Alta disponibilidad gracias a las tecnologías de virtualización y redes

High availability thanks to virtualisation and network technologies

Josep Vidal Canet, Sergio Cubero Torres

Resumen

Utilizando software open source (linux, xen, heartbeat, DRBD) se han diseñado e implementado arquitecturas geográficamente distribuidas (10 km) tolerantes a fallos basadas en las tecnologías de redes y virtualización. En estas arquitecturas los recursos físicos se encuentran virtualizados y replicados vía IP a larga distancia. Al conseguir presentar de manera virtual los recursos (CPU, disco y red) a los servicios independientemente de su ubicación física, podemos ejecutar los servicios de información sobre aquellos recursos físicos que estén disponibles en un momento dado. Además estas arquitecturas son capaces de detectar el fallo de algunos de sus componentes –desde un servidor a todo un centro de datos– y reorganizarse de tal manera que éste no les afecte. La reorganización consiste básicamente en migrar los servicios de información, del centro afectado por el fallo, hacia aquellos recursos computacionales que estén disponibles en un momento dado. Para facilitar el proceso de migración de los sistemas de información de unos recursos físicos a otros, éstos se ejecutan sobre máquinas virtuales. Así mismo, los discos físicos que albergan las máquinas virtuales, se encuentran distribuidos y replicados a través de la red IP. De esta manera el proceso de migración consiste simplemente en parar la máquina virtual que soporta el sistema de información del centro de datos afectado por el fallo y arrancarla sobre el centro de datos disponible. Finalmente, con el software de alta disponibilidad heartbeat, se ha automatizado el proceso de detectar los fallos de componentes de la arquitectura, y ejecutar los servicios de información sobre aquellos recursos físicos disponibles.

Palabras clave: virtualización, máquinas virtuales, migración.

Para facilitar el proceso de migración de los sistemas de información de unos recursos físicos a otros, éstos se ejecutan sobre máquinas virtuales

Summary

Using open-source software (Linux, xen, heartbeat, DRBD), geographically distributed architectures (10 km) have been designed and implemented with fail-safe mechanisms based on network and virtualisation technologies. In these architectures, the physical resources are virtualised and replicated over long distances via IP. By presenting the resources (CPU, disk and network) to the services virtually, regardless of their physical location, we can run information services on the physical resources that are available at a given moment in time. Furthermore, these architectures are capable of detecting failures in any of their components (from a server to an entire data centre) and reorganising them so that they are not affected. Reorganisation consists basically of migrating the information services from the centre affected by the failure to the computer resources that are available at that time. To facilitate the process of migrating the information systems from one physical resource to another, they are run on virtual machines. Likewise, the physical disks that house the virtual machines are distributed and replicated over the IP network. Thus, the migration process consists simply of stopping the virtual machine that supports the information system at the data centre affected by the failure and starting it at another available data centre. Finally, the high-availability heartbeat software automates the process of detecting component failures and running information services on the available physical resources.

Keywords: virtualisation, virtual machines, migration.

La Universitat de València (UV) está trabajando en este tipo de arquitecturas de cara a garantizar un buen nivel de servicio para sus sistemas de información

1. Introducción

Los actuales requerimientos de nivel de servicio de los sistemas de información, necesitan de arquitecturas tolerantes a contingencias (incendios, inundaciones, fallos...) para conseguir alta disponibilidad y reducido tiempo de respuesta. Con la ayuda de las tecnologías actuales de virtualización y redes podemos construir arquitecturas geográficamente distribuidas tolerantes a fallos, con bajos tiempos de indisponibilidad debido a fallos de alguno de sus componentes: armarios de disco, servidores, software, mantenimientos, comunicaciones, etc.

