de la siguiente manera:

```
vxlist
TY   DEVICE                DISK   DISKGROUP   SIZE   FREE   STATUS
disk disk_0                -      -           -      -      uninitialized
disk mbi-ds8x000_8400      -      -           -      -      notsetup
disk mbi-ds8x000_8401      -      -           -      -      notsetup
disk mbi-ds8x000_8402      -      -           -      -      notsetup
disk mbi-ds8x000_8403      -      -           -      -      notsetup
```

Podemos usar la herramienta vxdisk para ver los discos disponibles mediante la lista de discos con vxdisk list o por path con vxdisk path. Si sale error en el STATUS, porque aún no los hemos inicializado, es normal.

8.3. Inicializar los discos

Con el comando "vxdisksetup" se puede realizar de una manera directa la inicialización de los discos:

```
vxdisksetup -if <DISCO>
```

Si nos da un error al intentar inicializar, ejecutar fdisk del disco, con la opción "w" (write table to disk and exit). También podemos inicializar los discos usando la herramienta de administración "vxdiskadm". Mediante este comando podemos realizar casi cualquier operación relacionada con los discos, de una manera rápida a través de menús.

Si ahora comprobamos los discos veremos que están inicializados:

```
vxlist
TY   DEVICE                DISK   DISKGROUP   SIZE     FREE   STATUS
disk disk_0                -      -           -        -      uninitialized
disk mbi-ds8x000_8400      -      -           199.95g  -      free
disk mbi-ds8x000_8401      -      -           199.95g  -      free
disk mbi-ds8x000_8402      -      -           199.95g  -      free
disk mbi-ds8x000_8403      -      -           199.95g  -      free

vxdisk list
DEVICE                TYPE          DISK   GROUP      STATUS
disk_0                auto:none     -      -          online invalid
mbi-ds8x000_8400      auto:cdsdisk  -      -          online
mbi-ds8x000_8401      auto:cdsdisk  -      -          online
mbi-ds8x000_8402      auto:cdsdisk  -      -          online
mbi-ds8x000_8403      auto:cdsdisk  -      -          online
```

Ahora ponemos las etiquetas a los nodos y a los discos.

A fin de poder identificar mejor los discos, Veritas permite añadir etiquetas a los discos, para saber a qué CPD o cabina pertenecen. Lo primero será dar a cada nodo una localización:

```
vxdctl set site=Torrejon          #Ejecutar en los nodos de Torrejón
vxdctl set site=Coslada           #Ejecutar en los nodos de Coslada
```

Comprobamos que cada nodo tiene su localización correcta:

```
vxdctl list | grep siteid
siteid:  Torrejon
```

Añadiremos los tags de los sites a cada disco (se pueden poner todos los discos de un mismo CPD a la vez):

```
vxdisk settag site=Torrejon mbi-ds8x001_8400 mbi-ds8x001_8401
vxdisk settag site=Coslada  mbi-ds8x000_8400 mbi-ds8x000_8401
```

También se puede crea contra el "enclosure", es decir, contra el conjunto de discos de una misma ubicación, en caso de que sean muchos con **vxlist encl**.

Comprobamos que los tags de cada disco son correctos:
vxdisk listtag

DEVICE	NAME	VALUE
mbi_ds8x000_8504	site	Coslada
mbi_ds8x000_8505	site	Coslada
mbi_ds8x001_8400	site	Torrejon
mbi_ds8x001_8401	site	Torrejon

8.4. Creación manual de DG con consistencia entre sitios

Una vez se ha inicializado el disco, será necesario crear un nuevo grupo de discos. Para ello, haremos lo siguiente:
vxdg [-s] init <NombreDG> mbi_ds8x000_8400 mbi_ds8x000_8401 mbi_ds8x000_8402 mbi_ds8x000

Donde shareddg es el nombre del DG, y a continuación TODOS los discos de AMBOS CPD's que vayan a usarse para este DG. Por defecto, usaremos el nombre "shareddg".

NOTA1: Al ser un DG en mirror, es necesario añadir los discos por pares, es decir, igual número de discos en cada CPD, con el mismo tamaño.
NOTA2: En caso de ser un DG para un Cluster-Filesystem, usar el modificador s. Con esto indicaremos que es un DG del tipo shared. Se necesita tener instalado el software VCFS.

Comprobaremos que todos los discos están en el nuevo DG, con los comandos:

```
VCS1:~ # vxlist dg
TY   DISKGROUP   IMPORT    STATUS       SIZE      FREE ID
dg   shareddg    private   enabled     1.42g   455.75m 1380741573.9.VCS1
VCS1:~ #
```

Private nos indica que es un DG que estará solo activo en un nodo del cluster a la vez (activo/pasivo). En caso de ser un DG para un Cluster-Filesystem (activo/activo), sería del tipo **shared**. Vemos que el DG tiene 4.04Tb. Realmente, son 2Tb en cada CPD, pero Veritas lo ve todo junto. Nunca podremos usar más de 2T en el cluster. Configuramos por cada site desde el que podremos acceder. Para ello, ejecutaremos desde el mismo nodo, los siguientes comandos:
vxdg-g <NombreDG> addsite Torrejon
vxdg-g <NombreDG> addsite Coslada

NOTA3: En caso de tener discos en un sólo CPD (como en el TFC), sólo se añade el tag de ese CPD.

Configuramos la consistencia entre sites. La consistencia entre sites es lo que crea realmente el "mirror" de discos.

NOTA4: Si sólo se dispone de un CPD como en el TFC, no tiene sentido ejecutar la consistencia entre sitios. Nos saltaremos este paso.
vxdg -g <NombreDG> set siteconsistent=on

Llegado a este punto, ya podríamos crear volúmenes lógicos.

8.5. Creación de LV'S

Para crear un nuevo volumen en un DG, lo haremos de la siguiente manera:
vxassist -g <NombreDG> make <NombreLV> <TamañoLV>

Mediante el comando **vxlist volume**, comprobamos que se ha creado correctamente:.

En el siguiente paso, sera dar formato a LV's y para dar formato a un volumen lógico, podemos usar el comando:
mkfs.vxfs /dev/vx/rdsk/<NombreDG>/<NombreLV>

También lo podemos hacer a través del path:
mkfs.vxfs /dev/vx/dsk/<NombreDG>/<NombreLV>

Para ver la información de los volúmenes creados usaremos vxprint. Podemos ver todos los volúmenes o los de un solo DG usando el parametro "-g shareddg", en nuestro caso no seria necesario al tener un solo DG.
vxprint -g <NombreDG> -hvt

Aplicando todo lo anterior y amoldando a las necesidades del TFC y el montaje no sobre máquinas fisicas sino en máquinas virtuales de Virtualbox, el resultado del montaje de los discos y filesystems para el TFC es el siguiente:

```
VCS1:~ # vxlist
TY   DEVICE   DISK   DISKGROUP        SIZE      FREE STATUS
disk sda      -      -                   -         - uninitialized
disk sdb      sdb    shareddg       989.87m   455.75m imported
disk sdc      sdc    shareddg       465.87m      0.00 imported

TY   DISKGROUP   IMPORT    STATUS       SIZE      FREE ID
dg   shareddg    private   enabled     1.42g   455.75m 1380741573.9.VCS1

TY   VOLUME          DISKGROUP        SIZE STATUS    LAYOUT   LINKAGE
vol  compartido_lv    shareddg      500.00m healthy   concat   -
vol  jboss_lv         shareddg      200.00m healthy   concat   -
vol  mysql_lv         shareddg      300.00m healthy   concat   -

TY   FS               FSTYPE        SIZE      FREE   %USED DEVICE_PATH                          MOUNT_POINT
fs   /                ext3         4.92g     1.17g    75% /dev/sda1                           /
fs   compartido       vxfs       500.00m   465.74m     1% /dev/vx/dsk/shareddg/compartido_lv   /compartido
fs   jboss-5.0.0.GA   vxfs       200.00m    68.79m    65% /dev/vx/dsk/shareddg/jboss_lv        /opt/jboss-5.0.0.GA
fs   mysql            vxfs       300.00m   278.29m     2% /dev/vx/dsk/shareddg/mysql_lv        /opt/mysql

TY   NAME   TYPE   STATE    WWN   PNAME
hba  c0     SCSI   online   -     c0
hba  c1     SCSI   online   -     c1
hba  c2     SCSI   online   -     c2

TY   ENCLR_NAME    ENCLR_MODEL   ARRAY_TYPE    STATUS      ENCLR_SNO
encl other_disks   ATA           OTHER_DISKS   connected   OTHER_DISKS
VCS1:~ #
```

```
VCS1:~ # vxprint
Disk group: shareddg

TY NAME         ASSOC        KSTATE   LENGTH   PLOFFS   STATE    TUTIL0  PUTIL0
dg shareddg     shareddg     -        -        -        -        -       -

dm sdb          sdb          -        2027264  -        -        -       -
dm sdc          sdc          -        954112   -        -        -       -

v  compartido_lv fsgen       ENABLED  1024000  -        ACTIVE   -       -
pl compartido_lv-01 compartido_lv ENABLED 1024000 -     ACTIVE   -       -
sd sdb-03       compartido_lv-01 ENABLED 69888 0        -        -       -
sd sdc-01       compartido_lv-01 ENABLED 954112 69888   -        -       -

v  jboss_lv     fsgen        ENABLED  409600   -        ACTIVE   -       -
pl pruebas_lv-01 jboss_lv    ENABLED  409600   -        ACTIVE   -       -
sd sdb-01       pruebas_lv-01 ENABLED 409600  0        -        -       -

v  mysql_lv     fsgen        ENABLED  614400   -        ACTIVE   -       -
pl mysql_lv-01  mysql_lv     ENABLED  614400   -        ACTIVE   -       -
sd sdb-02       mysql_lv-01  ENABLED  614400   0        -        -       -
```

9. CONFIGURACIÓN DE VCS

9.1. Archivos de configuración

Los ficheros de configuración los tenemos en el directorio /etc/VRTSvcs/conf/config/.

- **types.cf** define los tipos que podemos usar, aquí se define por ej. el tipo Apache, Oracle, Metromirror, etc.
- **main.cf** aquí definimos los recursos del cluster.

9.2. Parada y arranque de VCS

Arrancamos el cluster en cada nodo con hastart.

- **hastart**. Inicia el VCS en el nodo.
- **hastart -onenode**. Se usa el parametro onenode para clusters de un solo nodo, de forma que arranca sin los mecanismos de protección, como por ejemplo los usados para evitar split brain.

