

**UNIVERSIDAD LUTERANA SALVADOREÑA
FACULTAD DE CIENCIAS Y EL HOMBRE Y LA NATURALEZA**

Cátedra: Sistemas Operativos de Redes.

Docente: Ingeniero. Manuel de Jesús Flores Villatoro.

Evaluación: Defensa Final.

Fecha de entrega: 14/11/2015.

Clúster de Servidor de Ficheros de Alta Disponibilidad.

Estudiantes:

| No. | Apellidos | Nombres | Carnet | Participación |
|------------|------------------|----------------|---------------|----------------------|
| 1 | Coto Zelaya | Jorge Alberto | CZ01132889 | (100%) |
| 2 | Rivas | Carlos Alberto | R01121167 | (100%) |

Índice de contenido

| | |
|---|----|
| Introducción..... | 1 |
| Objetivos..... | 2 |
| General:..... | 2 |
| Específicos:..... | 2 |
| Marco Teórico..... | 3 |
| Descripción del proyecto:..... | 3 |
| Beneficios de la tecnología clúster:..... | 5 |
| Clasificación de los clústeres:..... | 5 |
| Componentes de un clúster..... | 6 |
| Nodos:..... | 6 |
| Almacenamiento:..... | 7 |
| Sistema operativo..... | 7 |
| Conexiones de red..... | 8 |
| Middleware:..... | 10 |
| Ambientes de Programación Paralela:..... | 10 |
| Servicios Utilizados:..... | 11 |
| ¿Qué pasa cuando este servidor deja de dar servicio?..... | 12 |
| ¿Cómo lo hacemos?..... | 13 |
| Conexiones de red..... | 13 |
| Diagrama de Red..... | 14 |
| Construcción del proyecto..... | 15 |
| Configuraciones Clúster Servidores..... | 15 |
| Escenario de pruebas..... | 27 |
| Prueba de Fallo del Servidor principal..... | 28 |
| Buenas prácticas para la construcción del proyecto..... | 30 |
| Conclusión..... | 33 |
| Recomendación..... | 34 |
| Bibliografía..... | 35 |

Índice de figuras

| | |
|--|----|
| Ilustración 1: Diagrama de Red..... | 14 |
| Ilustración 2: nodo1..... | 15 |
| Ilustración 3: nodo2..... | 15 |
| Ilustración 4: interfaz_1..... | 16 |
| Ilustración 5: interfaz_2..... | 16 |
| Ilustración 6: configuracion_interfaz..... | 17 |
| Ilustración 7: configuracion_interfaz..... | 18 |
| Ilustración 8: host_1..... | 18 |
| Ilustración 9: host_2..... | 19 |
| Ilustración 10: apache2..... | 19 |

| | |
|--|----|
| Ilustración 11: instalaciones_servicios..... | 19 |
| Ilustración 12: descomprimir..... | 20 |
| Ilustración 13: authkeys..... | 20 |
| Ilustración 14: authkeys2..... | 21 |
| Ilustración 15: ha.cf nodo 2..... | 22 |
| Ilustración 16: ha.cf nodo1..... | 22 |
| Ilustración 17: haresources nodo2..... | 22 |
| Ilustración 18: haresources nodo1..... | 22 |
| Ilustración 19: reiniciar heartbeat..... | 23 |
| Ilustración 20: global_common.conf..... | 23 |
| Ilustración 21: r0.res..... | 23 |
| Ilustración 22: cfdisk..... | 25 |
| Ilustración 23: comprobacion_DRBD..... | 25 |
| Ilustración 24: Prueba_heartbeat..... | 27 |

Introducción

La necesidad de tener siempre un servicio activo, de tal manera que siempre este activo el servicios de ficheros en un servido, tratando de evitar errores tales como que si el servidor se arruine, falle por causa de software o hardware, o cualquier otra falla y no pueda satisfacer la demanda de uno o varios usuarios, nos hace implementar servidores de alta disponibilidad. Al hablar de servidor de alta disponibilidad nos estamos refiriendo a clúster, que permitan tener en funcionamiento de forma permanente el acceso a los ficheros que son demandados por los usuarios.

Un clúster es un conjunto de maquinas que trabajan como una solo, a cada máquina se le dará el nombre de **nodo**, su funcionamiento será permanente, como son varias maquinas o servidores su trabajo permitirá que siempre esté disponible, ya que si un servidor falla entrara en actividad otro de tal forma que siempre el servicio este activo.

En este proyecto se han configurado dos servidores virtuales y un cliente, para demostrar que si funcionan los cluster y que si hay alta disponibilidad.

La **alta disponibilidad** hace referencia a que un sistema debe estar funcionando las 24 horas del día los 365 días al año. Para poder alcanzar esta característica es necesario el uso de técnicas de balanceo de carga y de recuperación ante fallos (*failover*).

Para lograr el objetivo de crear un servidor de alta disponibilidad se ha tenido que configurar en el presente proyecto dos nodos(servidores) y un cliente con maquinas virtuales que constituyen el clúster que tendrá la responsabilidad de mantener el servicio activo, para ello se ha tenido que usar Virtualbox, configurándoles su IP a cada nodo, he incluyendo una IP virtual que se encargara de realizar los enlaces con los nodos para mantener el servicio activo.

La configuración de del cluster ha demandado el uso de software libre tal es el caso del paquete **Heartbeat** que esta encargado de hacer un Servicio(Demonio) que proporciona funcionalidad de infraestructura de cluster (comunicación y pertenencia) a sus clientes.

Pacemaker que es el encargado de verificar el estado de los recursos que le han sido asignados, **DRBD** (Distributed Replicated Block Device), que permite hacer la replica o espejo en tiempo real, claro no se dejara de lado el sistema operativo, en este caso se ha utilizado Debian Wheezy 8.0, Después de hacer la configuración y realizar todas la pruebas necesarias el cluster funciona de manera óptima, haciendo de este proyecto un éxito.

Objetivos

General:

- ✓ Realizar un Clúster de Servidor de Ficheros de Alta Disponibilidad que permita a los usuarios mantener y acceder a sus documentos y archivos cuando estos lo deseen, sin interrupción alguna en cuanto a disponibilidad y descarga o subida de un fichero.

Específicos:

1. Utilizar un sistema operativo basado en GNU/LINUX con su distribución Debian Wheezy en su versión mas reciente.
2. Editar y Actualizar los repositorios de Debian Wheezy en ambos servidores.
3. Configurar el Archivo que contiene las interfaces de red en ambos servidores.
4. Instalar los servicios necesarios en los dos servidores para la implementación de dicho proyecto.
5. Configurar los archivos del paquete Heartbeat en ambos servidores.
6. Configurar los archivos del paquete DRBD en ambos servidores.
7. Realizar pruebas de comprobación que dicho proyecto esta funcional.

Marco Teórico

CLÚSTER DE SERVIDOR DE FICHEROS DE ALTA DISPONIBILIDAD

En este proyecto se detalla una serie de definiciones el cual veremos a continuación:

1. **Clúster:** El término clúster (del inglés *cluster*, "grupo" o "raíz") se aplica a los conjuntos o conglomerados de computadoras unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuesen una única computadora.
2. **Alta disponibilidad:** Son clústeres cuyo objetivo de diseño es el de proveer disponibilidad y confiabilidad. Estos clústeres tratan de brindar la máxima disponibilidad de los servicios que ofrecen. La confiabilidad se provee mediante software que detecta fallos y permite recuperarse frente a los mismos, mientras que en hardware se evita tener un único punto de fallos.
3. **Nodos:** es un punto de intersección, conexión o unión de varios elementos que confluyen en el mismo lugar, En redes de computadoras cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo

Descripción del proyecto:

El proyecto consiste en implementar un Clúster de Servidor de Ficheros de Alta Disponibilidad. Se deberán configurar dos servidores de archivos que contengan la misma información y que se repliquen de manera que cuando un servidor falle, el servidor activo brinde el servicio sin interrupción del servicio.

Requisitos preliminares:

1. Tres computadoras (o máquinas virtuales), un cliente y dos servidores
2. Sistema Operativo GNU/Linux
3. Switch que interconecte los equipos.

Entonces el proyecto de servidor de archivos brindan que los usuarios pueden almacenar de forma centralizada sus documentos permite una mejor experiencia en la web y una mejor confianza de que puedes acceder a tus archivos en cualquier momento. El hecho de hacerlo de esta forma proporciona muchas ventajas en el entorno: el usuario puede cambiar de dispositivo sin perder el acceso a los ficheros, el administrador de sistemas puede controlar correctamente el acceso a los mismos estableciendo los niveles de seguridad necesarios, controlar el espacio físico que ocupan los datos, hacer copias de seguridad, etc.