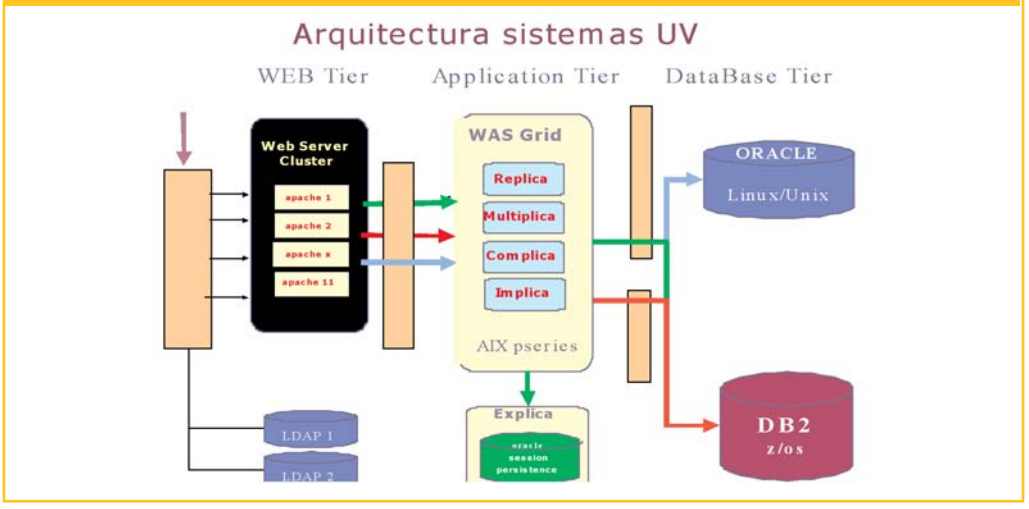
La Universitat de València (UV) está trabajando en este tipo de arquitecturas de cara a garantizar un buen nivel de servicio para sus sistemas de información. Fruto de ello, es el reciente proyecto de implantación de un centro de respaldo remoto, ubicado a 10 kilómetros del centro de datos. Gracias a esta infraestructura, se están desarrollando arquitecturas distribuidas del tipo clusters activo/activo y activo/pasivo con automatic failover, para minimizar los tiempos de respuesta y maximizar la disponibilidad de los sistemas.

La que aquí se presenta es una arquitectura geográficamente distribuida con automatic failover donde los recursos físicos se encuentran virtualizados y replicados vía IP a larga distancia. Al conseguir presentar de manera virtual los recursos (CPU, disco y red) a los servicios independientemente de su ubicación física, podemos ejecutar los servicios de información sobre aquellos recursos físicos que estén disponibles en un momento dado.

2. La alta disponibilidad en la UV

Actualmente en la UV disponemos del siguiente diseño arquitectónico (ver imagen 1) para los sistemas de información corporativos basados en los motores de bases de datos ORACLE y DB2 (automatrícula, personal, contabilidad, secretaria virtual, etc.)

IMAGEN 1



La capa web es la encargada de servir los elementos estáticos de las aplicaciones web

A grandes rasgos, podemos observar que los servidores están divididos en tres grupos o capas: los servidores web, los servidores de aplicaciones y los servidores de datos. Cada grupo está especializado en servir un determinado volumen de carga. Para cada capa se ha hecho un diseño exprofeso, de cara a garantizar un buen nivel de servicio, entendiendo por éste el % de disponibilidad y el tiempo de respuesta de las aplicaciones.

En los siguientes apartados, se describe por capas la arquitectura, razonando en cada capa el diseño elegido.

2.1. La Capa Web

La capa web es la encargada de servir los elementos estáticos de las aplicaciones web. Entre ellos, las páginas html, las imágenes, hojas de estilo, etc. Entre otros requerimientos, debe estar preparada para soportar picos de carga, sobretodo en periodos críticos como la automatrícula. Teniendo esto en cuenta, se ha diseñado un clúster de servidores web (imagen 2), donde todos los nodos están activos. Existen un total de 11 nodos web ubicados tanto en servidores físicos (linux corriendo bajo AMD opterones) como en máquinas virtuales (Xen). Las peticiones entrantes son distribuidas entre los distintos nodos (servidores web Apache), por un balanceador implementado por el software open source pound (http://www.apsis.ch/pound/). Éste detecta las caídas de los servidores web Apache, marcándolos a partir de ese momento como no disponibles. A los servidores marcados como no