Para detener el motor de VCS existen diferentes mecanismos:

- **hastop -local**. Detiene el motor del cluster y las aplicaciones monitorizadas por VCS en el nodo desde donde se ejecuta el comando.
- **hastop -all**. Detiene el motor del cluster y las aplicaciones monitorizadas por VCS en todos los nodos que componen el cluster.
- **hastop -local -force**. Detiene el motor del cluster (monitorización) en el nodo, pero dejando las aplicaciones monitorizadas "online". En este caso las aplicaciones bajo el control de VCS siguen corriendo en el nodo, pero no se monitoriza su estado con lo que ante una incidencia en alguna de ellas no se tomará ninguna acción.
- **hastop -all -force**. Detiene el motor del cluster (monitorización) en todos los nodos que componen el cluster, pero dejando las aplicaciones monitorizadas "online". En este caso las aplicaciones bajo el control de VCS siguen corriendo en el nodo, pero no se monitoriza su estado con lo que ante una incidencia en alguna de ellas no se tomará ninguna acción.

9.3. Editar la configuración del cluster-server

Para editarlo podemos hacerlo de 2 maneras: editando los ficheros de configuración, usando comandos de consola y editando y desde la consola web VOM.

En cualquiera de las 2 modalidades es necesario hacer el cambio en solo un servidor, los cambios se propagan solos al resto de nodos.

9.3.1. Server Parado

Con el servidor parado podemos editar los ficheros de configuración sin problemas, una vez arranque el server se propagaran los cambios. Si modificamos los recursos la secuencia sería:

- Parar el server por si acaso este corriendo:
  hastop -all

- Salvar la configuración anterior:
  cp main.cf main.cf.save

- Editar.
  vi main.cf

- Comprobar los cambios.
  hacf -verify /etc/VRTSvcs/conf/config
  hacf -generate /etc/VRTSvcs/conf/config

- Arrancar el servicio.
  hastart

9.3.2. Desde consola

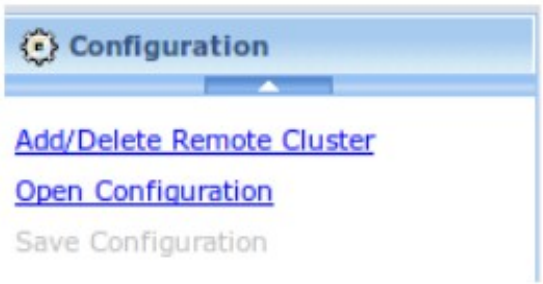
Si el server esta en marcha, no podemos modificar la configuración ya que esta en modo solo lectura, tenemos que cambiarlo a lectura escritura, realizar los cambios y subirlos para que los distribuya dejando la configuración en solo lectura.

haconf -makerw

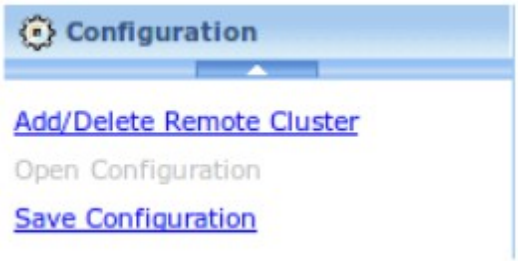
- Ejecutamos los comandos de modificación necesarios (ie. Hasys ....)
  haconf -dump -makero

9.3.3. Desde VOM

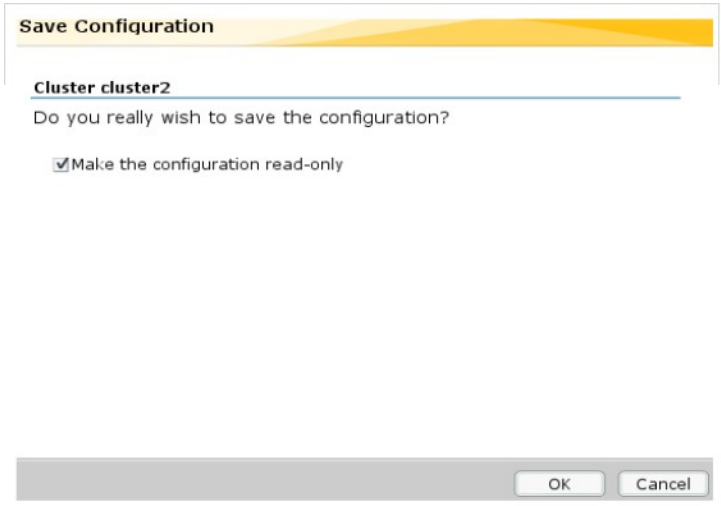
Una vez nos conectamos a la consola web de administración de la VOM (https://172.29.99.152:8181/cmc), no podemos modificar las configuración del cluster. Antes tenemos que abrirla para lectura escritura. Desde la pagina principal del cluster tenemos a la izquierda el panel de configuración con la opción “Open Configuration”, la pulsamos para poder realizar los cambios que necesitemos.



Una vez abierta la configuración realizamos los cambios y podemos salvarlo para que se propaguen los cambios por el cluster.



Al salvar nos la da la opción de devolver la configuración a modo de solo lectura, si no vamos a realizar mas cambios debemos dejarlo en solo lectura.



9.4. Configurar recursos

Storage agentes
Agentes para la gestion de discos, volumenes, etc...

- DiskGroup. Activa, desactiva y monitoriza Veritas Volume Manager (VxVM) disk group.
- DiskGroupSnap. Activa, desactiva y monitoriza disk groups usados en fire drill testing.
- DiskReservation. Reserva discos para garantizar el acceso exclusivo a los mismos. Solo para linux.
- Volume agent. Activa, desactiva y monitoriza Veritas Volume Manager (VxVM) volume.
- LVMLogicalVolume. Activa, desactiva y monitoriza Logical Volume Manager (LVM) logical volumes. Solo para Loinux y Hp-Ux.
- LVMVG. Activa, desactiva y monitoriza Logical Volume Manager (LVM) volume group.

Network agents
Agentes de red.

- NIC. Monitoriz una NIC (Network Interface Card)
- IP. Monitoriza una dirección IP.
- MultiNICA. Monitoriza multiples interfaces de red.
- IPMultiNIC. Gestiona Ip virtual configurada como un alias de una interfaz de un recurso MultiNICA.
- DNS. Actualiza y monitoriza el canonical name (CNAME) mapeando en DNS cuando se realiza un Failover entre redes.

File share agents
Agentes para la gestión de FS compartidos.

- NFS. Inicializa y monitoriza los recursos NFS, incluyendo el cliente y servidor. Se puede usar junto al agente NFSRestart.
- NFSRestart. Recupera el bloqueo NFS después de un reinicio o de una caída. De forma que previene corrupciones y asegura alta disponibilidad del bloqueo de NFS. Se configura con NFS agent como el recurso mas alto en el grupo de recursos.
- Share. Comparte, descomparte y monitoriza un directorio por NFS, requiere un servicio previo de tipo NFS o que este arrancado a nivel de SSOO el servidor NFS.
- Samba. Conjunto de 3 agentes que integran HA para las comparticiones Samba. Incluye los agentes SambaServer, SambaShare, and NetBIOS agents.
- NetBIOS. Inicia, para y monitoriza el demonio nmdb.

Services and Applications agents
Agentes de servicios y aplicaciones.

- Apache. Configura Apache en HA.
- Application. Arranca, para y monitoriza aplicaciones. Info: Application - howto
- Process. Arranca, para y monitoriza procesos de usuario.
- ProcessOnOnly. Arranca y monitoriza procesos de usuario.

Application tiene que monitorizar la aplicación que ejecuta de al menos una de las 3 maneras que tiene:

- PidFile. Controlamos que exista un fichero Id.
- MonitorProcesses. Monitorizamos una lista de procesos.
- MonitorProgram. Creamos un script que se encarga de monitorizar la aplicación. Debe devolver un 110 para indicar que la aplicación es totalmente fiable. 100 es que no es fiable. Ejemplo:

```
#!/bin/bash
USER="ora92"
PROCESS="/ora92/app/oracle/product/10/bin/tnslsnr LISTENER -inherit"
if [ $( ps -u ${USER} -o args | grep "${PROCESS}" | grep -vc grep ) -gt 0 ]
then
  exit 110
fi
exit 100    # OFFLINE
```

VCS infrastructure and support agents

- NotifierMngr. Monitoriza el proceso VCS notifier.
- VRTSWebApp. Arranca, para y monitoriza las aplicaciones configuradas usando Veritas Web Server (VRTSWeb).
- Proxy Agent. Monitoriza el estado de un recurso en un sistema local o remoto.
- Phantom Agent. Determina el estado de Service group con recursos de tipo None Only.
- RemoteGroup. Monitoriza el estado de un grupo de recursos remoto.

Testing agents

- ElifNone. Monitoriza un fichero, comprobando que no falte.
- FileNone. Monitoriza un fichero.
- FileOnOff. Crea un fichero, lo monitoriza y lo borra.
- FileOnOnly. Crea un fichero y lo monitoriza

9.4.1. Añadiendo configuración a MAIN.CF

Para añadir un recurso desde shell podemos o bien ejecutar comandos para crear recursos o bien editar el fichero main.cf. Nos vamos a centrar en el fichero de configuración main.cf.

Para crear un nuevo grupo tenemos que editar el fichero main.cf con el HA parado. Al añadir el grupo, lo primero que definimos son sus características:

- SystemList. Es la lista de servidores que usaran el grupo.
- AutoStarList. Indicamos los servers en el orden de arranque.
- ClusterFailOverPolicy. Indica el comportamiento del failover. Puede ser automático o manual, en caso de configurarse como manual el cluster no balancearía los recurso sin interacción del operador.
- OnlineRetryLimit. Numero de intentos para levantar el grupo.
- OnlineRetryInterval. Tiempo entre reintentos.

Una vez definidas las características del grupo definimos los recursos del mismo y sus parámetros, por ejemplo, para un recurso de tipo IP tendremos que indicar el device, la IP y la mascara de red.

Si un parámetro es distinto según el servidor pondremos el parámetro y el servidor con "@", si no especificamos server es una configuración global para el recurso, independientemente del servidor en el que se encuentre.

Por ultimo definiríamos las relaciones entre recursos, para crear el árbol de inicio, de forma que si un recurso

depende de otro no arranque hasta que el anterior no este arriba.