El Clúster como lo especificaba anteriormente se aplica a los conjuntos o conglomerados de computadoras unidos entre sí normalmente por una red de alta velocidad y que se comportan como si fuesen una única computadora.

La tecnología de clústeres ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software para aplicaciones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

El cómputo con clústeres surge como resultado de la convergencia de varias tendencias actuales que incluyen la disponibilidad de microprocesadores económicos de alto rendimiento y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, así como la creciente necesidad de potencia computacional para aplicaciones que la requieran. Simplemente, un clúster es un grupo de múltiples ordenadores unidos mediante una red de alta velocidad, de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes de escritorio. Los clústeres son usualmente empleados para mejorar el rendimiento y/o la disponibilidad por encima de la que es provista por un solo computador típicamente siendo más económico que computadores individuales de rapidez y disponibilidad comparables.

De un clúster se espera que presente combinaciones de los siguientes servicios:

1. Alto rendimiento
2. Alta disponibilidad
3. Balanceo de carga
4. Escalabilidad

La construcción de los ordenadores del clúster es más fácil y económica debido a su flexibilidad: pueden tener todos la misma configuración de hardware y sistema operativo (clúster homogéneo), diferente rendimiento pero con arquitecturas y sistemas operativos similares (clúster semihomogéneo), o tener diferente hardware y sistema operativo (clúster heterogéneo), lo que hace más fácil y económica su construcción.

Para que un clúster funcione como tal, no basta solo con conectar entre sí los ordenadores, sino que es necesario proveer un sistema de manejo del clúster, el cual se encargue de interactuar con el usuario y los procesos que corren en él para optimizar el funcionamiento.

Beneficios de la tecnología clúster:

Las aplicaciones paralelas escalables requieren: buen rendimiento, baja latencia, comunicaciones que dispongan de gran ancho de banda, redes escalables y acceso rápido a archivos. Un clúster puede satisfacer estos requisitos usando los recursos que tiene asociados a él; La tecnología clúster permite a las organizaciones incrementar su capacidad de procesamiento usando tecnología estándar, tanto en componentes de hardware como de software que pueden adquirirse a un costo relativamente bajo.

Clasificación de los clústeres:

El término clúster tiene diferentes connotaciones para diferentes grupos de personas. Los tipos de clústeres, establecidos de acuerdo con el uso que se dé y los servicios que ofrecen, determinan el significado del término para el grupo que lo utiliza. Los clústeres pueden clasificarse según sus características:

- ➔ HPCC (*High Performance Computing Clusters*: clústeres de alto rendimiento).
- ➔ HA o HACC (*High Availability Computing Clusters*: clústeres de alta disponibilidad).
- ➔ HT o HTCC (*High Throughput Computing Clusters*: clústeres de alta eficiencia).

Alto rendimiento: Son clústeres en los cuales se ejecutan tareas que requieren de gran capacidad computacional, grandes cantidades de memoria, o ambos a la vez. El llevar a cabo estas tareas puede comprometer los recursos del clúster por largos periodos de tiempo.

Alta disponibilidad: Son clústeres cuyo objetivo de diseño es el de proveer disponibilidad y confiabilidad. Estos clústeres tratan de brindar la máxima disponibilidad de los servicios que ofrecen. La confiabilidad se provee mediante software que detecta fallos y permite recuperarse frente a los mismos, mientras que en hardware se evita tener un único punto de fallos.

Alta eficiencia: Son clústeres cuyo objetivo de diseño es el ejecutar la mayor cantidad de tareas en el menor tiempo posible. Existe independencia de datos entre las tareas individuales. El retardo entre los nodos del clúster no es considerado un gran problema.

Componentes de un clúster

En general, un clúster necesita de varios componentes de software y hardware para poder funcionar:

1. Nodos.
2. Almacenamiento.
3. Sistemas operativos.
4. Conexiones de red.
5. Middleware.
6. Protocolos de comunicación y servicios.
7. Aplicaciones.
8. Ambientes de programación paralela.

Nodos:

Pueden ser simples ordenadores, sistemas multiprocesador o estaciones de trabajo (workstations). En informática, de forma muy general, un nodo es un punto de intersección o unión de varios elementos que confluyen en el mismo lugar. Ahora bien, dentro de la informática la palabra nodo puede referirse a conceptos diferentes según el ámbito en el que nos movamos:

- En redes de computadoras cada una de las máquinas es un nodo, y si la red es Internet, cada servidor constituye también un nodo.
- En estructuras de datos dinámicas un nodo es un registro que contiene un dato de interés y al menos un puntero para referenciar (apuntar) a otro nodo. Si la estructura tiene solo un puntero, la única estructura que se puede construir con él es una lista, si el nodo tiene más de un puntero ya se pueden construir estructuras más complejas como árboles o grafos.

Almacenamiento:

El almacenamiento puede consistir en una NAS, una SAN, o almacenamiento interno en el servidor. El protocolo más comúnmente utilizado es NFS (Network File System), sistema de ficheros compartido entre servidor y los nodos. Sin embargo existen sistemas de ficheros específicos para clústeres como Lustre (CFS) y PVFS2.

Tecnologías en el soporte del almacenamiento en discos duros:

- IDE o ATA: velocidades de 33, 66, 100, 133 y 166 MB/s
- SATA: velocidades de 150, 300 y 600 MB/s
- SCSI: velocidades de 160, 320, 640 MB/s. Proporciona altos rendimientos.
- SAS: aúna SATA-II y SCSI. Velocidades de 300 y 600 MB/s
- Las unidades de cinta (DLT) son utilizadas para copias de seguridad por su bajo costo.

NAS (Network Attached Storage) es un dispositivo específico dedicado al almacenamiento a través de red (normalmente TCP/IP) que hace uso de un sistema operativo optimizado para dar acceso a través de protocolos CIFS, NFS, FTP o TFTP.

Por su parte, DAS (Direct Attached Storage) consiste en conectar unidades externas de almacenamiento SCSI o a una SAN (*storage area network*: ‘red de área de almacenamiento’) a través de un canal de fibra. Estas conexiones son dedicadas.

Mientras NAS permite compartir el almacenamiento, utilizar la red, y tiene una gestión más sencilla, DAS proporciona mayor rendimiento y mayor fiabilidad al no compartir el recurso.

Sistema operativo

Un sistema operativo debe ser multiproceso y multiusuario. Otras características deseables son la facilidad de uso y acceso. Un sistema operativo es un programa o conjunto de programas de computadora destinado a permitir una gestión eficaz y segura de sus recursos. Comienza a trabajar cuando el gestor de arranque carga en memoria su núcleo y gestiona el hardware de la máquina desde los niveles más básicos, permitiendo también la interacción con el usuario. Se puede encontrar normalmente en la mayoría de los aparatos electrónicos que utilicen microprocesadores para funcionar, ya que gracias a estos podemos entender la máquina y que ésta cumpla con sus funciones (teléfonos móviles, reproductores de DVD, radios, computadoras, etc.).

Ejemplos:

- **GNU/Linux**
 - ABC GNU/Linux1
 - OpenMosix
 - Rocks2
 - Kerrighed
 - Cóndor
 - Sun Grid Engine
- **Unix**
 - Solaris
 - HP-UX
 - AIX
- **Windows**
 - NT
 - 2000 Server
 - 2003 Server
 - 2008 Server
- **Mac OS X**
 - Xgrid
- **Solaris**
- **FreeBSD**

Conexiones de red

Los nodos de un clúster pueden conectarse mediante una simple red Ethernet con placas comunes (adaptadores de red o NICs), o utilizarse tecnologías especiales de alta velocidad como Fast Ethernet, Gigabit Ethernet, Myrinet, InfiniBand, SCI, etc.

- **Ethernet**
 - Son las redes más utilizadas en la actualidad, debido a su relativo bajo coste. No obstante, su tecnología limita el tamaño de paquete, realizan excesivas comprobaciones de error y sus protocolos no son eficientes, y sus velocidades de transmisión pueden limitar el rendimiento de los clústeres. Para aplicaciones con paralelismo de grano grueso puede suponer una solución acertada.
 - La opción más utilizada en la actualidad es Gigabit Ethernet (1 Gbit/s), siendo emergente la solución 10 Gigabit Ethernet (10 Gbit/s). La latencia de estas tecnologías está en torno a los 30 a 100 μ s, dependiendo del protocolo de comunicación empleado.