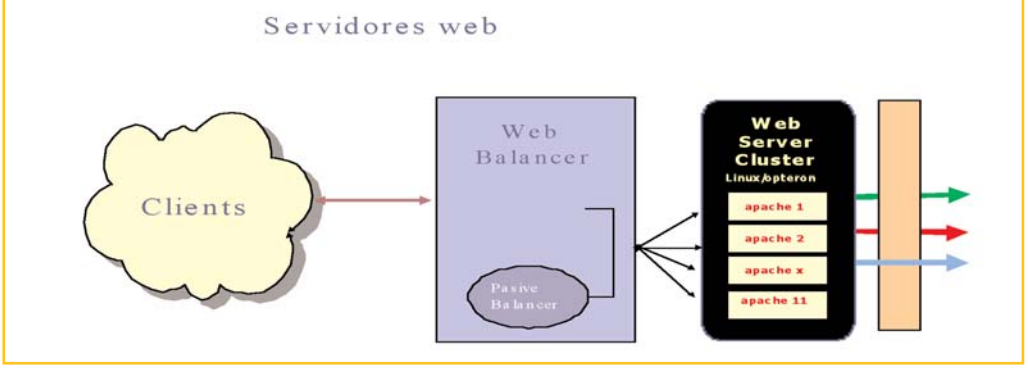
En caso de caída o mal funcionamiento, heartbeat detecta la caída del nodo balanceador migrando los recursos

disponibles, ya no se les envía más tráfico hasta que no se recuperan. Adicionalmente, cada nodo, tiene un mecanismo de failover, que permite recuperar un servidor web apache caído en menos de 2 minutos.

Para evitar que el balanceador sea el punto único de fallo de toda la arquitectura, se ha implementado un mecanismo de failover automático que garantiza su disponibilidad. Este mecanismo consiste en monitorizar el funcionamiento del balanceador con el software open source de alta disponibilidad heartbeat. En caso, de caída o mal funcionamiento, heartbeat detecta la caída del nodo balanceador migrando los recursos (IP pública del servidor y el software de balanceo) al nodo pasivo (ver imagen 2). Con este sistema logramos que el tiempo de indisponibilidad debido a una caída del balanceador sea inferior a los 20 segundos. Esto se consigue, entre otras cosas, gracias a la actualización de las cachés de ARP, mediante el envío de un ARP gratuito para informar de la nueva MAC address asociada a la IP del balanceador.

La comunicación de la capa web, con la de aplicaciones se hace mediante el plugin de websphere

IMAGEN 2



2.2. Servidores de aplicaciones

Tanto la capa web, como la capa de aplicaciones pueden ser implementadas fácilmente en arquitecturas distribuidas, escalables y tolerantes a fallos

La mayoría de las aplicaciones web corporativas, se han desarrollado siguiendo el estándar J2EE. El runtime elegido para correrlas ha sido el WebSphere Application Server, debido a su mejor integración con los sistemas legacy (mainframes). Utilizando este runtime, se ha implementado un grid (distintas máquinas con distintos SO colaborando en un mismo objetivo) para la ejecución de aplicaciones J2EE. De la misma manera que los servidores web se especializan en servir peticiones de objetos estáticos (html, imágenes, etc.), el grid se especializa en servir objetos dinámicos: componentes J2EE tales como JSPs (Java Server Pages), servlets y EJBs (Enterprise Java Beans).

Este grid puede verse desde un doble punto de vista :

- Por un lado, tenemos las máquinas físicas: Cinco servidores pseries (power4, power5), corriendo distintas versiones del sistema operativo AIX. Cuatro de los cinco servidores se dedican al runtime encargado de correr las aplicaciones java. El quinto servidor está dedicado a albergar la persistencia de sesiones, que pronto explicaremos.

- Desde el punto de vista lógico podemos ver el sistema como un conjunto de clusters de máquinas virtuales J2EE (JVMs). El número de JVMs que contiene un clúster, así como su distribución por los distintos servidores físicos, dependerá de la criticidad de las aplicaciones que este corre. A más criticidad, mayor número de máquinas virtuales y mayor distribución de éstas por los distintos servidores físicos. Así mismo las sesiones de las aplicaciones albergadas en clusters críticos, se

almacenan en Bases de Datos (BBDD). De esta manera en caso de que algún componente del clúster caiga (un servidor o máquina virtual java), las sesiones que éste contenía, son recuperadas de la BBDD para ser servidas por otro miembro del clúster. Este mecanismo permite que el usuario no note el fallo de un componente de la arquitectura (JVM o servidor físico).