Ejemplo base, para poder hacer una prueba de balanceo de un DiskGroup entre nodos:

```
group ClusterGIS (
        SystemList = { EMGISL106 = 0, EMGISL105 = 1 }
        AutoStartList = { EMGISL106, EMGISL105 }
        ClusterFailOverPolicy = Auto
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )

DiskGroup ClusterDiskGroup (
      DiskGroup = shareddg
        )
```

Ejemplo de grupo que levanta una IP, monta los FS y arranca una BBDD Oracle:

```
DiskGroup ClusterDiskGroup (
        DiskGroup = clusterdg
        )
    IP SUIP (
        Device = eth0
        Address = "172.29.99.212"
        NetMask = "255.255.254.0"
        )
    Mount Ora92Mount (
        MountPoint = "/ora92"
        BlockDevice = "/dev/vx/dsk/clusterdg/ora92"
        FSType = vxfs
        FsckOpt = "-y"
        )
    Mount pacve9_oradataMount (
        MountPoint = "/pacve9_oradata"
        BlockDevice = "/dev/vx/dsk/clusterdg/pacve9_oradata"
        FSType = vxfs
        FsckOpt = "-y"
        )
    Mount pacve9_orarestoMount (
        MountPoint = "/pacve9_oraresto"
        BlockDevice = "/dev/vx/dsk/clusterdg/pacve9_oraresto"
        FSType = vxfs
        FsckOpt = "-y"
        )
    NIC SUETH (
        Device = eth0
        )
    Netlsnr ListenerOra10 (
        Owner = ora92
        Home = "/ora92/app/oracle/product/10"
        Listener = LISTENER
        )
    Oracle Oracle10 (
        Sid = PACVE11
        Owner = ora92
        Home = "/ora92/app/oracle/product/10"
        DetailMonitor = 1
        User = veritas
        Pword = aogMfoCmjOemBod
        Table = test
        )
    ListenerOra10 requires Oracle10
    Ora92Mount requires ClusterDiskGroup
    Oracle10 requires Ora92Mount
    Oracle10 requires pacve9_oradataMount
    Oracle10 requires pacve9_orarestoMount
    SUIP requires SUETH
    pacve9_oradataMount requires ClusterDiskGroup
    pacve9_orarestoMount requires ClusterDiskGroup
```

En el ejemplo anterior, hemos monitorizado Oracle usando los recursos del agente de Oracle pero también podríamos hacerlo con recursos genéricos, como application para Oracle y el listener.

En este caso configuraríamos dependencias entre Oracle y Listener.

```
Application Oracle9 (
        User = ora92
        StartProgram @emvcsl903pac = "/dbadmin/emvcsl903pac/scripts/arranca_bd.sh PACVE11"
        StartProgram @emvcsl904pac = "/dbadmin/emvcsl904pac/scripts/arranca_bd.sh PACVE11"
        StopProgram @emvcsl903pac = "/dbadmin/emvcsl903pac/scripts/para_bd.sh PACVE11"
        StopProgram @emvcsl904pac = "/dbadmin/emvcsl904pac/scripts/para_bd.sh PACVE11"
        MonitorProcesses = { ora_pmon_PACVE11 }
        )

Application ListenerOra (
        User = ora92
        StartProgram @emvcsl903pac = "/dbadmin/emvcsl903pac/scripts/arranca_listener.sh LISTENER"
        StartProgram @emvcsl904pac = "/dbadmin/emvcsl904pac/scripts/arranca_listener.sh LISTENER"
        StopProgram @emvcsl903pac = "/dbadmin/emvcsl903pac/scripts/para_listener.sh LISTENER"
        StopProgram @emvcsl904pac = "/dbadmin/emvcsl904pac/scripts/para_listener.sh LISTENER"
        MonitorProgram @emvcsl903pac = "/dbadmin/emvcsl903pac/scripts/monitor_listener.sh"
        MonitorProgram @emvcsl904pac = "/dbadmin/emvcsl904pac/scripts/monitor_listener.sh"
        MonitorProcesses = { "" }
        )
ListenerOra requires Oracle9
```

Para comprobar el estado del listener y no tener problemas para monitorizar el proceso del mismo al tener parámetros separados por espacios, podemos optar por un script de monitorización como el descrito más abajo. Este script devuelve 110 si todo va bien y 100 si esta caído (ERROR), los valores de 101 a 109 se usan para indicar que esta arriba pero con problemas mientras que 110 es todo OK.

```
#!/bin/bash
USER="ora92"
PROCESS="/ora92/app/oracle/product/10/bin/tnslsnr LISTENER -inherit"

#ps -u ${USER} -o args | grep  "${PROCESS}" | grep -vc grep

if [ $( ps -u ${USER} -o args | grep "${PROCESS}" | grep -vc grep) -gt 0 ]
then
  exit 110
fi
exit 100    # OFFLINE
```

9.5. Relaciones entre recursos

Los recursos se pueden linkar entre ellos de forma que un recurso dentro de un grupo no arranque antes que otro recurso, y así crear un árbol para el inicio de los recursos.

Los enlaces no tienen porque ser del tipo 1:1, sino que un recurso puede depender de varios recursos o varios recursos depender de él.

MAIN.CF

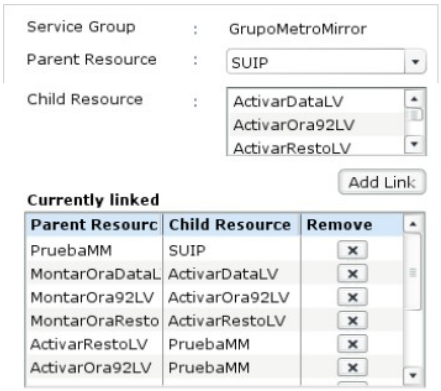
Dentro de la sección del grupo de recursos añadimos las relaciones entre recursos, indicando para el recurso de quien depende para poder ejecutar.

```
 SUIP requires SUETH
PruebaMM requires SUIP
ActivarOra92LV requires PruebaMM
MontarOra92LV requires ActivarOra92LV
Oracle9 requires MontarOra92LV
```

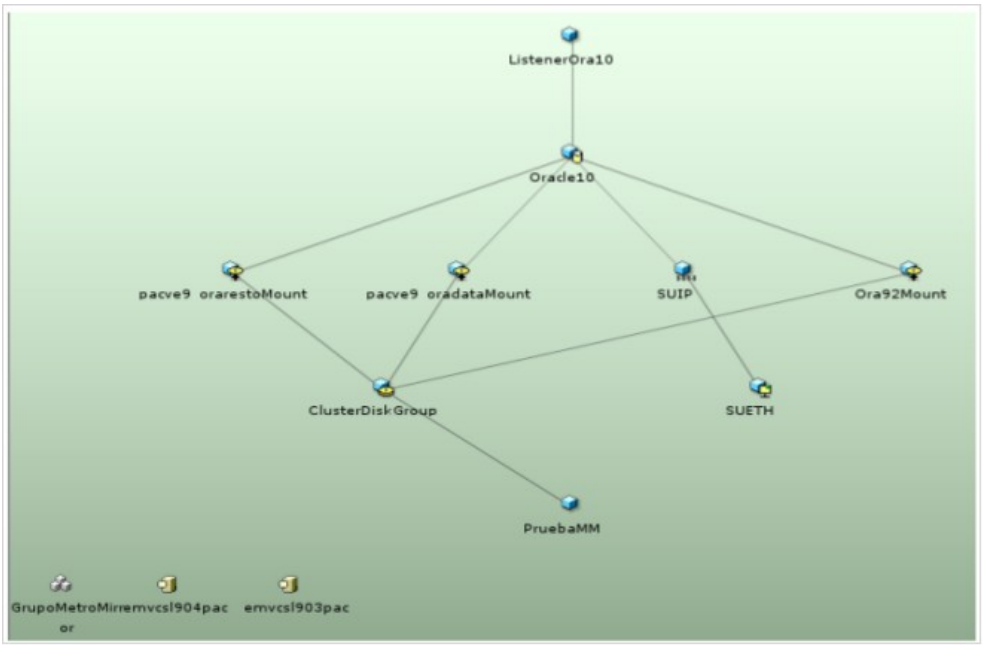
GUI

El orden de los enlaces va de abajo a arriba, de forma que el padre depende del hijo para poder arrancar. Al entrar en un grupo de recursos podemos configurar las dependencias desde el menú de la izquierda en la opción "Link/Unlink Resource".

Al pulsar el menú nos mostrara una ventana donde configurar las dependencias, tendremos que indicar el recurso padre, luego el recurso hijo y añadir el link, teniendo en cuenta que en este caso es el padre quien espera a que el hijo termine de arrancar para poder iniciarse.



Una vez creados los links podemos ver como queda el arbol de recursos en la pagina del grupo de recursos pulsando en la pestaña "Resources". Los recursos que estén a la misma altura en el árbol arrancaran a la vez, siempre que no tengan hijos o estos ya hayan terminado de arrancar.



9.6. Relaciones entre grupos

La relación entre grupos es parecida a la que encontramos entre recursos, para configurarlos desde la web vamos a la pestaña de grupos del cluster y pulsamos en el frame de la izquierda en la opción “Link Service Groups”.

Además de las relaciones padre hijo, también tenemos que tener especificar el tipo de relación y la fuerza de la relación.

Según el tipo podemos tener 2 tipos de dependencias:

- **Online**. El grupo padre debe esperar a que el hijo termine este online para poder iniciarse. Puede combinarse con cualquier tipo de localización.
- **Offline**. El grupo padre solo puede arrancar si el grupo hijo esta offline y viceversa. De esta forma se evita que algunas aplicaciones no puedan arrancar a la vez. Solo puede combinarse con localización Local.

3 tipos de localización, que se combinan con los tipos de localización para obtener el tipo de relación.

- **Local**. Los grupos padre e hijo están en el mismo sistema.
- **Global**. Pueden estar en cualquien sistema del cluster.
- **Remote**. El grupo padre no puede estar en el mismo sistema que el grupo hijo.

La fuerza de la relación puede ser de tres tipos:

- **Soft**. Indica un numero de condiciones mínimas para que los grupos padre e hijo puedan activarse. Un link Soft tiene las siguientes peculiaridades:
  - Si el grupo hijo falla el padre no se desactiva inmediatamente. Si el hijo puede realizar un failover el padre se mantendra online.
  - Cuando ambos grupos estan online, padre e hijo, podemos apagar cualquiera de ellos mientras el otro se mantiene online.
  - Al crear el link no es necesario que el hijo este online, indiferentemente de si el padre esta online.