- En todo caso, es la red de administración por excelencia, así que aunque no sea la solución de red de altas prestaciones para las comunicaciones, es la red dedicada a las tareas administrativas.
- **Myrinet (Myrinet 2000 y Myri-10G).**
 - Su latencia es de 99 a 10 μ s, y su ancho de banda es de 2 a 10 Gbit/s (para Myrinet 2000 y Myri-10G, respectivamente).
 - Es la red de baja latencia más utilizada en la actualidad, tanto en clústeres como en MPP; está presente en más de la mitad de los sistemas del top500. Tiene dos bibliotecas de comunicación a bajo nivel (GM y MX). Sobre estas bibliotecas están implementadas MPICH-GM, MPICH-MX, Sockets-GM y Sockets MX, para aprovechar las excelentes características de Myrinet. Existen también emulaciones IP sobre TCP/IP, IPoGM e IPoMX.
- **InfiniBand**
 - Es una red surgida de un estándar desarrollado específicamente para realizar la comunicación en clústers. Una de sus mayores ventajas es que mediante la agregación de canales (x1, x4 y x12) permite obtener anchos de banda muy elevados. La conexión básica es de 2 Gbit/s efectivos y con 'quad connection' x12 alcanza los 96 Gbit/s. No obstante, los startups no son muy altos, se sitúan en torno a los 10 μ s.
 - Define una conexión entre un nodo de computación y un nodo de I/O. La conexión va desde un Host Channel Adapter (HCA) hasta un Target Channel Adapter (TCA). Se está usando principalmente para acceder a arrays de discos SAS.
- **SCI (*scalable coherent interface*) IEEE standard 1596-1992**
 - Su latencia teórica es de 1,43 μ s y su ancho de banda de 5333 Mbit/s bidireccional. Al poder configurarse con topologías de anillo (1D), toro (2D) e hipercubo (3D) sin necesidad de switch, se tiene una red adecuada para clústers de pequeño y mediano tamaño.
 - Al ser una red de extremadamente baja latencia, presenta ventajas frente a Myrinet en clústeres de pequeño tamaño al tener una topología punto a punto y no ser necesaria la adquisición de un conmutador. El software sobre SCI está menos desarrollado que sobre Myrinet, pero los rendimientos obtenidos son superiores, destacando SCI Sockets (que obtiene startups de 3 microsegundos) y ScaMPI, una biblioteca MPI de elevadas prestaciones.

- Además, a través del mecanismo de preloading (LD_PRELOAD) se puede conseguir que todas las comunicaciones del sistema vayan a través de SCI-SOCKETS (transparencia para el usuario).

Middleware:

El middleware es un software que generalmente actúa entre el sistema operativo y las aplicaciones con la finalidad de proveer a un clúster lo siguiente:

- Una interfaz única de acceso al sistema, denominada SSI (*Single System Image*), la cual genera la sensación al usuario de que utiliza un único ordenador muy potente;
- Herramientas para la optimización y mantenimiento del sistema: migración de procesos, *checkpoint-restart* (congelar uno o varios procesos, mudarlos de servidor y continuar su funcionamiento en el nuevo host), balanceo de carga, tolerancia a fallos, etc.;
- Escalabilidad: debe poder detectar automáticamente nuevos servidores conectados al clúster para proceder a su utilización.

Existen diversos tipos de middleware, como por ejemplo: MOSIX, OpenMOSIX, Cándor, OpenSSI, etc.

El middleware recibe los trabajos entrantes al clúster y los redistribuye de manera que el proceso se ejecute más rápido y el sistema no sufra sobrecargas en un servidor. Esto se realiza mediante políticas definidas en el sistema (automáticamente o por un administrador) que le indican dónde y cómo debe distribuir los procesos, por un sistema de monitorización, el cual controla la carga de cada CPU y la cantidad de procesos en él.

El middleware también debe poder migrar procesos entre servidores con distintas finalidades:

- balancear la carga: si un servidor está muy cargado de procesos y otro está ocioso, pueden transferirse procesos a este último para liberar de carga al primero y optimizar el funcionamiento;
- Mantenimiento de servidores: si hay procesos corriendo en un servidor que necesita mantenimiento o una actualización, es posible migrar los procesos a otro servidor y proceder a desconectar del clúster al primero;
- Priorización de trabajos: en caso de tener varios procesos corriendo en el clúster, pero uno de ellos de mayor importancia que los demás, puede migrarse este proceso a los servidores que posean más o mejores recursos para acelerar su procesamiento.

Ambientes de Programación Paralela:

Los ambientes de programación paralela permiten implementar algoritmos que hagan uso de recursos compartidos: CPU (*central processing unit*: 'unidad central de proceso'), memoria, datos y servicios.

Un **cluster de alta disponibilidad** es un conjunto de dos o más máquinas que se caracterizan por mantener una serie de servicios compartidos y por estar constantemente monitorizándose entre sí. Podemos dividirlo en dos clases:

Alta disponibilidad de infraestructura: Si se produce un fallo de hardware en alguna de las máquinas del cluster, el software de alta disponibilidad es capaz de arrancar automáticamente los servicios en cualquiera de las otras máquinas del cluster (failover). Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original (failback). Esta capacidad de recuperación automática de servicios nos garantiza la alta disponibilidad de los servicios ofrecidos por el cluster, minimizando así la percepción del fallo por parte de los usuarios.

Alta disponibilidad de aplicación: Si se produce un fallo del hardware o de las aplicaciones de alguna de las máquinas del cluster, el software de alta disponibilidad es capaz de arrancar automáticamente los servicios que han fallado en cualquiera de las otras máquinas del cluster. Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original. Esta capacidad de recuperación automática de servicios nos garantiza la integridad de la información, ya que no hay pérdida de datos, y además evita molestias a los usuarios, que no tienen por qué notar que se ha producido un problema.

No hay que confundir un cluster de alta disponibilidad con un cluster de alto rendimiento. El segundo es una configuración de equipos diseñado para proporcionar capacidades de cálculo mucho mayores que la que proporcionan los equipos individuales (véanse por ejemplo los sistemas de tipo Cluster Beowulf), mientras que el primer tipo de cluster está diseñado para garantizar el funcionamiento ininterrumpido de ciertas aplicaciones.

Servicios Utilizados:

1- **Heartbeat** es un Servicio(Demonio) que proporciona funcionalidad de infraestructura de cluster (comunicación y pertenencia) a sus clientes.

Esto permite a los clientes tener conocimiento de la presencia (o desaparición) de los procesos en otras maquinas e intercambiar fácilmente mensajes entre ellos.

Para resultar útil a los usuarios el dominio HEARTBEAT necesita emplearse en combinación con un gestor de recursos del cluster (cluster resource manager (CRM)) el cual posee la tarea de iniciar y parar los servicios (Direcciones IP, servidores web...) a los cuales el cluster aportará alta disponibilidad. *Pacemaker* es el gestor de recursos de cluster preferido para los clusters

basados en Heartbeat.

Para que sirve: Heartbeat es el Encargado de revisar que cada nodo se halle funcionando. En caso un nodo falle migrará los recursos a otro nodo que también se halle ejecutando el servicio heartbeat.

2- Pacemaker: Verifica el estado de los recursos (o servicios) que le han sido asignados. En caso algún servicio falle, en caso se halla configurado, Pacemaker puede reiniciarlo.

Mientras heartbeat se encarga que revisar el estado de cada nodo; Pacemaker es el responsable de verificar el estado de los servicios que deseemos sean HA dentro de los nodos.

3- DRBD (*Distributed Replicated Block Device*):

Permite mirror remoto en tiempo real (equivalente a RAID-1 en red), algo muy difícil de conseguir con otros sistemas como rsync ya que éste no puede trabajar en tiempo real por su consumo de memoria y CPU.

DRDB crea un dispositivo de bloques *drbd0* accesible desde ambos servidores. El servidor primario es el que tiene acceso RW en el dispositivo *drbd0*: cada vez que escribe algo en *drbd0* lo escribe en la partición física y esos mismos datos se envían por TCP/IP al servidor secundario (que sólo tiene acceso RO) consiguiendo que ambas particiones físicas estén sincronizadas, exactamente igual que un RAID-1.

¿Qué pasa cuando este servidor deja de dar servicio?

Ya sea porque hay que hacer las actualizaciones correspondientes, mantenimiento del hardware que lo soporta, ampliaciones, un problema cualquiera que deja el servidor frito, etc. Pues, que **los usuarios dejan de tener acceso** a sus documentos y ya la hemos armado.