Finalmente, la comunicación de la capa web, con la de aplicaciones se hace mediante el plugin de websphere (imagen 1). Éste es un módulo que se añade al servidor web Apache, con la finalidad de enrutar las peticiones de componentes J2EE (JSPs, Servlets, EJBs) a las distintas máquinas virtuales ubicadas en la capa de los servidores de aplicaciones. De esta manera, cada vez que un objeto dinámico es demandado, el servidor Apache lo pasa al plugin. Este comprueba en que clúster está instalado, para finalmente enrutarlo hacia una máquina virtual de este clúster, ubicada en un determinado servidor físico.

2.3. Sistemas de información

Tanto la capa web, como la capa de aplicaciones pueden ser implementadas fácilmente en arquitecturas distribuidas, escalables y tolerantes a fallos. Esto en gran medida es debido a que la información que éstos contienen es modificada esporádicamente (2 o 3 veces al día de media), de manera que es relativamente sencillo mantener sincronizada la información en una arquitectura distribuida del tipo clúster. Por ejemplo, cada vez que se instala una aplicación en un nodo del clúster web o del servidor de aplicaciones el resto de nodos se sincronizan con el primero para quedar el clúster en un estado consistente.

Actualmente no existe una solución general para todos los sistemas de información

Sin embargo, implementar los sistemas de información en una arquitectura distribuida, escalable y tolerante a fallos, no es una cosa tan sencilla. En la UV se han probado varias alternativas para afrontar este problema. Entre ellas:

- Mantener replicados y sincronizados aquellos sistemas de información cuya frecuencia de actualización es muy baja. Esta solución se ha aplicado con éxito para algunos sistemas informacionales.

- Implementar arquitecturas SSI (Single System Image). Una imagen única de sistema (Single System Image, SSI), tal y como se define en la wikipedia, es una propiedad de un sistema que oculta la naturaleza heterogénea y distribuida de los recursos, y los presenta a los usuarios y a las aplicaciones como un recurso computacional unificado y sencillo. En concreto, en la UV se ha testeado el software open source OpenSSI (ssi.uv.es), aunque todavía el proyecto no ofrece el grado de madurez necesario para un entorno de producción.

- Sistemas Gestores de Bases de Datos (SGBD) clusterizables. Son una buena solución para la alta disponibilidad de BBDD (bases de datos). Para el caso del SGBD ORACLE, existe la posibilidad de clusterizar la BBDD. La versión ORACLE 10g ha facilitado y mejorado la clusterización de la BBDD gracias a la incorporación de la tecnología GRID. En la UV, se ha probado el producto clusterizando una base de datos ORACLE 10g, utilizando el sistema de ficheros OCFS (ORACLE Cluster File System) sobre unos discos compartidos por iSCSI.

Gracias a las tecnologías de redes y virtualización podemos diseñar arquitecturas distribuidas tolerantes a fallos, donde los recursos físicos se encuentran virtualizados y replicados vía IP a larga distancia

Actualmente no existe una solución general para todos los sistemas de información. Mientras que algunos SGBD ofrecen la posibilidad de clusterizar la BBDD, como el caso de ORACLE 10g, éstos suelen requerir de un medio de almacenamiento compartido. Este medio compartido, sobre el que se monta un cluster file system, suele ser el punto único de fallo de toda la arquitectura. Además, hay que tener en cuenta que algunos sistemas de información no se circunscriben a una única fuente de datos. Estos, suelen ser sistemas heterogéneos que constan de varias fuentes de datos, posiblemente implementadas en distintos SGBD, de ficheros, usuarios, aplicaciones, scripting, procesos batch, etc.

Todos estos problemas se intentan resolver en la arquitectura que a continuación se presenta.