- **Firm**. Tiene más condiciones para activar o desactivar los grupos. Sus caracteristicas son:
  - Si el hijo falle se parara el padre. Si el hijo no puede realizar failover el padre permanecerá parado.
  - Si el padre falla el hijo se mantendrá online.
  - El hijo no se puede apagar mientras el padre esta online. El padre sí puede pararse y el hijo se mantendra online.
  - Al crear el link el padre debe estar offline.

- **Hard**. Es el tipo más restrictivo, sus caracteristicas son:
  - Si el hijo falla se pone offline al padre antes de llevar a offline al hijo. Si el hijo balancea el padre también balanceara según el tipo de relación. Si el hijo no puede hacer un failover el padre permanecerá offline.
  - Si el padre falla el hijo se pondra offline.
  - Solo admite configuraciones Online local Hard.
  - Solo se puede definir una relación Hard de un nivel.
  - Poner online un grupo hijo no tiene porque levantar online al grupo padre.
  - Poner el padre offline no tiene porque poner offline al hijo.
  - El padre no puede ponerse online si el hijo esta offline.

10. CONFIGURACIÓN DE AGENTES EN VCS

10.1. Instalación de agentes

Dependiendo del ROL y los servicios que vaya a controlar el cluster, será necesario instalar los diversos agentes disponibles. La mayoría de los descrito en este apartado, vienen por defecto con la instalación de VCS.

La última versión de los agentes, se puede encontrar en el link:
https://sort.symantec.com/agents

10.2. Agentes de VCS para TFC

10.2.1. Agente GROUP

Este agente es el que se encarga de definir los grupos de recursos.

Ejemplo de una definición de un grupo de recursos, para un VCS activo/pasivo, donde por defecto los recursos arrancarían en el HOSTNAME1 y en caso de caída, el failover se realiza sobre el HOSTNAME2.

```
group Grupo_<nombre_grupo> (
        SystemList = { HOSTNAME1 = 0, HOSTNAME2 = 1 }
        AutoStartList = { HOSTNAME1, HOSTNAME2 }
        ClusterFailOverPolicy = Auto
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )
```

Ejemplo de una definición de un grupo de recursos, para VCS activo/activo, donde todos los recursos de este grupo arrancarían en ambos nodos a la vez.

```
group Grupo_<nombre_grupo> (
        SystemList = { HOSTNAME1 = 0, HOSTNAME2 = 1 }
        AutoStartList = { HOSTNAME1, HOSTNAME2 }
        AutoFailOver = 0
        Parallel = 1
        )
```

Dependencias obligatorias: No depende de nada.

Modificar/añadir agente desde línea de comandos:

```
haconf -makerw
hagrp -add Grupo_<nombre_grupo>
hagrp -modify Grupo_<nombre_grupo> SystemList HOSTNAME1 0 HOSTNAME2 1
hagrp -modify Grupo_<nombre_grupo> AutoStartList HOSTNAME1 HOSTAME2
hagrp -modify Grupo_<nombre_grupo> Parallel [0|1]
hagrp -modify Grupo_<nombre_grupo> AutoFailOver [0|1]
hagrp -modify Grupo_<nombre_grupo> ClusterFailOverPolicy [Auto| Manual|Connected]
hagrp -modify Grupo_<nombre_grupo> OnlineRetryLimit 3
hagrp -modify Grupo_<nombre_grupo> OnlineRetryInterval 120
haconf -dump -makero
```

Con el comando **hagrp -display Grupo_<nombre_grupo>** podemos ver todos los argumentos modificables del agente.

10.2.2. Agente DISKGROUP

El agente DiskGroup se encarga de activar, desactivar y lo monitorizar los DG's. Si los parámetros **StartVolumes** y **StopVolumes** están a 1 también se encarga de activar y desactivar los volúmenes lógico del DG.

Ejemplo de agente:

```
DiskGroup DG_<NombreDG> (
        DiskGroup = <NombreDG>
        )
```

Dependencias obligatorias: No depende de nada.

Modificar/añadir agente desde línea de comandos:

```
haconf -makerw
hares -add DG_<NombreDG> DiskGroup Grupo_<nombre_grupo>
hares -modify DG_<NombreDG> Enabled 1
hares -modify DG_<NombreDG> DiskGroup <NombreDG>
haconf -dump -makero
```

Con el comando **hares -display Grupo_<nombre_grupo>** podemos ver todos los argumentos modificables del agente.

10.2.3. Agente MOUNT

Monta los volúmenes en el punto de montaje especificado.

Ejemplo de agente para montaje de un FS de tipo VXFS:

```
Mount Montaje_<NombreLV> (
        MountPoint = "<MontajeLV>"
        BlockDevice = "/dev/vx/dsk/<NombreDG>/<NombreLV>"
        FSType = vxfs
        FsckOpt = “-y"
        )
```

Ejemplo de agente para montaje de un FS de tipo EXT3:

```
Mount Montaje_<NombreLV> (
        MountPoint = "<MontajeLV>"
        BlockDevice = "/dev/vx/dsk/<NombreVG>/<NombreLV>"
        FSType = ext3
        FsckOpt = “-y"
        )
```

Ejemplo de agente para montaje de un NFS:

```
Mount Montaje_NFS_<NombreImportNFS> (
        MountPoint = "<PathMontaje>"
        BlockDevice = "<HOSTNAME>:<PATH_EXPORTADO>"
        FSType = nfs
        )
```

Dependencias:

Montaje_<NombreLV> requires DG_<NombreDG>

Modificar/añadir agente desde línea de comandos:

```
haconf -makerw
hares -add Montaje_<NombreLV> Mount Grupo_<nombre_grupo>
hares -modify Montaje_<NombreLV> Enabled 1
hares -modify Montaje_<NombreLV> Critical 1
hares -modify Montaje_<NombreLV> BlockDevice "/dev/vx/dsk/<NombreVG>/<NombreLV>"
hares -modify Montaje_<NombreLV> FSType "vxfs"
hares -modify Montaje_<NombreLV> MountPoint "<PathMontaje>"
hares -modify Montaje_<NombreLV> FsckOpt %-y
haconf -dump -makero
```

Modificación online de las dependencias:

```
haconf -makerw
hares -link  Montaje_<NombreFS> <shareddg>
haconf -dump -makero
```

Con el comando **hares -display Grupo_<nombre_grupo>** podemos ver todos los argumentos modificables del agente.

10.2.4. Agente IP

Este agente se encarga de levantar una IP virtual sobre una o varias interfaces que le definamos. Este agente es dependiente (aunque no obligatoriamente) de un recurso del tipo NIC.

Ejemplo de agente:

```
IP IP_Cluster_<aplicacion> (
        Device @NODO1 = eth[0-9]
        Device @NODO2 = eth[0-9]
        Address = "XX.XX.XX.XX"
        NetMask = "255.255.254.0"
        )
```

Si en todos los nodos del cluster la interface donde se levanta la IP virtual es la misma, no hace falta definir los dos parámetros Device.

Dependencias:
IP_Cluster_<aplicacion> requires NIC_Grupo_<nombre_grupo>_eth[0-9]

Modificar/añadir agente desde línea de comandos:

```
haconf -makerw
hares -add IP_Cluster_<aplicacion> IP Grupo_<nombre_grupo>
hares -modify IP_Cluster_<aplicacion> Enabled 1
hares -modify IP_Cluster_<aplicacion> Address XX.XX.XX.XX
hares -modify IP_Cluster_<aplicacion> NetMask 255.255.254.0
hares -modify IP_Cluster_<aplicacion> Device eth[09]
haconf -dump -makero
Modificación online de las dependencias:
haconf -makerw
hares -link IP_Cluster_<aplicacion> NIC_eth[0-9]
haconf -dump -makero
```

Modificación online de las dependencias:

```
haconf -makerw
hares -link IP_Cluster_<aplicacion> NIC_eth[0-9]
haconf -dump -makero
```

10.2.5. Agente NIC

Este agente se encarga de monitorizar una interface de red. Ejemplo de agente:

```
NIC NIC_Grupo_<nombre_grupo>_eth[0-9] (
        Device @EMPV6L201PAC = eth[0-9]
        Device @EMPV6L202PAC = eth[0-9]
        NetworkHosts = { "XX.XX.XX.1" }
        Mii = 0
        )
```

Dependencia: No depende de nada.

Modificar/añadir agente desde línea de comandos:

```
haconf -makerw
hares -add NIC_eth[0-9] NIC NIC_Grupo_<nombre_grupo>_eth[0-9]
hares -modify NIC_eth[0-9]0 Enabled 1
hares -modify NIC_eth[0-9]0 Mii "0"
hares -local NIC_eth[0-9] Device
hares -modify NIC_eth[0-9] Device "eth[0-9]" -sys HOSTNAME1
hares -modify NIC_eth[0-9] Device "eth[0-9]" -sys HOSTNAME2
hares -modify NIC_eth[0-9] NetworkHosts "XX.XX.XX.1"
hares -modify NIC_eth[0-9] PingOptimize "1"
hares -modify NIC_eth[0-9] Critical 1
haconf -dump -makero
```

10.2.6. Agente NFS

Si queremos manejar un servidor NFS desde VCS, es necesario usar más de un agente, en concreto tres; **NFS**, **NFSRestart** y **Share**.

Están definidos por defecto en types.cf, por lo que no es necesario instalar nada adicional.

La combinación de los tres agentes hará posible levantar un servidor de NFS en uno o varios de los nodos de VCS.

10.2.6.1. Agente NFS principal

Es el agente que se encarga de levantar el demonio de NFS. Su configuración mínima es muy simple, bastando con declararlo para que el demonio levante.

Un ejemplo de utilización del agente sería:

```
NFS NFSdaemon (
        Nproc = 16
        )
```

Donde **Nproc** son las peticiones concurrentes que el servidor es capáz de atender. Por defecto son 8 (levanta 8 procesos o demonios).

10.2.6.2. Agente NFSRestart

Este agente, aunque no es obligatorio usarlo, es sumamente útil, ya que se encarga de manejar el proceso **lock recovery** el cual cosiste en recuperar **record locks** antes de un reinicio o una caía del servido no planificado, evitando corrupción de archivos tanto en la parte cliente como en la parte servidor. Además proporciona alta disponibilidad para los record locks en caso de balanceo del servicio entre nodos.

El agente arranca, para y monitoriza tres demonios: smsyncd, statd, and lockd. Si se configura el agente NFSRestart para usar lock recovery, el agente NFSRestart arranca el demonio smsyncd. Este demonio copia los “locks” de NFS del/los FS que se exportan al archivo local /var/lib/nfs y viceversa.