Cada vez más muchos servicios de TI se convierten en vitales para el funcionamiento de la empresa. Las empresas también han cambiado, han pasado de un modelo donde encontrábamos ventanas de tiempo para poder hacer las tareas de mantenimiento, a un modelo donde estas ventanas cada vez son menos. Usuarios que se conectan desde casa al atardecer, noche o fin de semana; delegaciones en la otra punta del mundo con husos horarios diferentes, entorno de producción 24×7 que se ha integrado en la red informática, etc... Y no hablo de las grandes empresas (que ya lo tienen), sino también de las PYMES, que cada vez más adoptan (o tienen que adoptar) este modelo.

Frente a este reto empresarial, nuestros departamentos de TI tienen que dar una respuesta firme, que permita dormir tranquilos, sin trabajar a deshoras y corriendo, pudiendo hacer las tareas de almacenamiento tranquilos y sin que aparezca la tan temible gota de sudor frío.

¿Cómo lo hacemos?

Buscando la **alta disponibilidad** del servicio crítico. Es decir, un sistema que me permita apagar uno de los servidores que ofrece el servicio sin que eso suponga la pérdida del mismo.

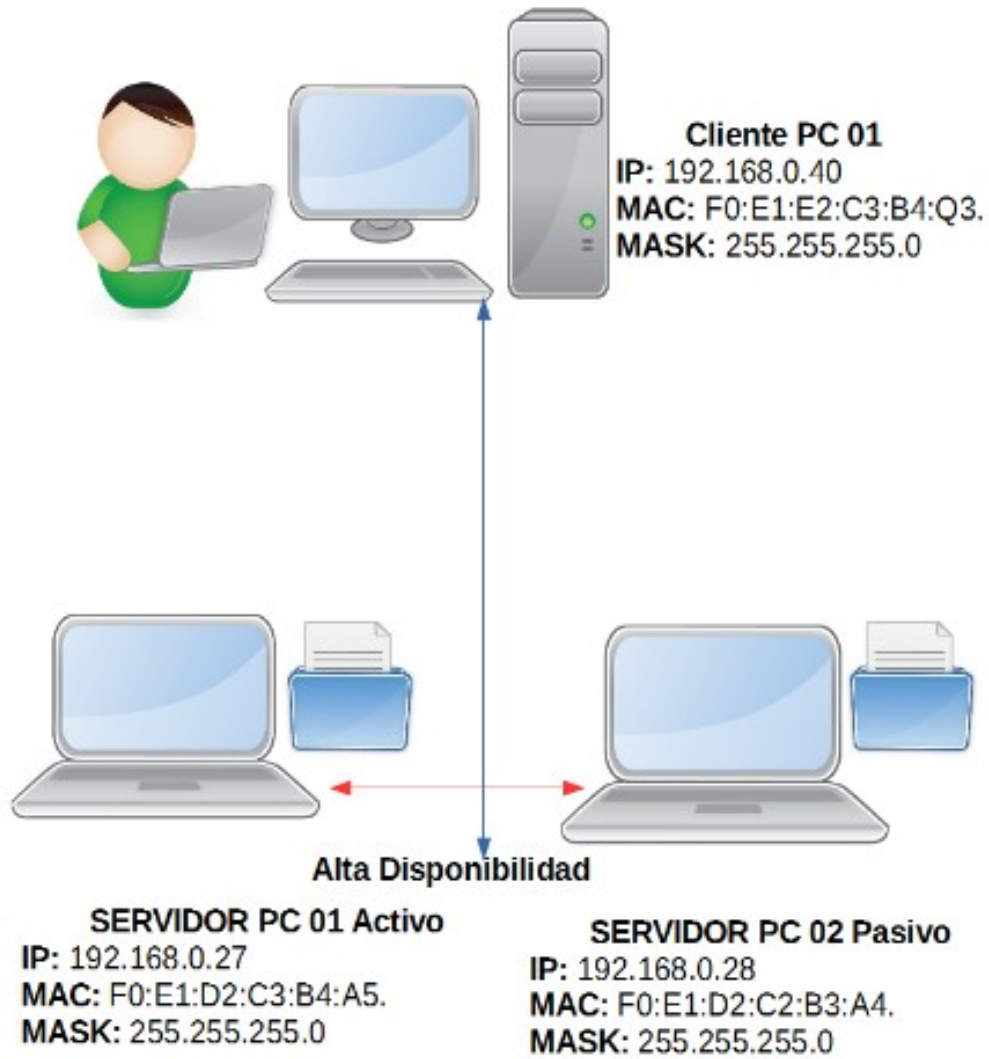
Anteriormente se conseguía mediante dos servidores y un almacenamiento compartido entre ellos, muchas veces **una solución cara al alcance de pocos**. Uno de los servidores es el que ofrece el servicio, el otro está a la expectativa de si cae el primero. Eran montajes muy habituales en bases de datos, archivos o servidores de correo electrónico. En caso de caída del primero (ya sea por mantenimiento o por un error del mismo), el segundo coge la batuta del servicio. Debido a la evolución tecnológica, actualmente, algunos servicios han mejorado esta disponibilidad incorporando métodos propios. Por ejemplo, Microsoft Exchange y Microsoft SQL Server utilizan una **tecnología de bases de datos en alta disponibilidad**, donde aparte del servidor, el servicio también se encarga de copiar los datos en ficheros diferentes asegurando la consistencia de los mismos (ya no se dispone de un servicio de almacenamiento compartido). Pero continúan existiendo una serie de servicios (servidor de archivos, DHCP, Hyper-V) en que hay que asegurar su disponibilidad que por la tipología del servicio no requieren de las otras tecnologías que he comentado. Para ellos, continúa existiendo el clúster de conmutación por error.

Los costes en hardware, licencias y la virtualización también han ayudado a **hacer más accesibles estas tecnologías para la gran mayoría**, pudiéndose aplicar de forma habitual.

Conexiones de red

Los servidores disponen de **dos tarjetas de red** (Ethernet0 y iSCSI), por lo tanto, hay que asegurarse que la tarjeta que aparece en el panel de control se corresponde a la tarjeta física (uno de los trucos que siempre funciona es desconectar el cable para ver quien pierde el enlace). Es decir, hay que **comprobar que la conexión física se corresponde con la lógica** (tarjeta de red del panel de control), sino cuando se asignen las direcciones IP podemos sufrir. También hay que asegurar que los servicios del servidor se inicien con la tarjeta que corresponde a la red (Ethernet0) donde habrá los usuarios y no a las secundarias (iSCSI).

Diagrama de Red



Construcción del proyecto

Configuraciones Clúster Servidores

La configuración básica de un cluster Heartbeat en GNU/Linux suele ser muy utilizado en producción a nivel profesional. Se llama así porque el servidor que tiene el servicio manda “latidos” unos ping con los que verifican si el servidor principal está disponible o no. El Cliente se conecta a una IP virtual que comparten ambos nodos, y dependiendo como lo tengamos configurado, uno recibiría la petición, en caso de que se detecte que no funciona correctamente el otro nodo esclavo obtiene el control del servicio en segundos. Esta es la forma de configurarlo:

Nodo 1:

Hostname nodo1

IP address: 192.168.0.27

Nodo 2:

Hostname nodo2

IP address: 192.168.0.28

IP Virtual: 192.168.0.29

1- Para empezar pondremos el nombre el equipo en ambas maquinas el cual lo identificamos con el archivo que contiene el **Hostname**, para eso ejecutaremos el comando:

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo1:/home/jorge# nano /etc/hostname
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/hostname
nodo1
```

2- Nodo 2.

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/home/jorge# nano /etc/hostname
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/hostname
nodo2
```

3- Ahora configuramos las interfaces de redes en las dos maquinas para eso, primeramente ejecutamos este comando en las dos maquinas: **ifconfig -a** que es el que muestra el estado de las interfaces de red activas.