3. Arquitectura propuesta

Teniendo en cuenta nuestras necesidades, la arquitectura ideal para la alta disponibilidad de sistemas de los sistemas de información, sería una solución del tipo SSI (Single System Image), de manera que presente los recursos hardware distribuidos como un solo sistema, de cara a los usuarios y aplicaciones. Desafortunadamente, la tecnología no está lo suficientemente madura como para implementar sistemas de misión crítica en ella.

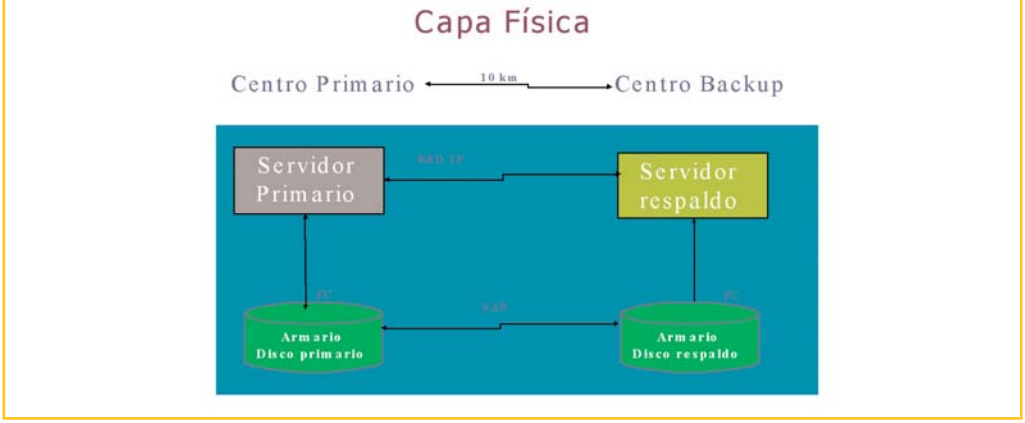
No obstante, gracias a las tecnologías de redes y virtualización podemos diseñar arquitecturas distribuidas tolerantes a fallos, donde los recursos físicos se encuentran virtualizados y replicados vía IP a larga distancia. Al conseguir presentar de manera virtual los recursos (CPU, disco y red) a los servicios independientemente de su ubicación física, podemos ejecutar los servicios de información sobre aquellos recursos físicos que estén disponibles en un momento dado. Ésta arquitectura es ideal para servidores con persistencia de datos tales como bases de datos, servidores de correo, servidores de ficheros, etc.

Es misión del administrador tener esto en cuenta a la hora de elegir tanto el sistema de ficheros como el SGBD

Los componentes de esta arquitectura los podemos dividir en dos grupos:

a) Los recursos físicos geográficamente distribuidos entre el centro primario y el de respaldo: las redes IP y de almacenamiento SAN y los servidores físicos (imagen 3).

IMAGEN 3



b) Los recursos lógicos, software open source de virtualización (XEN), redundancia de discos por IP (DRBD) y failover automático (Heartbeat).

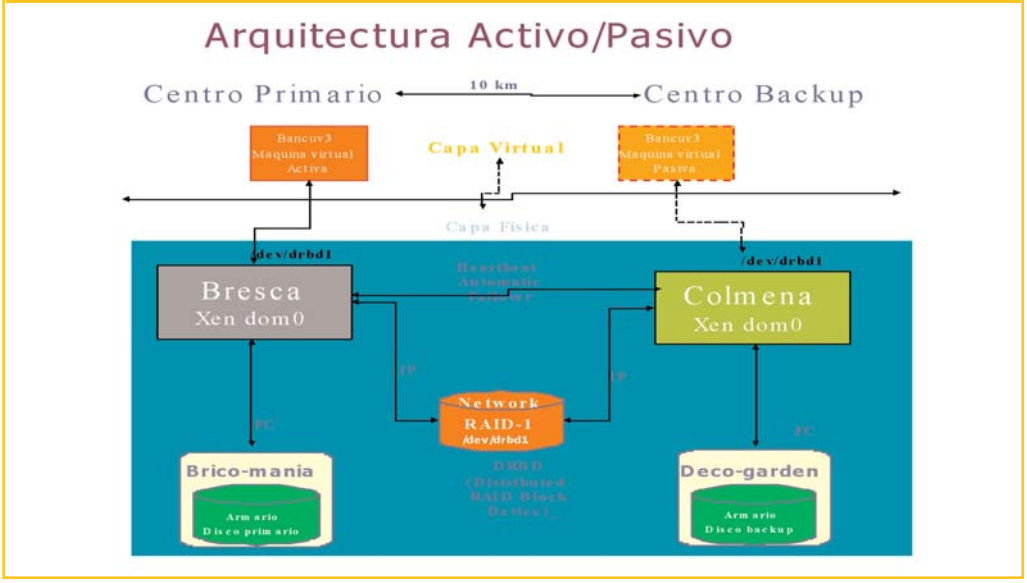
La arquitectura activo/pasivo presenta un solución flexible, sencilla y barata para implementar arquitecturas de alta disponibilidad