Un ejemplo de utilización del agente sería:

```
NFSRestart NFSrestart (
        NFSRes = NFSdaemon
        NFSLockFailover = 1
        LocksPathName = "/pruebaCFS"
        )
```

Donde:

- NFSRes. Agente NFS.
- NFSLockFailover. Valor booleano. Con un 1, activamos lo descrito anteriormente.
- LocksPathName. Path con alta disponibilidad sobre el que se volcará la información de los "locks" de NFS.

10.2.6.3. Agente Share

Este agente define los recursos que se desean exportar. Comparte, des-comparte y monitoriza los recursos definidos como exportados. Este agente sólo debe exportar archivos y directorios que están en recursos de alta disponibilidad.

Un ejemplo de utilización del agente sería:

```
Share export_pruebaCFS (
        Options = "-o rw"
        PathName = "/pruebaCFS"
        Client = VCS2 .tfc.es
        )
```

Donde:

- PathName. Atributo obligatorio. Recurso a exportar.
- Options. Opciones del export. Los referidos en el man de exportfs.
- Client. Clientes a los que se permite importar el recurso

10.2.6.4. Ejemplos de utilización del agente NFS

En el path **/etc/VRTSvcs/conf/sample_nfs/** se pueden encontrar ejemplos de utilización de los tres agentes.

10.2.7. Agente APPLICATION

A continuación vemos un ejemplo básico de configuración del agente:

```
Application <aplicacion> (
        User = <usuario_aplicacion>
        MonitorProgram = "/usr/local/bin/monitor_APLICACION.sh"
        PidFiles = { "/<path>/<aplicacion>.pid" }
        MonitorProcesses = { "<aplicacion>" }
        StartProgram = "/usr/local/bin/APLICACION start"
        StopProgram = "/usr/local/bin/APLICACION stop"
        )
```

Dependencias obligatorias: No depende de nada.

En lo que concierne a la monitorización, se puede usar cualquiera de los tres modos (o combinación de ellos): MonitorProgram, PidFiles o MonitorProcesses.

MonitorProgram. Script implementado por nosotros, que devuelve un 110 si está ok el proceso o un 100 en caso de estar caído.

El contenido del archivo de monitorización debe devolver un código de error 110 si el proceso está corriendo y un 100 si no lo está.

```
USER="usuario_aplicacion"
PROCMS="proceso_a_monitorizar"
```

```
if [ $( ps -u ${USER} -o args | grep "${PROCMS}" | grep -vc grep ) -gt 0 ]
        then
        # ONLINE
        exit 110
fi

# OFFLINE
exit 100
```

Modificar/añadir agente desde línea de comandos:

```
haconf -makerw
hares -add <aplicacion> Application Grupo_<nombre_grupo>
hares -modify <aplicacion> Enabled 1
hares -modify <aplicacion> User <usuario_aplicacion>
hares -modify <aplicacion> MonitorProgram "/usr/local/bin/monitor_APLICACION.sh"
hares -modify <aplicacion> StartProgram "/usr/local/bin/APLICACION start"
hares -modify <aplicacion> StopProgram "/usr/local/bin/APLICACION stop"
haconf -dump -makero
```

NOTA: El archivo de monitorización así como los scripts de arranque/parada deben de existir antes de crear el agente.

10.2.8. Agente de VOM

Comprobar que tenemos instalado el agente de VOM en la máquina (se instala por defecto):
```
rpm -qa | grep VRTSsfmh
VRTSsfmh-4.1.119.0-0
```

Para dar de alta el cluster y los nodos en el VOM, acceder a la VOM por consola web y proceder de la sigueinte manera:

Acceder con el usuario root, passwd <elegido al instalar VOM>.

Desde la consola web, acceder a “Settings” y pulsar sobre “Host Management”. Añadir los nuevos host del nuevo cluster. El usuario y la passwd es la del usuario root en los nuevos nodos.

10.2.9. Agente JBOSS

Debemos instalar los siguientes paquetes:

VRTSjboss-5.1.0.0-GA_GENERIC_noarch
VRTSappab-5.1.8.0-GA_GENERIC_noarch

Una vez instalados, nos desplazaremos al path:
```
/opt/VRTSagents/ha/bin/AgentBuilder
```

Ejecutar:
```
./agentbuilder JBoss -base vcs60 -platform linux -ssh -system <NODO1> -system <NODO2>
```

En el path /etc/VRTSvcs/conf/config se genera un archivo de configuración de VCS JbossTypes.cf. En este archivo está la definición del agente de Jboss. Debemos hacer un “include” de este archivo en el main.cf.

Un ejemplo de uso de este agente sería:
```
include  JbossTypes.cf
.
.
```

.
```
JBoss Jboss (
        StartProgram = "/prueba/jboss/bin/run.sh"
        AgentDirectory = "/opt/VRTSagents/ha/bin/JBoss"
        AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
        )
```

10.2.10. Agente APACHE

Este agente, viene ya agregado a la definición de agentes por defecto. Lo podemos comprobar, yendo a la sección "Apache" dentro del archivo **/etc/VRTSvcs/conf/config/types.cf**

A continuación vemos una configuración típica del agente, agregada al archivo
/etc/VRTSvcs/conf/config/types.cf
```
Apache Apache (
        PidFile = "/var/run/httpd2.pid"
        IntentionalOffline = 1
        httpdDir = "/usr/sbin/httpd2prefork"
        ConfigFile = "/etc/apache2/httpd.conf"
        ResLogLevel = [INFO|TRACE|DEBUG]
        )
```

Dependencias obligatorias: No depende de nada.

Modificar/añadir agente desde línea de comandos:
```
haconf -makerw
hares -add Apache Apache Grupo_<nombre_grupo>
hares -modify Apache Enabled 1
hares -modify Apache PidFile "/var/run/httpd2.pid"
hares -modify Apache httpdDir "/usr/sbin/httpd2prefork"
hares -modify Apache ConfigFile "/etc/apache2/httpd.conf"
hares -modify Apache ResLogLevel [INFO|TRACE|DEBUG]
```

10.2.11. Agente de BBDD MySQL

Para el TFC se ha decidido la utilización de MySQL por ser open source. Los atributos para la configuración del agente de MySQL son los siguientes:

Table 1-2 Sample MySQL server resource attributes

Attribute	Value
MonitorProcessPatterns	"/usr/local/mysql/bin/mysql --basedir=/usr/local/mysql" "/bin/sh/usr/local/mysql/bin/mysqld_safe --user mysql"
MonitorSequence	MonitorProcessPatterns PidFilesPatterns ListenAddressPort MonitorProgram
ResLogLevel	INFO
StartProgram	/usr/local/mysql/bin/mysql_safe --user mysql&
StopProgram	/usr/local/mysql/bin/mysql_stop

```
type MySQL (
static int ToleranceLimit = 1
static boolean AEPTimeout = 1
static str AgentFile = "/opt/VRTSvcs/bin/Script50Agent"
static str AgentDirectory = "/opt/VRTSagents/ha/bin/MySQL"
static str ArgList[] = { ResLogLevel, State, IState, MySQLUser,
MySQLAdmin, MySQLAdminPasswd, EnvFile, BaseDir, DataDir, MyCnf,
HostName, Port, SecondLevelMonitor, MonitorProgram }
str ResLogLevel = INFO
str MySQLUser = mysql
```

```
str MySQLAdmin = root
str MySQLAdminPasswd
str EnvFile
str BaseDir
str DataDir
str MyCnf
str HostName
int Port = 3306
int SecondLevelMonitor = 0
str MonitorProgram
static int IMF{} = { Mode=2, MonitorFreq=5, RegisterRetryLimit=3 }
static str IMFRegList[] = { BaseDir DataDir MySQLUser }
)
```

A continuación vemos una configuración típica del agente, agregada al archivo /etc/VRTSvcs/conf/config/main.cf

```
MySQL mysql (
Critical = 0
MySQLAdminPasswd = iwlWruVujUwwMunUl
BaseDir = "/opt/mysql/mysql"
DataDir = "/var/lib/mysql"
HostName = "vcssx074.vxindia.veritas.com"
)
```

10.2.12. Agente de BBDD ORACLE

En cambio, lo normal en proyectos empresariales medios es la utilización de bases de datos Oracle y la configuración para este tipo de BBDD sera como sigue:

Para poder usar el agente de Oracle, debemos incluir en la configuración de Veritas la definición del agente. Para ello, haremos un "include" de **OracleTypes.cf** en el archivo principal de configuración del cluster main.cf.

```
include "OracleTypes.cf"
```

A continuación vemos una configuración típica del agente, agregada al archivo /etc/VRTSvcs/conf/config/types.cf

```
Netlsnr ListenerOra10 (
        Owner = ora92
        Home = "/ora92/app/oracle/product/10"
        Listener = LISTENER
        )

Oracle Oracle10 (
        Sid = PACVE11
        Owner = ora92
        Home = "/ora92/app/oracle/product/10"
        DetailMonitor = 1
        User = veritas
        Pword = aogMfoCmjOemBod
        Table = test
        )
```

Dentro del recurso Oracle, el campo Pword tiene que tener la password encriptada. La password se encripta con el comando:
vcsencrypt agent <password del usuario veritas en Oracle>

10.2.12.1. Funciones del agente ORACLE

El agente de Oracle se encarga de los procesos de la base de datos. Entre las operaciones que realiza el agente están:

- **Online**. Inicia la BBDD usando el comando svrmngrl o sqlplus con "startup force pfile=$PFile". El

arranque por defecto es “STARTUP_FORCE”, se puede configurar para distintos tipos de arranque.

- **Offline**. Para la BBDD usando el comando svrmngrl o sqlplus con “shutdown immediate” Inmediate es la opcion por defecto, se puede modificar para parar la BBDD de distintas maneras.
- **Monitor**. Comprueba el estado de la BBDD, se puede usar en 2 modos: basico y detallado.
- **Clean**. Fuerza el apagado de la base de datos con “shutdown abort”, en caso de que la BBDD no pare ejecuta lo siguiente:
    - Busca en la table de procesos los relacionados con la instacia.
    - Mata los procesos.
    - **Info**. Da información dinamica y estatica de la BBDD.
    - **Action**. Ejecuta una acción predefinida en la BBDD.