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo1:/home/jorge# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:31:4b:b2
          inet addr:192.168.0.27  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe31:4bb2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:434 errors:0 dropped:0 overruns:0 frame:0
          TX packets:614 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:121237 (118.3 KiB)  TX bytes:169333 (165.3 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:566 errors:0 dropped:0 overruns:0 frame:0
          TX packets:566 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:44517 (43.4 KiB)  TX bytes:44517 (43.4 KiB)
```

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/home/jorge# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 08:00:27:31:4b:b2
          inet addr:192.168.0.28  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe31:4bb2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:930 errors:0 dropped:0 overruns:0 frame:0
          TX packets:688 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:228386 (223.0 KiB)  TX bytes:173104 (169.0 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:270 errors:0 dropped:0 overruns:0 frame:0
          TX packets:270 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:27985 (27.3 KiB)  TX bytes:27985 (27.3 KiB)
```

4-

Seguidamente configuramos el fichero **/etc/network/interfaces** en ambas maquinas; le daremos las IPs que detallamos anteriormente; para esto detenemos la interfaz de red eth0 con el comando **ifdown eth0** en ambas maquinas, y finalmente iniciamos la interfaces de red eth0 con el comando **ifup eth0**, posteriormente reiniciamos los servicios de las interfaces de redes con el comando **/etc/init.d/networking restart**

Nodo 1:

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nod01:/home/jorge# ifdown eth0

root@nod01:/home/jorge# nano /etc/network/interfaces |
```

Como observamos configuramos la interfaz eth0 al nodo1

```
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/network/interfaces

## This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

#ASIGNAMOS UNA IP AL NOD01
auto eth0
iface eth0 inet static
    address 192.168.0.27
    netmask 255.255.255.0
```

Ilustración 6: configuracion_interfaz

Iniciar la interfaz de red eth0:

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nod01:/home/jorge# ifup eth0
```

Reiniciamos las interfaces de red y reiniciamos el equipo para obtener una mejor configuración.

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nod01:/home/jorge# /etc/init.d/networking restart
[warn] Running /etc/init.d/networking restart is deprecated because it may not re-enable some interfaces ... (warning)
[...] Reconfiguring network interfaces...mount: block device /dev/drbd0 is write-protected, mounting read-only
mount: Wrong medium type
done.
```

Nodo 2:

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/home/jorge# ifdown eth0

root@nodo2:/home/jorge# nano /etc/network/interfaces
```

Como observamos configuramos la interfaz eth0 al nodo2

```
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

#ASIGNAMOS UNA IP AL NODO2
auto eth0
iface eth0 inet static
    address 192.168.0.28
    netmask 255.255.255.0
```

Iniciar la interfaz de red eth0:

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/home/jorge# ifup eth0
```

Reiniciamos las interfaces de red y reiniciamos el equipo para obtener una mejor configuración.

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/home/jorge# /etc/init.d/networking restart
[warn] Running /etc/init.d/networking restart is deprecated because it may not re-enable some interfaces ... (warning)
[....] Reconfiguring network interfaces...mount: block device /dev/drbd0 is write-protected, mounting read-only
mount: Wrong medium type
done.
```

5- Si no tenemos un servidor DNS corriendo en nuestra red tendremos de usar nuestro archivo; en donde colocamos las Ips que hemos asignado y su hostname para esto nos vamos a **/etc/hosts** quedando de esta manera en las dos maquinas:

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo1:/home/jorge# nano /etc/hosts
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/hosts

127.0.0.1 localhost
127.0.1.1 debian.debian.com debian
#127.0.0.1 hr.dev

192.168.0.27 nodo1
192.168.0.28 nodo2
```

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/home/jorge# nano /etc/hosts

Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/hosts

127.0.0.1 localhost
127.0.1.1 debian.debian.com debian
#127.0.0.1 hr.dev

192.168.0.27 nodol
192.168.0.28 nodo2
```

6- Ahora procedemos a instalar los servicios que utilizaremos **en ambas maquinas**, los cuales son: **Apache2, Heartbeat, y DRBD**.

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo1:/home/jorge# apt-get install apache2

Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/home/jorge# apt-get install heartbeat
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
heartbeat va está en su versión más reciente.

Archivo Editar Ver Buscar Terminal Ayuda
root@nodo1:/home/jorge# apt-get install drbd8-utils
```

7- Los pasos que veremos a continuación se configuraran en las dos maquinas, para iniciar configuraremos el paquete heartbeat, el cual tiene 3 ficheros de configuración:

- 1. ha.cf:** Configura información básica del cluster
- 2. haresources.cf:** Configura los grupos de recursos tipo init
- 3. authkeys:** Configura la Autenticación de red.

Heartbeat dispone de plantillas para configurarlos de forma manualmente para esto nos iremos a **/usr/share/doc/heartbeat/**.Entonces Podemos tomarlo pero lo copiaremos a a este directorio **/etc/heartbeat;** para que nosotros lo empezemos a hacer nuestras configuraciones.

Como observamos copiamos los archivos al directorio que especificamos y también comprobamos que en el directorio `cd /etc/ha.d/` están también. Entonces como vemos en la imagen los descomprimimos. Con el comando `gzip -d nombredearchivo`; Seguidamente procedemos a hacer nuestra configuración.

```

Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/# cd /usr/share/doc/heartbeat/
root@nodo2:/usr/share/doc/heartbeat# ls -l
total 80
-rw-r--r-- 1 root root 1873 oct 20 2011 apphbd.cf
-rw-r--r-- 1 root root 645 oct 20 2011 authkeys
-rw-r--r-- 1 root root 3701 oct 20 2011 AUTHORS
-rw-r--r-- 1 root root 18385 oct 20 2011 changelog.Debian.gz
-rw-r--r-- 1 root root 19884 oct 20 2011 changelog.gz
-rw-r--r-- 1 root root 3507 dic 16 2010 copyright
-rw-r--r-- 1 root root 4438 oct 20 2011 ha.cf.gz
-rw-r--r-- 1 root root 2197 oct 20 2011 haresources.gz
-rw-r--r-- 1 root root 2935 oct 20 2011 README
-rw-r--r-- 1 root root 7701 oct 20 2011 README.cts.gz
root@nodo2:/usr/share/doc/heartbeat# cp ha.cf.gz haresources.gz authkeys /etc/heartbeat/
root@nodo2:/usr/share/doc/heartbeat# cd /etc/ha.d/
root@nodo2:/etc/ha.d# ls -l
total 36
-rw-r--r-- 1 root root 645 oct 20 13:05 authkeys
-rw-r--r-- 1 root root 4438 oct 20 13:05 ha.cf.gz
-rwxr-xr-x 1 root root 745 oct 20 2011 harc
-rw-r--r-- 1 root root 2197 oct 20 13:05 haresources.gz
drwxr-xr-x 2 root root 4096 oct 20 12:57 rc.d
-rw-r--r-- 1 root root 692 oct 20 2011 README.config
drwxr-xr-x 2 root root 4096 oct 20 12:57 resource.d
-rw-r--r-- 1 root root 2082 mar 10 2014 shellfuncs
root@nodo2:/etc/ha.d# gzip -d ha.cf.gz
root@nodo2:/etc/ha.d# gzip -d haresources.gz

```

8-Iniciamos la configuración en ambas maquinas con el archivo **authkeys** que esta ubicado en `/etc/ha.d/authkeys`. Pero primero le damos permiso para que solo root tenga acceso `chmod 600 /etc/ha.d/authkeys`

authkeys es el fichero de configuración mas sencillo de todos. Contiene el método de autenticación basado en (sha1) con la clave que se usara para firmar los paquetes . este fichero tiene que ser idéntico en todos los servidores y no debe tener ningún usuario acceso de lectura a excepción de root; este fichero nos servirá para la comunicación entre los nodos del clúster.

```

Archivo Editar Ver Buscar Terminal Ayuda
root@nodol:/home/jorge# nano /etc/ha.d/authkeys
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/ha.d/authkeys
# Use only on physically secure networks.
#
auth 1
1 crc
#2 sha1 HI!

```

```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/home/jorge# nano /etc/ha.d/authkeys
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/ha.d/authkeys
#           Use only on physically secure networks.
#
auth 1
1 crc
#2 sha1 HI!
```

9- Ahora configuramos **ha.cf** le aporta a Heartbeat la información de la configuración básica. Configura los nodos, pulsaciones serial, la manera de registrar los logs intervalo de tiempo muerto y pulsaciones.

Ejemplo nodo 1:

```
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility local0
keepalive 2
deadtime 30
warntime 10
initdead 120
udpport 694
bcast eth0
ucast eth0 192.168.0.28
auto_failback on
node nodo1
node nodo2
```

Ejemplo nodo 2:

```
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility local0
keepalive 2
deadtime 30
warntime 10
initdead 120
udpport 694
bcast eth0
ucast eth0 192.168.0.27
auto_failback on
node nodo1
node nodo2
```



```

Archivo Editar Ver Buscar Terminal Ayuda
root@nodo1:/home/jorge# nano /etc/ha.d/ha.cf

Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/ha.d/ha.cf

debugfile /var/log/ha-debug
#
#
# File to write other messages to
#
logfile /var/log/ha-log

```

```

Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/home/jorge# nano /etc/ha.d/ha.cf

Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/ha.d/ha.cf

# logfacility must be "none".
#
# File to write debug messages to
debugfile /var/log/ha-debug
#
# File to write other messages to
#
logfile /var/log/ha-log

```

La explicación

logfile y logfacility: Fichero de log y nivel.

keepalive: Especifica cada cuanto tiempo Heartbeat enviará paquetes para comprobar la disponibilidad de los nodos, 2 segundos.

deadtime: Heartbeat confirmará que un nodo ha caído, 30 segundos.

wartime: Heartbeat avisará cuando un nodo falle tras 10 segundos.

initdead: Tiempo máximo que Heartbeat esperará a que un nodo arranque, 60 segundos.

udpport 694: Puerto que sera usado.

bcast eth0 o eth1: Broadcast Interfaces Heartbeat.

ucast eth0 192.168.0.28: es la ip del nodo 2 que se pone en el cluster 1 según el ejemplo.

ucast eth0 192.168.0.27: es la ip del nodo 1 que se pone en el cluster 2 según el ejemplo.

auto_failback on: Determinando el servicio que debe responder el principal o el master.

node: Las dos últimas líneas son las listas de nodos que conforman el cluster.

10- Por ultimo configuramos en ambas maquinas el fichero **/etc/ha.d/haresources;** Que es el que contiene información de los recursos que queremos osea la alta disponibilidad. Como por ejemplo:

```

Archivo Editar Ver Buscar Terminal Ayuda
root@nodo1:/home/jorge# nano /etc/ha.d/haresources

Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/ha.d/haresour

nodo1 IPaddr2::192.168.0.29/24/eth0 apache2

```

```

Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/home/jorge# nano /etc/ha.d/haresources

Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/ha.d/haresour

nodo1 IPaddr2::192.168.0.29/24/eth0 apache2

```

11- Una vez se instala heartbeat se inicia ahora como ya isismos las configuraciones reiniciamos el servicio heartbeat con el comando **service heartbeat restart**. En ambas maquinas. EJEMPLO:

```
root@nodol:/home/jorge# service heartbeat restart
Stopping High-Availability services: Done.

Waiting to allow resource takeover to complete:Done.

Starting High-Availability services: IPAddr2[5092]: INFO: Resource is stopped
Done.

root@nodol:/home/jorge#
```

OJO: AHORA IR AL ESCENARIO DE PRUEBA PARA VER COMO FUNCIONA HASTA EL MOMENTO CREANDO UN HTML PARA PODER MOSTRAR LOS RESULTADOS.

12- Una vez finalizada la configuración del paquete heartbeat a continuación Los pasos que veremos serán de mucha concentración, para continuar configuraremos el paquete **DRBD**, el cual tiene 2 ficheros de configuración:

- **global_common.conf**
- **r0.res**

Primero cargamos los módulos en ambas maquinas y editamos el primer archivo ejemplo

```
root@nodol:/home/jorge# modprobe cn
root@nodol:/home/jorge# modprobe drbd
root@nodol:/home/jorge# nano /etc/drbd.d/global_common.conf

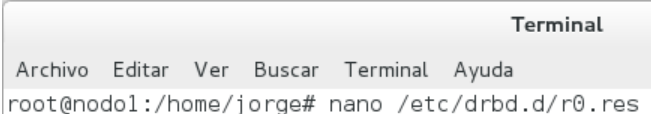
root@nodo2:/home/jorge# modprobe cn
root@nodo2:/home/jorge# modprobe drbd
root@nodo2:/home/jorge# nano /etc/drbd.
drbd.conf drbd.d/
root@nodo2:/home/jorge# nano /etc/drbd.d/global_common.conf
```

Una vez abierto el archivo solo hacemos esto en ambas maquinas:

```
global {
    usage-count no;
}
```

13- Seguidamente hacemos un archivo de texto en el directorio **/etc/drbd.d/** y una vez hecho lo abrimos para hacer nuestra configuración. Este paso es en ambas maquinas.

```
root@nodol:/home/jorge# touch /etc/drbd.d/r0.res
```



The screenshot shows a terminal window titled "Terminal" with a menu bar containing "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The command prompt shows the user is in the directory /etc/drbd.d/ and has just executed the command 'nano /etc/drbd.d/r0.res'.

14- Esto mismo debe de estar en ambas maquinas. Ejemplo:

```

#!/bin/bash
resource r0 {
protocol C;
handlers {
    pri-on-incon-degr "echo o > /proc/sysrq-trigger ; halt -f";
    pri-lost-after-sb "echo o > /proc/sysrq-trigger ; halt -f";
    local-io-error "echo o > /proc/sysrq-trigger ; halt -f";
    pri-lost "echo primary DRBD lost | mail -s "DRBD Alert" adm@gmail.com
adm@gmail.com";
}
startup {
    degr-wfc-timeout 60;
}
disk {
    on-io-error detach;
}
net {
    sndbuf-size 512k;
    timeout 60;
    connect-int 12;
    ping-int 12;
    ping-timeout 9;
    max-buffers 20480;
    cram-hmac-alg "sha1";
    shared-secret "123";
    after-sb-0pri discard-older-primary;
    after-sb-1pri violently-as0p;
    after-sb-2pri disconnect;
    rr-conflict disconnect;
}
syncer {
    rate 100M;
    al-extents 257;
}
    on nodo1 {
        device /dev/drbd0;
        disk /dev/sda6;
        address 192.168.0.27:7793;
        meta-disk internal;
    }
    on nodo2 {
        device /dev/drbd0;
        disk /dev/sda6;
        address 192.168.0.28:7793;
        meta-disk internal;
    }
}

```

15- Una vez configurado r0.res, procedemos a crear nuestra partición en ambas maquinas para esto ejecutamos el comando **fdisk** que nos mostrara esto; aquí hemos creado esa partición sda6 que vamos a usar como unidad logica de tipo ext4 de 1.7GB; después de haber hecho esto reiniciamos las dos maquinas. Para que ya nos guarden los cambios.

```

cfdisk (util-linux 2.20.1)

Unidad de disco: /dev/sda
Tamaño: 68719476736 bytes, 68.7 GB
Cabezas: 255   Sectores por pista: 63   Cilindros: 8354

```

| Nombre | Indicadores | Tipo | Tipo de S.F. | [Etiqueta] | Tamaño(MB) |
|--------|-------------|----------|----------------------|------------|------------|
| sda1 | Inicio | Primaria | Espacio libre | | 1,05 |
| sda5 | | Primaria | ext4 | | 66036,18 |
| sda6 | | Lógica | Linux swap / Solaris | | 1019,94 |
| sda6 | | Lógica | ext4 | | 1662,33 |

```

root@nod01:/home/jorge# ls -a /dev/sda6
/dev/sda6
-

```

16- Seguidamente ejecutamos estos comandos en el **nod01(Maquina 1)** primeramente y después en el **nod02(Maquina 2)**:

```

drbdadm create-md r0
modprobe drbd
drbdadm up r0

```

Después Sincronizamos por primera vez (Esto solo lo ejecutamos en el nodo1 “NO” en el nodo2):

```

drbdadm -- --overwrite-data-of-peer primary r0

```

Después ejecutamos para comprobar esto: **cat /proc/drbd**

```

root@nod01:/home/jorge# cat /proc/drbd
version: 8.3.11 (api:88/proto:86-96)
srcversion: F937DCB2E5D83C6CCE4A6C9
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
   ns:24 nr:4 dw:28 dr:697 al:2 bm:1 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0

```

17- Seguidamente iniciamos los servicios de DRBD en las dos maquinas:

/etc/init.d/drbd start : iniciamos el servicio.

Ahora la maquina 1 la asignamos como principal con el comando:

drbdadm primary all : ponemos como primaria el nodo 1.

mkfs.ext4 /dev/drbd0: formateamos la partición creada solo en el nodo1.

mkdir /home/jorge/datos: Creamos la carpeta datos en /home/jorge/: para sincronizar nuestros ficheros.

mount -t ext4 /dev/drbd0 /home/jorge/datos/ :Finalmente montamos nuestro dispositivo;

18- Una de las cosas para no estar montando este dispositivo cada vez que encienda nuestro ordenador con este comando **mount -t ext4 /dev/drbd0 /home/jorge/datos/ , nos vamos y modificamos esto:**

```
root@nod01:/home/jorge# nano /etc/ha.d/haresources
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/ha.d/haresources
nod01 IPAddr2::192.168.0.29/24/eth0 drbddisk::r0 Filesystem::/dev/drbd0::/home/jorge/datos::ext4 apache2
```

Escenario de pruebas

Para el escenario de prueba primeramente se mostrara el funcionamiento del paquete Heartbeat ya configurado anteriormente, Sucesivamente la demostración de DRBD.

Para eso ya instalado apache2, es ahí donde nosotros vamos a hacer nuestra primera prueba.

1- Nos dirigimos a la terminal del nodo1(Maquina 1) y asignamos permisos al directorio **www** y como observamos editamos el archivo **index.html** donde ponemos en una etiqueta **CLUSTER 01 FUNCIONA**. Y después lo guardamos.