A continuación se describe el diseño de la arquitectura de alta disponibilidad: los servidores tienen definidos LUNs (Logical Unit Number) en cada uno de los armarios de disco. Con el software de mirror por IP DRBD (Distributed RAID Block Device), se hace un espejo de las LUNs del centro primario contra las del centro de respaldo. Sobre estos espejos se definen las máquinas virtuales XEN sobre las que corren los servicios de información. Estas máquinas se ejecutan por defecto, en las CPUs del centro primario, modificando la información en las LUNs del armario de discos primario, la cual a su vez es replicada por la red IP contra el armario de discos de respaldo. Ante cualquier contingencia (fallo de armario de discos, fallo del hardware del servidor primario), el software de failover automático (heartbeat), detecta la indisponibilidad del servicio de información, migrando las máquinas virtuales del centro primario a los recursos computacionales (CPUs, discos) del centro de respaldo. Ante una

contingencia, el usuario solamente detecta un tiempo de indisponibilidad del servicio inferior al minuto; el necesario para mover los servicios de información del centro primario al de respaldo.

Veamos paso a paso el funcionamiento de la arquitectura activo/pasivo propuesta ante una contingencia. Tal y como podemos ver en la figura 4, por defecto la máquina virtual Bancuv3, la cual implementa el sistema de información Secretaria Virtual (expedientes académicos, datos personal, nóminas, etc.) corre sobre los recursos hardware (servidores, discos, etc.) del centro primario.

IMAGEN 4



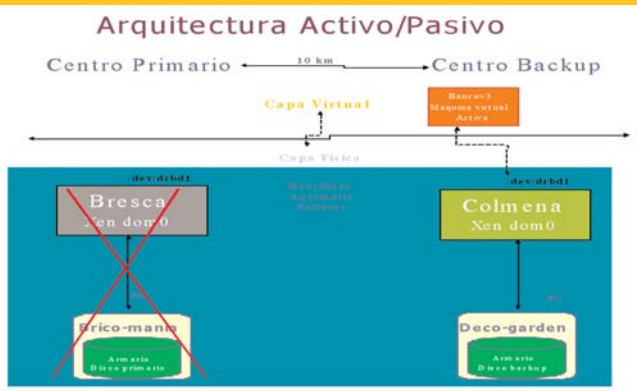
Una solución mejor consiste en utilizar los recursos del centro para albergar nuevas máquinas virtuales