10.2.12.2. Arraque y parada

Las opciones de arranque son:

- STARTUP_FORCE (Default). Ejecuta el comando “startup_force pfile='pfile'” si el pfile esta definido, en caso contrario lo ejecuta con el valor por defecto.
- STARTUP. Ejecuta el comando “startup pfile='pfile'” si esta configurado en caso contrario lo ejecuta con loso valores por defecto.
- RESTRICTED. Arranca la BBDD en modo restricted.
- RECOVERDB.Ejecuta un recovery de la BBDD.
- CUSTOM. Usa un script sql predefinido (start_custom_$SID.sql), el script tiene que estar en “/opt/VRTSagents/ha/bin/Oracle”.
- SRVCTLSTART. Usa srvctlstart para arranar la BBDD. En el caso de Oracle RAC deben definirse manualmente las opciones.

Las opciones de parada son:

- IMMEDIATE (Default). Ejecuta “shutdown inmediate”
- TRANSACTIONAL.
- CUSTOM. Ejecuta un script sql de parada: shut_custom_$SID.sql.
- SRVCTLSTOP.

10.2.12.3. Monitorización del agente

El agente de Oracle ofrece 2 niveles de monitorización, básico y detallado.

- **Basico**. Permite 2 tipos de monitorización

  Process check
  Health check

Para definir el tipo de monitorización usamos el parametro MonitorOption, y puede tener los siguientes valores:

0 (Default). Process check. El agente monitoriza la tabla de procesos en busca de ora_dbw, ora_smon, ora_pmon y ora_lgwr para comprobar que Oracle esta corriendo.

1 Health check. Solo esta soportado a partir de Oracle 10g. Utiliza las APIs de health check de Oracle para monitorizar la SGA y recibir información de la instacia.

- **Detallado**. El agente realiza una transaccion en una tabla de pruebas para comprobar que esta funcionando correctamente la BBDD. Utiliza el atributo “DetailMonitor” para definir monitorización detallada. La monitorización se realiza mediante los scripts (SqlTest.pl y SimpleTest.pl) de “/opt/VRTSagents/ha/bin/Oracle/”. Ambos actualizan el timestamp de una tabla de test para monitorizar la BBDD:

SqlTest.pl. Script que comprueba si la BBDD esta abierta antes de actualizar el timestamp. Si la tabla esta en

modo restricted, quiesced o suspended devuelve un acierto de manera que solo se realiza la monitorización basica.

SimpleTest.pl. No chequea la BBDD, simplemente actualiza el timestamp.

Antes de activar la monitorización detallada debemos crear una tabla de test y dar permiso de ejecución al script de monitorización, podemos usar un script personalizado o uno de los scripts proporcionados por el agente. Si el script devuelve un 100 es que ha fallado, un codigo de retorno de 101 a 110 indica que ha ido bien. Necesitamos también un usuario para ejecutar los comandos.

```
connect / as sysdba
create user  <User>
identified by <Pword>
default tablespace USERS
temporary tablespace TEMP
quota 100K on USERS;
grant create  session to <User>;
create table <User>.<Table> (  tstamp date );
insert into <User>.<Table> (tstamp) values  (SYSDATE);
```

10.2.12.4. Control de errores

El agente puede manejar errores de Oracle en el estado de monitorización detallada, clasificando los mismos según gravedad y actuando según acciones predeterminadas.

El agente incluye una referencia a oraerror.dat (/opt/VRTSagents/ha/bin/Oracle/oraerror.dat) que lista los posibles errores de Oracle y la accion a tomar en caso de encontrarnos un error. La información en el archivo se almacena en el siguiente formato:

Oracle_error_string:action_to_be_taken

Por ejemplo:

01035:WARN
01034:FAILOVER

La lista de acciones predefinidas a realizar:

- **IGNORE**. Cuando detectamos un error que esta en el archivo de oraerror relaizamos la acción indicada, en caso de no encontrarlo lo ignoramos.
- **UNKNOWN**. Marcamos el recurso como estado Unknown y mandamos una notificación.
- **WARN**. Marcamos el recurso como estado Online y mandamos una notificación.
- **FAILOVER** (DEFAULT). Marcamos el recurso como estado Offline, esto provoca un fallo en el grupo de recursos y hace que balancee al siguiente nodo disponible.
- **NOFAILOVER**. Congela el grupo de recursos y lo marca como Offline.

10.2.13. Agente CFSMOUNT

Este agente se encarga de montar un FS del tipo VCFS en todos los nodos del cluster.

A continuación vemos un ejemplo básico de configuración del agente:

```
CFSMount Montaje_CFS_<NombreLV> (
Critical = 0
MountPoint = "<MontajeLV>"
BlockDevice = "/dev/vx/dsk/<NombreDG>/<NombreLV>"
MountOpt @HOSTNAME1 = "cluster"
MountOpt @HOSTNAME2 = "cluster"
NodeList = { HOSTNAME1, HOSTNAME2 }
)
```

Dependencias obligatorias:

Montaje_CFS_<NombreLV> requires Shared_DG_<NombreDG>

Modificar/añadir agente desde línea de comandos:

```
haconf -makerw
hares -add  Montaje_CFS_<NombreLV> CFSMount Grupo_<nombre_grupo>
hares -modify Montaje_CFS_<NombreLV> Enabled 1
hares -modify Montaje_CFS_<NombreLV>
hares -modify Montaje_CFS_<NombreLV>
```

10.3. Configurar cluster como un solo nodo

Si por alguna razón nos falla un nodo del cluster podemos convertirlo en un cluster con un solo nodo, primero debemos desactivar el arranque de LLT y GAB.

En ambos nodos:

```
mv /etc/init.d/rc3.d/S06llt /etc/init.d/rc3.d/X06llt
mv /etc/init.d/rc5.d/S06llt /etc/init.d/rc5.d/X06llt
mv /etc/init.d/rc5.d/S07gab /etc/init.d/rc5.d/X07gab
mv /etc/init.d/rc5.d/S07gab /etc/init.d/rc5.d/X07gab
```

Modificamos la configuración de Veritas (main.cf) con Veritas parado para quitar todas las referencias a la maquina eliminada y hacemos el verify y generate de la configuración. Hacer un backup de la configuración antes de modificarla es recomendable para volver a incluir el nodo que estamos eliminando.

En el nodo principal:

```
hastop -all
cd /etc/VRTSvcs/conf/config/
cp -a main.cf main.cf-FECHA-backup

 <Modificar main.cf>

hacf -verify /etc/VRTSvcs/conf/config
hacf -generate /etc/VRTSvcs/conf/config
```

Arrancamos el cluster indicando que arranque como un solo nodo.

```
hastart -onenode
```

10.3.1. Recuperar la configuración MULTINODO

Para volver al estado anterior basta con reactivar los servicios y recuperar la configuración anterior. Es recomendable reiniciar los nodos.

En ambos nodos:

```
mv /etc/init.d/rc3.d/X06llt /etc/init.d/rc3.d/S06llt
mv /etc/init.d/rc5.d/X06llt /etc/init.d/rc5.d/S06llt
mv /etc/init.d/rc5.d/X07gab /etc/init.d/rc5.d/S07gab
mv /etc/init.d/rc5.d/X07gab /etc/init.d/rc5.d/S07gab
```

En el nodo principal:

```
cp main.cf-FECHA-backup main.cf
hacf -verify /etc/VRTSvcs/conf/config
hacf -generate /etc/VRTSvcs/conf/config
```

En ambos nodos:

```
/etc/init.d/llt start
/etc/init.d/gab start
hastart
```

En caso de algún problema puede ser más rápido reiniciar ambos nodos.

11. LICENCIAS VERITAS

Los comandos para poder comprobar las licencias de Veritas son:

- vxlicinst Instala una licencia para un producto de Symantec
- vxlicrep Muestra las licencias actuales
- vxlictest Muestra novedades y sus descripciones codificadas en una clave de licencia

11.1. Comprobar Licencias

Para comprobar las licencias que tenemos instaladas usaremos el comando vxlicrep, que nos mostrara información de todas las licencias que tengamos instaladas en el sistema.

```
vxlicrep
```

Las licencias autogeneradas en la instalación, son las que tienen el atributo keyless. Este tipo de licencia permite operar e instalar los productos de Veritas sin problemas, pero a la hora de solicitar soporte, no son válidos y es necesario instalar una licencia válida.

La forma de ver si tenemos licencias keyless habilitadas es:

```
vxkeyless -v display
```

Para deshabilitar todas las licencias keyless:

```
vxkeyless set NONE
```

11.2. Actualizar licencias

El comando para instalar licencias es:

```
vxlicinst -k XXXX-XXXX-XXXX-XXXX-XXXX-XXX
```

Con esto las licencias ya deberían aparecer actualizadas, lo podemos comprobar ejecutando de nuevo vxlicrep.

12. PRUEBAS DE FUNCIONAMIENTO Y RENDIMIENTO

El cluster esta configurado para dos nodos VCS1 y VCS2. Como vemos en la imagen:

```
VCS1:~ # hastatus -sum

-- SYSTEM STATE
-- System               State                Frozen

A  VCS1                 RUNNING              0
A  VCS2                 RUNNING              0

-- GROUP STATE
-- Group           System               Probed     AutoDisabled    State

B  Pruebas         VCS1                 Y          N               ONLINE
B  Pruebas         VCS2                 Y          N               OFFLINE
VCS1:~ #
```

En cluster se a configurado un grupo de recursos llamado Pruebas, en que hemos configurado lo siguiente:

```
VCS1:~ # hares -state
#Resource        Attribute             System     Value
Apache           State                 VCS1       ONLINE
Apache           State                 VCS2       OFFLINE
DG_shareddg      State                 VCS1       ONLINE
DG_shareddg      State                 VCS2       OFFLINE
Jboss            State                 VCS1       ONLINE
Jboss            State                 VCS2       OFFLINE
Montaje_NFS_lv   State                 VCS1       ONLINE
Montaje_NFS_lv   State                 VCS2       OFFLINE
Montaje_jboss_lv State                 VCS1       ONLINE
Montaje_jboss_lv State                 VCS2       OFFLINE
Montaje_mysql_lv State                 VCS1       ONLINE
Montaje_mysql_lv State                 VCS2       OFFLINE
MySQL            State                 VCS1       ONLINE
MySQL            State                 VCS2       OFFLINE
NFSServer        State                 VCS1       ONLINE
NFSServer        State                 VCS2       OFFLINE
VIP              State                 VCS1       ONLINE
VIP              State                 VCS2       OFFLINE
```

Se aprecia que todos los servicios están corriendo en el nodo VCS1. Si provocamos un apagado del servidor VCS1, veremos como los recursos se balancean al node VCS2:

Nos vamos al nodo VCS2 para hacer las comprobaciones pertinentes:

```
VCS2:~ # hastatus -sum

-- SYSTEM STATE
-- System               State                Frozen

A  VCS1                 EXITED               0
A  VCS2                 RUNNING              0

-- GROUP STATE
-- Group           System               Probed     AutoDisabled    State

B  Pruebas         VCS1                 Y          N               OFFLINE
B  Pruebas         VCS2                 Y          N               ONLINE
VCS2:~ # hares -state
#Resource        Attribute             System     Value
Apache           State                 VCS1       OFFLINE
Apache           State                 VCS2       ONLINE
DG_shareddg      State                 VCS1       OFFLINE
DG_shareddg      State                 VCS2       ONLINE
Jboss            State                 VCS1       OFFLINE
Jboss            State                 VCS2       ONLINE
Montaje_NFS_lv   State                 VCS1       OFFLINE
Montaje_NFS_lv   State                 VCS2       ONLINE
Montaje_jboss_lv State                 VCS1       OFFLINE
Montaje_jboss_lv State                 VCS2       ONLINE
Montaje_mysql_lv State                 VCS1       OFFLINE
Montaje_mysql_lv State                 VCS2       ONLINE
MySQL            State                 VCS1       OFFLINE
MySQL            State                 VCS2       ONLINE
NFSServer        State                 VCS1       OFFLINE
NFSServer        State                 VCS2       ONLINE
VIP              State                 VCS1       OFFLINE
VIP              State                 VCS2       ONLINE
VCS2:~ #
```

Observamos que todo el grupo Pruebas, esta corriendo correctamente en el nodo VCS2 y que no hemos perdido servicio de la aplicación, salvo el tiempo que el cluster tarda en balancear.

Consultamos los logs del VCS para ver de cuanto fue la perdida de servicio:

```
==============================================
Shutting down service MySQL ..done
==============================================

2013/12/21 13:50:43 VCS INFO V-16-1-10305 Resource MySQL (Owner: Unspecified, Group: Pruebas) is offline on VCS1 (VCS initiated)
2013/12/21 13:50:43 VCS NOTICE V-16-1-10300 Initiating Offline of Resource VIP (Owner: Unspecified, Group: Pruebas) on System VCS1
2013/12/21 13:50:44 VCS INFO V-16-1-10305 Resource VIP (Owner: Unspecified, Group: Pruebas) is offline on VCS1 (VCS initiated)
2013/12/21 13:50:44 VCS NOTICE V-16-1-10300 Initiating Offline of Resource Montaje_NFS_lv (Owner: Unspecified, Group: Pruebas) on System VCS1
2013/12/21 13:50:44 VCS NOTICE V-16-1-10300 Initiating Offline of Resource Montaje_jboss_lv (Owner: Unspecified, Group: Pruebas) on System VCS1
2013/12/21 13:50:44 VCS NOTICE V-16-1-10300 Initiating Offline of Resource Montaje_mysql_lv (Owner: Unspecified, Group: Pruebas) on System VCS1
2013/12/21 13:50:47 VCS INFO V-16-1-10305 Resource Montaje_jboss_lv (Owner: Unspecified, Group: Pruebas) is offline on VCS1 (VCS initiated)
2013/12/21 13:50:48 VCS INFO V-16-1-10305 Resource Montaje_NFS_lv (Owner: Unspecified, Group: Pruebas) is offline on VCS1 (VCS initiated)
2013/12/21 13:50:48 VCS INFO V-16-1-10305 Resource Montaje_mysql_lv (Owner: Unspecified, Group: Pruebas) is offline on VCS1 (VCS initiated)
2013/12/21 13:50:48 VCS NOTICE V-16-1-10300 Initiating Offline of Resource DG_shareddg (Owner: Unspecified, Group: Pruebas) on System VCS1
2013/12/21 13:50:49 VCS INFO V-16-2-13717 (VCS1) Output of the completed operation (imf_getnotification)
```

Fijándonos, por ejemplo, en el comportamiento de la base de datos MySQL en el balanceo de un nodo a otro, vemos que la caída del nodo VCS1 se detecto a las 13:50:43.

Ahora nos fijamos en el log para ver cuando arranco la BBDD en el nodo VCS2:

```
==============================================
Starting service MySQL ..done
==============================================

2013/12/21 13:51:21 VCS INFO V-16-1-10298 Resource MySQL (Owner: Unspecified, Group: Pruebas) is online on VCS2 (VCS initiated)
2013/12/21 13:51:21 VCS NOTICE V-16-1-10301 Initiating Online of Resource Jboss (Owner: Unspecified, Group: Pruebas) on System VCS2
2013/12/21 13:51:22 VCS INFO V-16-10031-504 (VCS2) Application:Jboss:online:Executed /etc/init.d/jboss_pruebas as user root
2013/12/21 13:51:22 VCS INFO V-16-2-13716 (VCS2) Resource(Jboss): Output of the completed operation (online)
```

Y podemos comprobar que termino de reiniciarse en el nodo VCS2 a las 13:51:22.

Por lo tanto, podemos concluir que entre la caída del nodo 1 y el balanceo de todo el aplicativo al nodo 2 pasan 39 segundos.

Vamos a realizar otra prueba provocando la caída del servidor apache en el nodo principal. Para ello ejecutamos un kill al proceso padre del apache en el nodo activo:

```
VCS1:/srv/www/htdocs # hastatus -sum

-- SYSTEM STATE
-- System               State                Frozen

A  VCS1                 RUNNING              0
A  VCS2                 RUNNING              0

-- GROUP STATE
-- Group           System               Probed     AutoDisabled    State

B  Pruebas         VCS1                 Y          N               ONLINE
B  Pruebas         VCS2                 Y          N               OFFLINE
VCS1:/srv/www/htdocs # ps -ef | grep -i http
root     18547     1  0 13:44 ?        00:00:00 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -k start
wwwrun   18549 18547  0 13:44 ?        00:00:00 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -k start
wwwrun   18550 18547  0 13:44 ?        00:00:00 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -k start
wwwrun   18551 18547  0 13:44 ?        00:00:00 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -k start
wwwrun   18552 18547  0 13:44 ?        00:00:00 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -k start
wwwrun   18553 18547  0 13:44 ?        00:00:00 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -k start
wwwrun   19031 18547  0 13:45 ?        00:00:00 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf -k start
root     19154  7791  0 13:46 pts/0    00:00:00 grep -i http
VCS1:/srv/www/htdocs # kill -9 18547
VCS1:/srv/www/htdocs # /etc/init.d/apache2 status
Checking for httpd2:                                                    dead
```

Comprobamos por navegador el acceso a la aplicación y vemos que no responde.

Y vemos el estado del grupo de recursos pruebas en el nodo 1 del cluster, comprobando que esta en fallo.

```
VCS1:/srv/www/htdocs # ps -ef | grep -i http
root     19862  7791  0 13:48 pts/0    00:00:00 grep -i http
VCS1:/srv/www/htdocs # hares -state
#Resource         Attribute              System     Value
Apache            State                  VCS1       FAULTED
Apache            State                  VCS2       OFFLINE
DG_shareddg       State                  VCS1       ONLINE
DG_shareddg       State                  VCS2       OFFLINE
Jboss             State                  VCS1       OFFLINE
Jboss             State                  VCS2       OFFLINE
Montaje_NFS_lv    State                  VCS1       ONLINE
Montaje_NFS_lv    State                  VCS2       OFFLINE
Montaje_jboss_lv  State                  VCS1       OFFLINE
Montaje_jboss_lv  State                  VCS2       OFFLINE
Montaje_mysql_lv  State                  VCS1       OFFLINE
Montaje_mysql_lv  State                  VCS2       OFFLINE
MySQL             State                  VCS1       OFFLINE
MySQL             State                  VCS2       OFFLINE
NFSServer         State                  VCS1       ONLINE
NFSServer         State                  VCS2       OFFLINE
VIP               State                  VCS1       OFFLINE
VIP               State                  VCS2       OFFLINE
```

VCS lo detecta y automáticamente procede a reiniciar todo el grupo correctamente como vemos en el detalle:

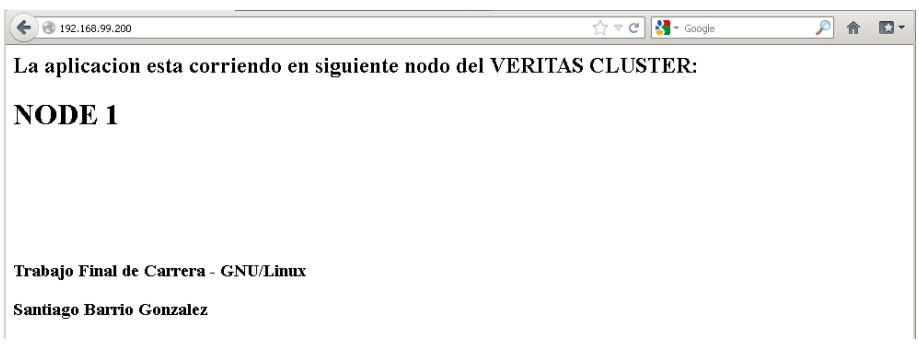
```
VCS1:/srv/www/htdocs # hastatus
attempting to connect....
attempting to connect....connected

group           resource        system          message
--------------- --------------- --------------- -------------------
                                VCS1            RUNNING
                                VCS2            RUNNING
Pruebas                         VCS1            STOPPING PARTIAL *FAULTED*
Pruebas                         VCS2            OFFLINE
-------------------------------------------------------------------------
                Apache          VCS1            *FAULTED*
                Apache          VCS2            OFFLINE
                Jboss           VCS1            ONLINE
                Jboss           VCS2            OFFLINE
                Jboss           VCS1            WAITING FOR OFFLINE
-------------------------------------------------------------------------
```

Finaliza el arranque cuando observamos el online en el recurso Apache.

```
group           resource        system          message
--------------- --------------- --------------- -------------------
                VIP             VCS1            ONLINE
                MySQL           VCS1            WAITING FOR ONLINE
                MySQL           VCS1            ONLINE
                Jboss           VCS1            WAITING FOR ONLINE
-------------------------------------------------------------------------
                Jboss           VCS1            ONLINE
                Apache          VCS1            WAITING FOR ONLINE
                Apache          VCS1            ONLINE
Pruebas                         VCS1            ONLINE
```

Ahora si nos vamos al navegador podremos comprobar que el aplicativo vuelve a dar página:



13. ANEXO. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

VCS	Veritas Cluster Server
VCFS	Veritas Cluster Filesystem Server
FS	Filesystem
DG	Disk Group
VSFCFS	Veritas Storage Foundation Cluster File System
VxVM	Veritas Volumen Manager
VSFHA	Veritas Storage Foundation High availability
LVM	Logical Volume Manager
CRG	Cluster Resource Group (grupo de recursos de cluster)
MV	Maquina Virtual
DHCP	Dynamic Host Configuration Protocol
VOM	Veritas Operations Manager
GNU	Licencia de documentación libre GNU para realizar software de libre uso
GUI	(Graphical User Interface, GUI) Interfaz Gráfica de Usuario
TFC	Trabajo final de carrera

14. ANEXO. ARCHIVOS DE CONFIGURACIÓN DE TFC

```
VCS2:~ # cat /etc/VRTSvcs/conf/config/main.cf
include "OracleASMTypes.cf"
include "types.cf"
include "Db2udbTypes.cf"
include "OracleTypes.cf"
include "SybaseTypes.cf"

cluster SAN_CLUSTER (
        UserNames = { admin = dKLdKFkHLgLLjTLfKI }
        EngineShutdown = PromptClusStop
        Administrators = { admin }
        )

system VCS1 (
        )

system VCS2 (
        )

group Pruebas (
        SystemList = { VCS1 = 0, VCS2 = 1 }
        AutoStartList = { VCS1, VCS2 }
        OnlineRetryLimit = 3
        OnlineRetryInterval = 120
        )
        Apache Apache (
                httpdDir = "/usr/sbin/httpd2-prefork"
                PidFile = "/var/run/httpd2.pid"
                ConfigFile = "/etc/apache2/httpd.conf"
                )

        Application Jboss (
                StartProgram = "/etc/init.d/jboss_pruebas start"
                StopProgram = "/etc/init.d/jboss_pruebas stop"
                MonitorProgram = "/usr/local/bin/monitorJboss"
                )

        Application MySQL (
                StartProgram = "/etc/init.d/mysql start"
                StopProgram = "/etc/init.d/mysql stop"
                MonitorProgram = "/usr/local/bin/monitorMySQL"
                )

        Application NFSServer (
                StartProgram = "/etc/init.d/nfsserver start"
                StopProgram = "/etc/init.d/nfsserver stop"
                MonitorProgram = "/usr/local/bin/monitorNFSServer"
                )

        DiskGroup DG_shareddg (
                DiskGroup = shareddg
                )

        IP VIP (
                Device = eth0
                Address = "192.168.99.200"
                NetMask = "255.255.255.0"
                )

        Mount Montaje_NFS_lv (
                MountPoint = "/compartido"
                BlockDevice = "/dev/vx/dsk/shareddg/compartido_lv"
                FSType = vxfs
```

```
        FsckOpt = "-y"
        )

Mount Montaje_jboss_lv (
        MountPoint = "/opt/jboss-5.0.0.GA"
        BlockDevice = "/dev/vx/dsk/shareddg/jboss_lv"
        FSType = vxfs
        FsckOpt = "-y"
        )

Mount Montaje_mysql_lv (
        MountPoint = "/opt/mysql"
        BlockDevice = "/dev/vx/dsk/shareddg/mysql_lv"
        FSType = vxfs
        FsckOpt = "-y"
        )

Apache requires Jboss
Jboss requires Montaje_jboss_lv
Jboss requires MySQL
Montaje_NFS_lv requires DG_shareddg
Montaje_jboss_lv requires DG_shareddg
Montaje_mysql_lv requires DG_shareddg
MySQL requires VIP
NFSServer requires Montaje_NFS_lv
VIP requires Montaje_NFS_lv
VIP requires Montaje_jboss_lv
VIP requires Montaje_mysql_lv

// resource dependency tree
//      group Pruebas
//      {
//      Apache Apache
//          {
//          Application Jboss
//              {
//              Mount Montaje_jboss_lv
//                  {
//                  DiskGroup DG_shareddg
//                  }
//              Application MySQL
//                  {
//                  IP VIP
//                      {
//                      Mount Montaje_NFS_lv
//                          {
//                          DiskGroup DG_shareddg
//                          }
//                      Mount Montaje_jboss_lv
//                          {
//                          DiskGroup DG_shareddg
//                          }
//                      Mount Montaje_mysql_lv
//                          {
//                          DiskGroup DG_shareddg
//                          }
//                      }
//                  }
//              }
//          }
//      Application NFSServer
//          {
//          Mount Montaje_NFS_lv
//              {
//              DiskGroup DG_shareddg
//              }
//          }
//      }
```

15. BIBLIOGRAFIA

15.1 Listado de libros utilizados

La bibliografía consultada para la realización del proyecto es la siguiente:

High Availability: Design, Techniques, and Processes
Floyd Piedad; Michael Hawkins
Editorial: Prentice Hall
ISBN: 0130962880

In Search Of Clusters
Gregory F. Pfister
Editorial: Prentice Hall
ISBN: 0138997098

Availability Management: Planning and Implementing Cross-Site Mirroring on IBM System i5
Nick Harris, Ted Bauer, Doug Bidwell, Dan Degroff, Mike Halda, Sabine Jordan, John Schrum
Editorial: IBM
ISBN: 0738489905

Clustering and IASPs for Higher Availability on the IBM eServer iSeries Server
Susan Powers, Ellen Dreyer Andersen, Sue Nee, David Salmon, Shashi Sethi, Lee Walkky
Editorial: IBM
ISBN: 0738422355

Hardware Management Console V7 Handbook
Stephen Hochstetler, JunHeum Min, Matt Robbins, Nancy Milliner, Narend Chand, Syamsul Hidayat
Editorial: IBM
ISBN: 0738486507

Shared Data Clusters: Scaleable, Manageable, and Highly Available Systems (VERITAS Series)
Dilip M. Ranade
Librería: Free State Books (Halethorpe, MD, U.S.A.)
ISBN: 047118070X / 0-471-18070-X

Computercluster: Grid-Computing, Mapreduce, Rechnerverbund, Projekt Athena, Oracle Rac, Hochverf Gbarkeit, Drbd, Veritas Cluster Server
Bücher Gruppe
Librería: ABC Books (Lowfield Heath, CRAWL, United Kingdom)
ISBN: 1158788940 / 1-158-78894-0

Administration of Veritas Cluster Server Secrets to Acing the Exam and Successful Finding and Landing
Stephanie Barlow
Librería: wordery (Norwich, NFLK, United Kingdom)
ISBN: 9781486157242

Veritas Storage Foundation
Volker Herminghaus
Librería: Rhein-Team Lörrach Ivano Narducci e.K. (Lörrach, BW, Germany)
ISBN: 9783540346104

Symantec Operations Readiness Tools
Librería: BuySomeBooks (Las Vegas, NV, U.S.A.)
ISBN: 9786136276496

15.2 Listado de WEB`S utilizadas

- Web del fabricante y enlace sobre el producto VCS http://www.symantec.com/es/es/cluster-server
- Documentación sobre Linux SUSE https://www.suse.com/es-es/documentation/sles11/singlehtml/book_sle_deployment/
- Detalles sobre requisitos para la instalación o actualización de VCS https://sort.symantec.com/checklist/install#report
- Resumen comercial de VCS
- http://www.symantec.com/content/en/us/enterprise/fact_sheets/b-veritas_cluster_server_DS_21213909.en-us.pdf http://www.nextvision.com/uploads/com_solucion/Symantec_Cluster_Server.pdf
- Sitio con orientaciones para la realización de proyectos final de carrera http://www.proyectosfindecarrera.com
- Enciclopedia libre online http://www.wikipedia.org
- Sitio del sitema operativo GNU http://www.gnu.org/
- Sitio Web de Universidad Oberta de Cataluña http://www.uoc.edu
- Proyectos anteriores de GNU/LINUX de la UOC (Repositorio Institucional).
- Sitio oficial del proyecto de alta disponibilidad libre Heartbeat http://www.linux-ha.org/wiki/Main_Page
- Proyecto HA Pacemaker http://clusterlabs.org/
- Base de datos Oracle MySQL http://dev.mysql.com/doc/refman/5.0/es/installing.html
- Pagina sobre Veritas Cluster http://www.veritas-cluster.com/
- Logival volume group información. http://es.wikipedia.org/wiki/LVM
- Archlinux https://wiki.archlinux.org/index.php/LVM_(Espa%C3%B1ol)
- Página oficial del sistema libre opensuse basado en Suse http://es.opensuse.org/
- Linux Professional Institute http://www.lpi.org
- Site de Alan Robertson fundador de Linux-HA http://techthoughts.typepad.com/
- Portal con información sobre todo tipo de clustering en GNU/Linux http://lcic.org
- Proyecto de alta disponibilidad en GNU/Linux http://www.keepalived.org/
- Web del sistema de gestión de base de datos enfocada a entornos empresariales http://www.oracle.com/
- Colección de software y paquetes linux HA https://theory.org/software/lvsm/
- http://www.ha-cc.org/high_availability/components/application_availability/cluster/high_availability_cluster/steeleye_lifekeeper/
- Proyecto Oscar HA http://xcr.cenit.latech.edu/ha-oscar/
- Cluster Resources, Torque Administrator's Manual http://www.clusterresources.com/torquedocs21/
- The Beowulf Cluster Site, http://www.beowulf.org/
- RFC 3530, Network File System (NFS) version 4 Protocol, http://tools.ietf.org/html/rfc3530
- Cluster Resources, Torque Resource Manager, http://www.clusterresources.com/products/torque/
- Cluster Resources, Maui Cluster Scheduler, http://www.clusterresources.com/products/maui/
- Cluster Resources, Torque Administrator's Manual http://www.clusterresources.com/torquedocs21/
- GNU Screen, http://www.gnu.org/software/screen/
- Rsync webpage, http://rsync.samba.org/
- Cluster SSH - Cluster Admin Via SSH, http://sourceforge.net/projects/clusterssh/
- Project C3 - Cluster command and control, http://www.csm.ornl.gov/torc/C3/
- Heartbeat http://www.linux-ha.org/wiki/Heartbeat
- Lista de correo de Linux-HA (Heartbeat) http://lists.linux-ha.org/cgi-bin/mailman/listinfo/linux-ha-announce
- Código fuente de Kernel GNU/Linux http://www.kernel.org
- Sitio Web oficial del proyecto de alto rendimiento Linux Virtual Server http://www.linuxvirtualserver.org
- Sitio oficial de sistema Unix http://www.unix.org