```
Archivo Editar Ver Buscar Terminal Ayuda
root@nod01:/home/jorge# chmod 777 /var/www/
root@nod01:/home/jorge# nano /var/www/index.html

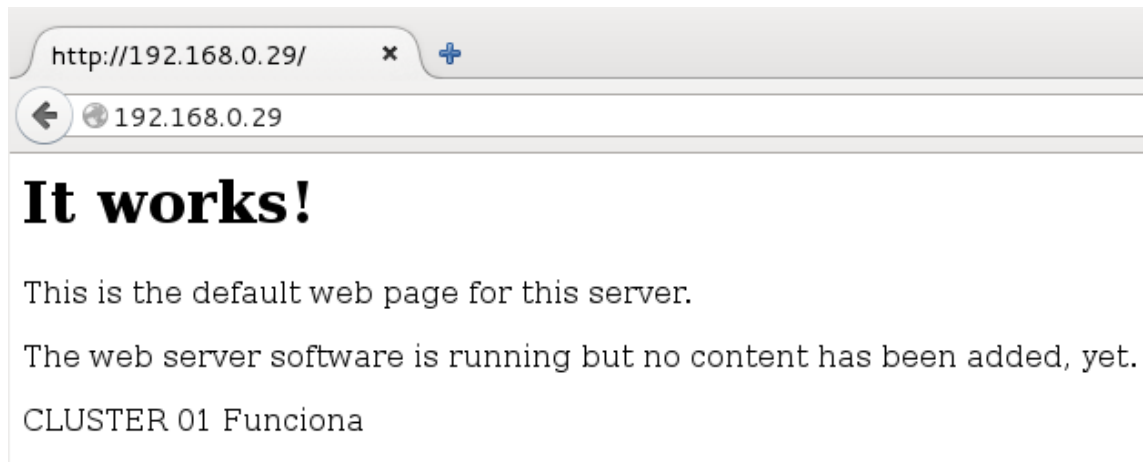
Terminal (como superusuario)
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /var/www/index.html

<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>


<p>CLUSTER 01 Funciona </p>

</body></html>
```

2- Ahora nos dirigimos al navegador; y ingresamos en la URL la dirección IP Virtual; como observamos el clúster 01 esta funcionando correctamente.



3- Ahora Nos dirigimos a la terminal del nodo2(Maquina 2) y asignamos permisos al directorio **www** y como observamos editamos el archivo **index.html** donde ponemos en una etiqueta **CLUSTER 02 FUNCIONA**. Y después lo guardamos.



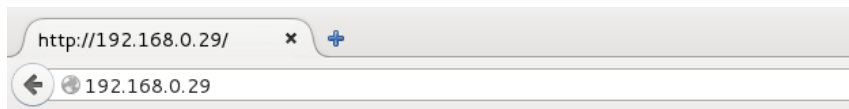
```
Archivo Editar Ver Buscar Terminal Ayuda
root@nodo2:/home/jorge# chmod 777 /var/www/
root@nodo2:/home/jorge# nano /var/www/index.html

Terminal

Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /var/www/index.html

<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
CLUSTER 02 Funciona
</body></html>
```

4- Ahora nos dirigimos al navegador; y ingresamos en la URL la dirección IP Virtual; como observamos el clúster 01 esta funcionando correctamente.



It works!

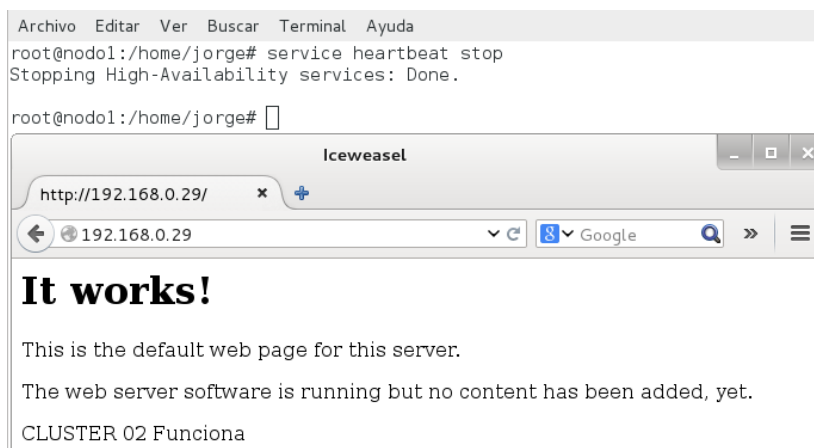
This is the default web page for this server.

The web server software is running but no content has been added, yet.

CLUSTER 01 Funciona

Prueba de Fallo del Servidor principal

Una vez detenemos el servicio **heartbeat**, automáticamente el nodo 2 brinda el servicio. Para esto ponemos siempre la IP Virtual en el navegador y como observamos nos muestra que el clúster 02 esta funcionando. Después solamente iniciamos el servicio **service heartbeat start**



Para la prueba de DRBD nos vamos a la terminal y nos vamos al directorio www y ahí creamos un archivo de texto el cual lo vamos a ver en el navegador; poniendo la IP Virtual seguidamente /nombre_del_archivo.txt; y como observamos en la imagen nos muestra correctamente.

The screenshot shows a web browser window displaying the contents of a directory named 'www'. The directory contains four files: 'proyectos' (a folder), 'index.html', 'prueba1.txt', and 'prueba2.txt'. Below the browser window is a terminal window titled 'Terminal (como superusuario)'. The terminal shows the following commands and output:

```
root@nod01:/home/jorge# cd datos/  
root@nod01:/home/jorge/datos# cd www  
root@nod01:/home/jorge/datos/www# touch prueba1.txt
```

The screenshot shows a web browser window displaying the 'Index of /proyectos' directory. The directory contains four files: 'Otros' (a folder), 'archivo1.txt', 'archivo2.txt', and 'archivo3.txt'. Below the browser window is a terminal window titled 'Terminal (como superusuario)'. The terminal shows the following commands and output:

```
root@nod01:/home/jorge# cd datos/  
root@nod01:/home/jorge/datos# cd www  
root@nod01:/home/jorge/datos/www# cd proyectos/  
root@nod01:/home/jorge/datos/www/proyectos#
```

The browser window shows the following table of contents for the directory:

| Name | Last modified | Size | Description |
|----------------------------------|-------------------|------|-------------|
| Parent Directory | - | - | |
| Otros/ | 28-Oct-2015 20:58 | - | |
| archivo1.txt | 28-Oct-2015 20:59 | 10 | |
| archivo2.txt | 28-Oct-2015 20:59 | 10 | |
| archivo3.txt | 28-Oct-2015 20:59 | 10 | |

Apache/2.2.22 (Debian) Server at 192.168.0.29 Port 80

Buenas prácticas para la construcción del proyecto

Cuando hablamos de buenas prácticas nos referimos a aquellas prácticas profesionales que resultan ser las mejores de entre todas las que los profesionales realizan para lograr los resultados esperados por sus clientes. Una buena práctica puede serlo tanto una acción muy sencilla como también un conjunto de acciones de mayor complejidad y magnitud siempre que sean prácticas y buenas. Para la construcción de nuestro proyecto se han tomado en cuenta las siguientes.

1. Defina el Alcance y los Objetivos del Proyecto

La creación de un servidor de ficheros de alta disponibilidad es el objetivo que se desea alcanzar con este proyecto, tomando en cuenta que será a través de un cluster y utilizando máquinas virtuales las cuales serán creadas y configuradas con Oracle VM Virtualbox, utilizando todos los recursos de software libre y hardware necesarios para ello.

El alcance o área de competencia define los límites del proyecto.

Los alumnos de la materia de SOR (Sistema Operativos de Redes) realizarán el proyecto utilizando para ellos todos los recursos que estén a su alcance los cuales se describen en el marco teórico del proyecto, los límites del proyecto serán la red LAN a la que estará supeditada la experiencia de la práctica que se realizarán en la configuración del proyecto, teniendo como limitantes la falta de equipo apropiado para realizarla, pero no dejando de lado los esfuerzos realizados por los estudiantes, el éxito dependerá del buen funcionamiento del proyecto (servidor de fichero de alta disponibilidad)

2. Defina las Tareas

Debe definir que tareas se esperan del proyecto. Si por ejemplo su proyecto es una campaña publicitaria para una nueva barra de chocolate, entonces una tarea sería producir el trabajo de arte para la publicidad. Por eso defina que cosas tangibles deben ser producidas y documéntelas con suficiente detalle para que cualquiera de los involucrados pueda llevarla a cabo correcta y eficientemente.

Los accionistas claves deben revisar la definición de las tareas y estar de acuerdo que las mismas reflejan adecuadamente lo que se espera.

Tareas:

1. Instalar en una maquina fisica Oracle VM Virtualbox.
2. Crear un cluster con maquinas virtuales, este estara constituido por dos nodos que seran los servidores mas un equipo cliente que puede ser fisico o virtual.
3. Realizar la investigacion de los paquetes de software requeridos para la realizacion del proyecto.
4. Instalacion de los paquetes descargados
5. configuracion de maquinas y software necesario para el funcionamiento del cluster
6. configuracion de las IP en los nodos y cliente, asi como la IP virtual que servira de enlace entre los servidores para darles alternacia.
7. Realizar pruebas de funcionamiento del cluster
8. puesta en marcha y produccion del cluster.

3. Planifique el Proyecto

Equipo que trabajo en la realizacion de el servidor de ficheros de alta dipsonibilidad (cluster de alta disponibilidad) fue constituido por tres alumnos de la carrera de licenciatura en Ciencias de la computacion inicialmente, debido a circunstancias ajenas a nuestra voluntad el equipo se redujo a dos miembros que han llevado hasta su fin el proyecto, para ellos se ha contado con los recursos de varias computadoras del aula donde se imparte la materia de SOR, pero tambien con una computadora portatil propiedad de una de los miembros del equipo, se realizo un presupuesto el cual se encuentra en la primera etapa del proyecto asi como el cronograma de actividades, las cuales fueron terminadas a su debido tiempo.

4. Comunicación

La planificación del proyecto resulta inútil si no es comunicada efectivamente al equipo de proyecto.

La comunicación entre los miembros del equipo desarrollador del proyecto fue un tanto escasa, quizas solo lo justo y necesario, lo bueno es que se trabajo en armonia, y se llevaron a cabo todas las tareas para la realizacion del mismo. Se explico como se realizaria y configuraria el servidor de alta disponibilidad, los pasos a seguir, el cumplimineto de lcada una de las etapas, hasta llegar a la finalizacion del mismo con éxito.

5. Seguimiento y Reporte de Avance del Proyecto

Se ajusto el plan de muchas maneras para volver a poner la planificación en el camino trazado para que siempre terminará equilibrando costos, cronograma de tareas y alcances. En el caso de que el equipo desarrollador del proyecto cambiara una de estas, entonces uno o los dos elementos restantes deberán inevitablemente ajustarse de forma acorde. Es justamente el balance estos tres elementos –conocidos como el triángulo del proyecto- lo que típicamente causa los mayores dolores de cabeza al equipo de proyecto.

6. Gestión del Cambio

En el presente proyecto no se dieron cambios en lo ya programado para la realización del proyecto, por lo que siguió su proceso de antemano establecido, si ninguna variación cronográfica, ni económica.

7. Gestión del Riesgo

No controlamos todos los riesgos porque estos pueden ser muchos y no todos tienen el mismo impacto.

Los riesgos más posibles son fallas en los equipos a utilizar (hardware), desconfiguración de algún componente del servidor de ficheros de alta disponibilidad (software), riesgos de infraestructura mínimos, impacto mínimo de funcionamiento.

Conclusión

La configuración de un cluster que funcione como un servidor de alta disponibilidad, nos ha permitido aprender sobre que software utilizar para lograrlo, y vencer los obtaculos que en el campo de las pruebas fueron apareciendo, así como solventar los errores que cometimos al crear los nodos virtuales y las configuraciones necesarias para lograr el objetivo.

Crear un cluster con dos nodos(servidores) y una maquina virtual como cliente, para ejecutar las pruebas así como realizar la investigación de que software eran necesarios para la realización del mismo, teniendo en cuenta la ayuda que nos brindara el docente encargado de la materia Sistemas Operativos de Redes nos ha permitido ejecutar el presente proyecto con éxito, no sin antes pasar por todos los problemas y trabajos emanados de las practicas realizadas, que nos dieron como resultado la experiencia de tener que usar software libre.

Instalar Oracle VM Virtualbox, crear las maquinas virtuales, instalar Heartbeat, Pacemaker, DRBD, Apache sobre la plataforma de Debian Wheezy, hacer las pruebas de configuración, solventar los errores que han aparecido asido gratificante pues nos ha permitido enriquecer nuestros conocimientos sobre que son, como funcionan, y para que sirve un cluster de alta disponibilidad.

Antes de este proyecto no sabíamos que era un servidor de ficheros de alta disponibilidad, ni como funcionaba, ahora podemos decir que hemos adquirido suficiente conocimiento como para configurar e instalar todo lo necesario para crearlo.

Recomendación

1. Antes de ponerse a querer crear un servidor de alta disponibilidad de ficheros seria bueno informarse primero de que es, como funciona y para que sirve.
2. Decidir si se va configurar en maquinas reales o en maquinas virtuales cada una de estas opciones tiene sus pro y sus contras.
3. Descargar el software requerido recomendamos licencias de tipo: Licencia Pública General de GNU (GPL) o una licencia propietaria con características adicionales.
4. Si se realizara con maquinas virtuales ver si la maquina física puede aceptar virtualizacion. La computadora en la que se harán las maquinas virtuales deberá estar bien dotada de recursos de hardware capas de soportar la virtualizacion y llenar requerimientos tales como suficiente cantidad de memoria RAM, un buen Disco Duro, y un microprocesador (CPU) con una buena velocidad, para que a la hora de arrancar las maquinas virtuales no se ponga lenta.
5. Crear un manual paso a paso de instalación del cluster, con capturas de pantalla y otro material que sirva para guiar a cualquier usuario que desee crear un servidor de ficheros de alta disponibilidad.
6. Apoyar la creación de un Cluster como servidor de ficheros de alta disponibilidad con la creación de un vídeo que muestre los pasos a seguir en la configuración para hacer mas fácil la misma.
7. Disponer de todo el hardware necesario para crear y configurar el cluster (servidor de ficheros de alta disponibilidad),
8. Estudiar todo lo referente a cluster de alta disponibilidad.
9. Preguntar, investigar, cuando aparezca algún error para solucionarlo. Apoyarse con el docente encargado de la materia SOR(Sistemas Operativos de Redes), o con otras personas que tengan experiencia en esta área.

Bibliografía

1. Clúster (informática)
https://es.wikipedia.org/wiki/Cl%C3%BAster_%28inform%C3%A1tica%29
2. Clúster de alta disponibilidad
https://es.wikipedia.org/wiki/Cl%C3%BAster_de_alta_disponibilidad
3. Como montar un Cluster de Alta disponibilidad en GNU/Linux
<http://gnu-linux-xx.blogspot.com/2011/01/como-montar-un-cluster-de-alta.html>
4. Alta Disponibilidad con Apache2 y Heartbeat en Debian Squeeze
<http://www.muspells.net/blog/2011/04/alta-disponibilidad-con-apache2-y-heartbeat-en-debian-squeeze/>
5. Clustering Alta disponibilidad en Linux. heartbeat
<http://www.nexolinux.com/clustering-alta-disponibilidad-en-linux-heartbeat/>
6. Guía para montar un servidor en Debian 8 Jessie

<http://dplinux.net/guia-servidor-en-debian/>
7. “Clúster de Un Servidor de Ficheros.” *JMSolanes*. Accessed August 21, 2015.
<http://www.jmsolanes.net/es/cluster-servidor-de-ficheros/>.
8. Cohn, Dennis. “Solo Confía...”: Alta Disponibilidad En Linux: Heartbeat Y Pacemaker.” *Solo Confía...*, 16, 2010. <http://denniscm20.blogspot.com/2010/04/alta-disponibilidad-en-linux-heartbeat.html>.
9. García, Alejandro. “Como Montar Un Cluster de Alta Disponibilidad En GNU/Linux.” *Todo Sobre GNU/Linux, S.O. Libres, Programacion En General Y Más.....*, 23, 2011. <http://gnu-linux-xx.blogspot.com/2011/01/como-montar-un-cluster-de-alta.html>.
10. “DrBd - Debian Wiki.” Accessed November 6, 2015. <https://wiki.debian.org/DrBd>.
11. Dominguez, Roberto. “Linux Tips: Hearbeat + DRBD.” *Linux Tips*, May 18, 2012. <http://robertolinux.blogspot.com/2012/05/hearbeat-drbd.html>.