Imaginemos que en un momento determinado ocurre un fallo sobre un recurso (avería hardware del servidor, fallo del armario de discos, inundación del centro primario, etc.) Evidentemente, ante este tipo de contingencias el SI dejará de funcionar. Llegados a este punto, el software de failover detectará la indisponibilidad del servicio. Para ello, está continuamente monitorizando (cada 10 segundos) el SI, para ver si está vivo o no. Una vez detectado el fallo del nodo primario, heartbeat procederá a levantar el sistema de información en el nodo del centro de respaldo. Para conseguir esto último, heartbeat llama a un script desarrollado por el administrador encargado de automatizar las acciones encaminadas a parar y arrancar los recursos en un determinado nodo. Para este caso en concreto, la única cosa que tiene que hacer el script para migrar los recursos de un nodo a otro, es reconfigurar el mirror de discos por IP para informar que vamos a utilizar el disco del centro de respaldo como primario (drbdsetup /dev/drbd1 primary) y levantar la máquina virtual sobre el servidor físico del centro de respaldo (xm create bancuv3.cfg). En este momento tenemos la máquina virtual arrancando sobre los recursos físicos del centro de respaldo. Como los discos del centro primario los tenemos en espejo contra los del centro de repaldo, no existe pérdida de información (en caso de que el protocolo de utilizar un protocolo síncrono) o la pérdida es mínima (protocolo asíncrono). Es misión del administrador tener esto en cuenta a la hora de elegir tanto el sistema de ficheros como el SGBD. Para estos entornos, es ideal elegir un File System (xfs, ext3, etc.) y un SGBD que automáticamente se recuperen en un estado consistente ante paradas abruptas. De esta manera, durante el proceso de inicialización de la máquina virtual se procederá a dejar en un punto consistente tanto el File System como la BBDD, para lo cual se desharán aquellas transacciones que no se hayan podido confirmar.

El proceso de migración consiste en parar la máquina virtual que soporta el sistema de información del centro de datos afectado por el fallo y arrancarla sobre el centro de datos disponible

La situación final, es la siguiente: Los recursos del centro primario (servidores, disco), sobre los que estaba corriendo el SI no se encuentran disponibles debido a la contingencia. A causa de esto, se ha migrado la máquina virtual que soporta el SI a los recursos hardware del centro de respaldo (imagen 5). La pérdida de información provocada por la contingencia ha sido mínima o ninguna dependiendo del protocolo que utilicemos para implementar el mirror por IP. Después de todo, los usuarios han detectado un tiempo de indisponibilidad inferior al minuto; el necesario para parar la máquina virtual en el servidor primario y arrancarla sobre el servidor standby/pasivo del centro de respaldo. Por último, destacar que una vez disponibles los recursos hardware, la migración de la máquina virtual al centro primario es trivial. Tan solo consiste en pararla en el servidor de respaldo, reconfigurar el mirror y arrancarla en el servidor del centro primario. Con la versión 8.0.6 del software DRBD, el proceso de migración de máquinas virtuales se puede hacer en caliente, i.e., sin que los usuarios noten la parada del servicio.

IMAGEN 5



La arquitectura activo/pasivo presenta un solución flexible, sencilla y barata para implementar arquitecturas de alta disponibilidad

4. Variación arquitectónica: Activo/Activo

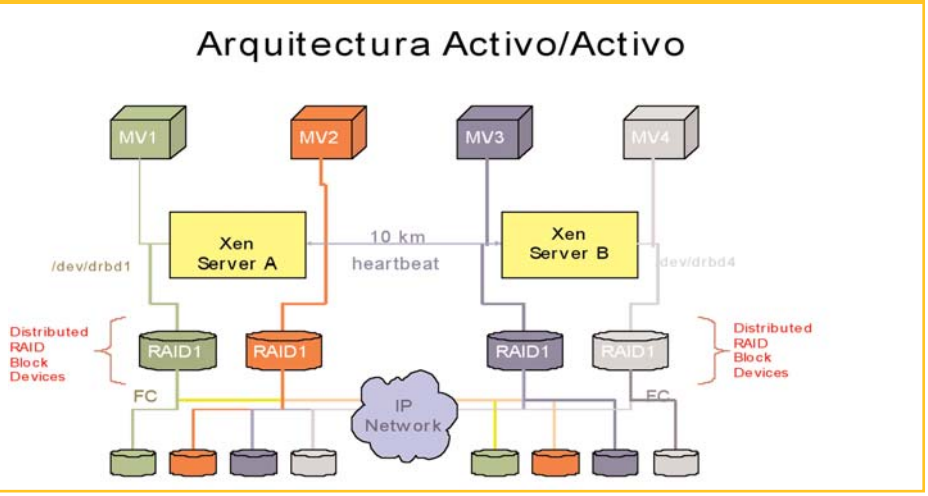
La arquitectura activo/pasivo anteriormente comentada, presenta una solución flexible, sencilla y barata para implementar arquitecturas de alta disponibilidad. El único inconveniente reside en el hecho de que la mitad de los recursos permanecen ociosos a la espera del fallo del servidor primario. Una solución mejor consiste en utilizar los recursos del centro para albergar nuevas máquinas virtuales. De esta manera ante una contingencia en uno de los dos centros de proceso de datos (el primario o el de respaldo), las máquinas virtuales son migradas a los recursos computacionales del centro no afectado por ésta. Evidentemente no podremos ofrecer el mismo nivel de servicio cuando ocurre un fallo, por ejemplo, en uno de los dos servidores, ya que el restante debe cargar con la máquinas virtuales del servidor averiado. Teniendo esto en cuenta la arquitectura activo/activo basada en las tecnologías de redes y virtualización queda como sigue:

Ante cualquier contingencia que afecte a uno de los dos centros, heartbeat detectará el fallo de los servicios de información

Disponemos de un conjunto de máquinas virtuales distribuidas entre servidores del centro primario y el centro de backup (imagen 6). Estás máquinas virtuales, correspondientes a diferentes SI (secretaría virtual, aula virtual, académico, personal...) contienen su información en diferentes espejos (/dev/drdb1, /dev/drdb2, /dev/drbd3, /dev/drbd4). Estos espejos, mantienen sincronizados por la red IP, los discos Fiber Channel del armario del centro primario contra los discos del armario del centro de backup. Adicionalmente disponemos del software de alta disponibilidad y failover automático heartbeat, monitorizando tanto los servidores y servicios del centro primario como los del centro de respaldo. Ante cualquier contingencia que afecte a uno de los dos centros, heartbeat detectará el fallo de los servicios de información. Finalmente, procederá a migrar los servicios de información del

centro afectado por la contingencia al centro de proceso de datos disponible. Evidentemente, una vez migrados los recursos del centro afectado al otro, el servidor restante dispondrá del doble de carga: además de las máquina virtuales que ya contenía, se le añaden las del servidor del centro afectado. Es por lo cual, que habrá más competencia por los recursos hardware reales reduciendo el rendimiento de los distintos sistemas de información.

IMAGEN 6



Habrá más competencia por los recursos hardware reales reduciendo el rendimiento de los distintos sistemas de información

5. Conclusiones

A lo largo del presente artículo se han presentado las distintas arquitecturas de sistemas utilizadas en la Universitat de València, para garantizar un buen nivel de servicio de los sistemas que automatizan el negocio universitario. Especial énfasis se ha hecho con la problemática de diseñar sistemas altamente disponibles para la capa de datos. Para este tipo de sistemas, utilizando software open source (linux, xen, heartbeat, DRBD) se ha diseñado y implementado arquitecturas geográficamente distribuidas (10 km) basadas en las tecnologías de redes y virtualización. Estas arquitecturas son capaces de detectar el fallo de algunos de sus componentes –desde un servidor a todo un centro de datos– y reorganizarse de tal manera que éste no les afecte. Esta reorganización consiste en migrar los servicios de información, del centro afectado por el fallo, hacia aquellos recursos computacionales que estén disponibles en un momento dado. Con el objetivo de facilitar el proceso de migración de los sistemas de información de unos recursos físicos a otros, éstos se ejecutan sobre máquinas virtuales, en vez de directamente sobre el hardware. Asimismo, los discos físicos que albergan las máquinas virtuales, se encuentran distribuidos y replicados a través de la red IP. De esta manera el proceso de migración consiste simplemente en parar la máquina virtual que soporta el sistema de información del centro de datos afectado por el fallo y arrancarla sobre el centro de datos disponible.

Estas arquitecturas son capaces de detectar el fallo de algunos de sus componentes y reorganizarse de tal manera que éste no les afecte

Josep Vidal Canet
(josep.vidal@uv.es)
Sergio Cubero
(sergio.cubero@uv.es)

Técnicos de sistemas de la Universitat de València